Using BAM and CEP for Process Monitoring in Cloud BPM

José Nicolás Martinez Garro
Facultad de Informática –UNLP - La Plata, Argentina
josemartinezgarro@gmail.com
and
Patricia Bazán
LINTI – Facultad de Informática –UNLP-La Plata, Argentina
pbaz@ada.info.unlp.edu.ar
and
Javier Díaz
LINTI – UNLP - Universidad Nacional de La Plata - La Plata, Argentina
jdiaz@info.unlp.edu.ar

***Abstract*— BAM (Business Activity Monitoring) and CEP (Complex Event Processing) have been deeply studied in traditional BPM, according to the procurement of relevant results about process execution, especially for improvement and monitoring purposes. Once BPM was introduced in the cloud and it is necessary to decompose processes in order to execute them in hybrid environments, the conception of monitoring has suffered several changes. In previous works we have designed an architecture for process execution and monitoring which considers several services in order to gather and show the recollected information as relevant results. A new approach over this architecture takes place in this work: it considers BAM and CEP as mechanisms for the information gathering process, and also enhances the monitoring application taking advantage of these two methodologies, now under the distributed perspective. Additionally we make a comparison about when is convenient to use BAM or CEP alternatively in a distributed environment, according to the nature of the recollected data, the type of indicators needed to be shown and also the latency window in the data used as a source.**

Keywords*— *BAM, CEP, BPM, Monitoring, cloud.

1. INTRODUCTION

The monitoring phase is one of the most important in the business process lifecycle, assuring the continuous improvement which is one of the major principles in the business process oriented paradigm. Both execution and monitoring aspects take a different meaning if they are taken from a classic centralized architecture through a hybrid collaborative environment where cloud and embedded components are combined, and even more these last ones could be public or private.

Facing this decentralized context, it is important to conserve the perspective of the original process model which has a centralized view, and on the other hand, to register every single detail and complex event related to the process in a distributed environment. All these data are really important in order to embrace the biggest amount of information which results useful for process measuring and improvement.

The current work takes a previous architecture implemented in [6] [7] and introduces over it two different technologies: BAM (*Business Activity Monitoring*) and CEP (*Complex Event Processing*) as a new way of obtaining useful data from the different distributed nodes in the architecture. Even though BAM and CEP are not new in the BPM paradigm, in this context they take a new significant because the instances to be monitored and the events to be followed belong to process instances which are distributed all over the cloud.

The present work is structured as follows: in Section II some related works are presented. In Sections III to V we make a review about BAM and CEP concepts and the implications of them being considered under the cloud perspective. We also revise the architecture presented in [6] and [7], and it is modified in order to introduce the BAM and CEP elements but with a distributed perspective. In Section VI different indicators corresponding to BAM and CEP are analyzed in order to implement the continuous improvement. Finally in Section VII some conclusions and ideas are presented for future works.

2. RELATED WORK

There are many references to BAM and CEP in current bibliography. Typically these two terms are considered under an embedded or traditional perspective. For example, in [1] a formal proposal for event processing in BPM is presented. This paper addresses the concept of CEP applied to BPM but under a traditional perspective, i.e. an embedded solution. Some concepts are taken from there and extrapolated to a cloud environment in the present work, especially those referred to the second generation of BAM. In [2] a similar idea to [1] is presented, but with some distributed elements, exploring the idea of BAM as a multi-application platform. In [3] it is possible to see a concrete implementation for monitoring cloud nodes, in particular nodes that include BPM and SOA applications, and considering also CEP alternatives. The main goal in the quoted work is to obtain relevant indicators for the continuous improvement cycle. In [6] and [7] we find a previous construction of an architecture for hybrid systems that takes in consideration different methods other than CEP and BAM in order to collect information about the distributed instances. This previous contribution was taken to be modified and complemented using these two named techniques in order to enrich the monitoring and continuous improvement lifecycle.

3. BAM and CEP

These two concepts were developed several years ago, and are often taken as mechanisms to obtain relevant information in service oriented systems, and particularly in BPM. In this section we analyze them under a traditional perspective in contraposition to a distributed or hybrid view, which is the main goal of the present work

BAM in traditional BPM

There are several considerations that we can establish in order to review the BAM and BPM relationship over these last years. Business Process Management and Business Activity Monitoring have separate lives and deployment patterns; as distinct technologies for the real-time enterprise, they are different arrows in the IT quiver. However, there are several topics that indicate a strong linkage between them:

• They are highly complementary and are partially converging. Even though they have independence since their conception, there are several grey areas where they connect.

• The enterprise will have many BAM and BPM tools, especially if it is considering a cloud based environment, with private and public nodes.

• BAM and BPM will often be deployed together to solve business-level problems. BPM is a useful income to BAM in order to support the decision making process.

• Some BAM functional requirements will be met by BPM's monitoring functions, instead of a "classic BAM" tool, because of the constant evolution they have presented.

BPM and BAM have three main areas of convergence (and potential conflict):

• BPM acting as "BPM+BAM"

• BPM serving as BAM's response mechanism or recipient

• BPM as a passive analytic/visualization model for BAM

It is important to remember that BAM is multi-application, correlating multiple sources of independent data; in each of these convergence scenarios, there is an implicit assumption that BAM is working with more than one underlying application [10] [1] [2].

Much of the focus of BAM has been simply put on process measurement. This has certainly proved fruitful, but companies started later to use their growing level of BAM expertise to target specific business problems, enabling them to deliver greatly increased returns. A range of second-generation strategies have emerged and leaded BAM into a much closer participation in the business, generating value way beyond than the first-generation process measurement approach. In particular, three new strategic areas have been developed:

• Business Assurance and Visibility

• Control Services

• Complex Pattern Recognition

Several high-level reviews of these new strategies and the associated techniques to implement them were made, followed by study case experiences and strategies put into practice. The key point is that companies across all industries have been examining BAM closely, particularly in light of these new developments and the CEP contributions. BAM can deliver significant benefits and address a wide range of business solutions. Anyone either using or looking to use BAM should consider carefully the strategies and techniques discussed below in order to deliver maximum business value and returns.

There are four basic elements for BAM, especially considering the second generation, which makes especial focus in transactions and events. These elements are concentrated in volumes, velocities, errors and special conditions. In terms of volumes, examples that should be measured are number of transactions, number of process events, transaction revenue, process revenue, line of business revenue, cost, margin, number of changes in a record, number of items consumed, number of calls, number of closed tickets, number of errors, inter alia.

In terms of velocities, there are several cases that should be considered. Some of them are process cycle-time, cycle-times of individual steps, wait-times between events, time remaining to completion and process throughput. In terms of errors, it is important to consider that the value and time-related attributes of business transactions provides vital information about the overall operational health of the business systems. This is fine when everything is working reliably, but even in the best systems there will always be problems. These may be due to flaws in processes, external problems such as hardware or software issues, or perhaps human errors. BAM tracks errors too, making it possible to identify where the problems are, so they can be fixed [10] [2] [3] [4].

Counting and measuring errors statistically helps to improve the understanding of errors, their frequencies, and any associated trends. The final aspect of BAM measurement relates to special conditions. These are conditions defined by the user. Although not specifically errors, they represent events that are relevant from a user perspective to the execution of business transactions. As with all the other measurements, BAM will track these special conditions and provide statistical and analytical information about them, raising alerts or taking actions when specified conditions are met. For example, a company might want to be alerted to any orders beyond a certain size or the presence of non-standard shipping instructions. In Fig. 1 it is possible to see the different components in a BAM engine environment [10] [4] [5].

CEP in traditional BPM. Relationship with BAM

The main goal of CEP is the real-time analysis of events, which provokes the comparison of databases that structure and analyze data sets. Simplified, CEP can be seen as a database approach turned around where databases establish a static amount of data and allow definition of queries which are then executed. They analyze the present amount of data and deliver a result which is always just as up to date as the database of the query. CEP defines event streams as structures: on the one hand event streams allow routing of arbitrary incoming events and a goal-oriented analysis; on the other hand queries are predefined on one or more event streams. Other systems like databases, for instance, can also be integrated into CEP queries for comparison reasons [1] [2] [5] [8].

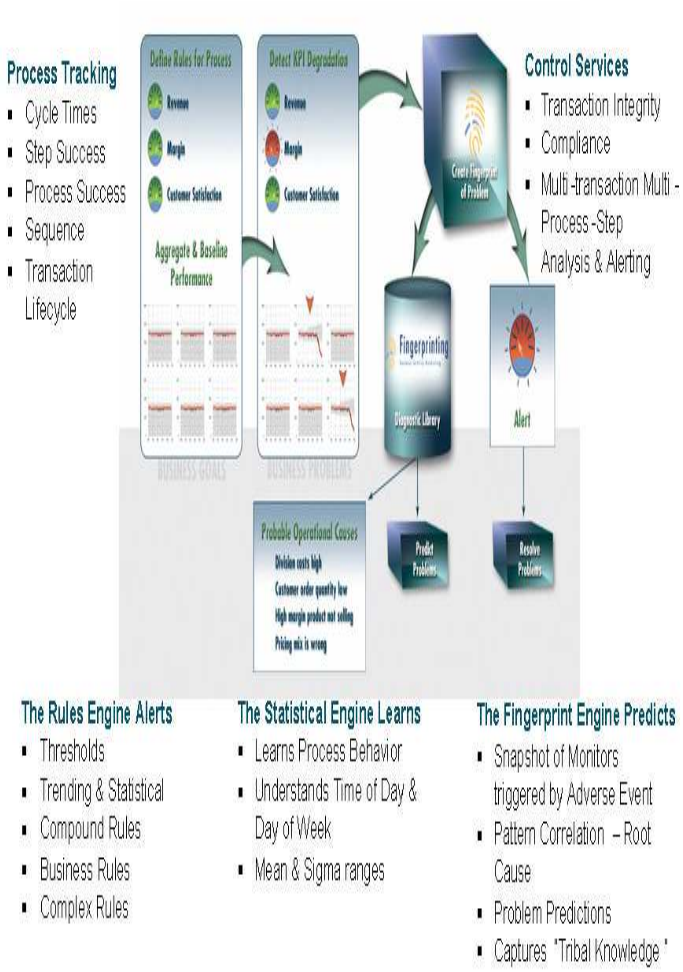

Fig. 1: BAM engines and components [4]

As events arrive arbitrarily, a static database is not given in CEP. Therefore, the predefined queries are evaluated frequently and searched for certain patterns in the event streams. The results can be displayed in frequently updating diagrams or some kind of trigger can be defined on the occurrence of a special pattern. As events are analyzed directly after they are delivered to a CEP system, results are generally nearly reflecting a real-time status. Furthermore, this approach can be used to filter important data and define queries for use cases with a huge amount of events or data. Consequently, CEP is also applicable to scenarios where databases cannot manage data storing fast enough without losing some data sets.

In this context, the relationship between BPM and BAM/CEP seems to be complementary. BAM is really useful in order to consider information about the activities in the process, about the different instances running inside the process engine, and for the business indicator making process. CEP is more global, it comprises more than just process instances and activities. It can also monitor different events related to business processes but not particularly coming from process instances, like peripheral data that belongs to the business and need to be consider in the decision making process. Because of these facts, CEP results very important and complementary to BAM in order to construct a dashboard that helps business analysts to provide important contributions about processes, about the business in general, and to make improvements in the business process lifecycle [5] [8] [9] [10].

In Fig. 2 it is possible to see the interaction between the different components according to each layer, and how the BAM engine and the CEP suite could interact in both directions in order to exchange relevant results.

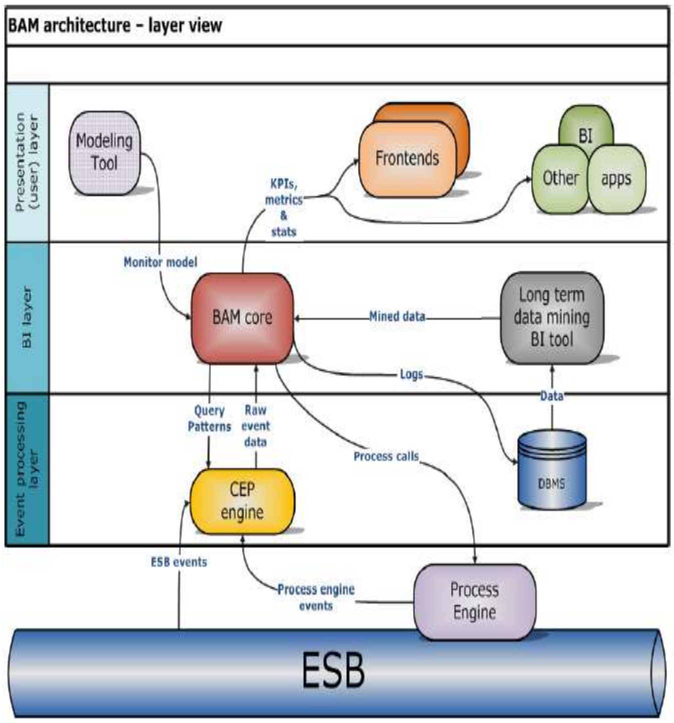

Fig 2: Layers, BAM and CEP [5]

BAM and CEP under a distributed perspective

In some previous works ([6], [7]) we have seen the interaction between the different components in a cloud oriented BPM system, where each part could be located in a public o private node. In case the system is a hybrid choice and there are several BPM engines executing at the same time, the monitoring function should collect data from each of them and preserve the original process perspective. If there are several BPM engines executing all over the architecture, also several BAM engines cloud exist. The question here is, while process decomposition is the mechanism to distribute different parts of the process along the architecture, and then it is necessary to join the parts in order to preserve the original process view, it is important to establish a mechanism to consider the results from the different BAM servers corresponding to each part of the process seamlessly [6] [7] [11] [12].

The indicators that could be obtained from any monitored node in the architecture should be considered now under the original decomposed process perspective. So the results in regard of volumes, velocities, errors and special conditions should be merged in order to provide global indicators but considering each node of the architecture. The architecture will be composed now for several BAM cores (one for each execution node) which contain the KPIs (*Key Process Indicators*), metrics and statistics according to the portion of the original process that was assigned by the distribution process. The final merge of the KPIs, metrics and statistics should be done with the original process in perspective, in order to provide a useful set of indicators and not only a partitioned view [10] [1] [2] [13] [14].

As it is shown in Fig. 3, CEP can be added to a BPM suite and used to analyze BPM events. These events as well as the queries are defined at modeling time. The stream architecture in the CEP system can be automatically generated based on the given definitions or modified manually, according to the structure of the BPMS. At runtime, defined events are created by running processes and delivered to an input stream automatically. Results of the queries can be observed via diagrams, or reactions can be defined also. In the context of a hybrid system, there are several BPM engines running at the same time and producing useful results for the CEP suite. For that reason it could be necessary to provide more than one suite, according to the embedded or cloud characteristic of the BPM engine. In this way, each suite is going to process the results related to the same area which it belongs, and the final results should be merged in order to provide an integrated view according to the original process [10] [15] [16].

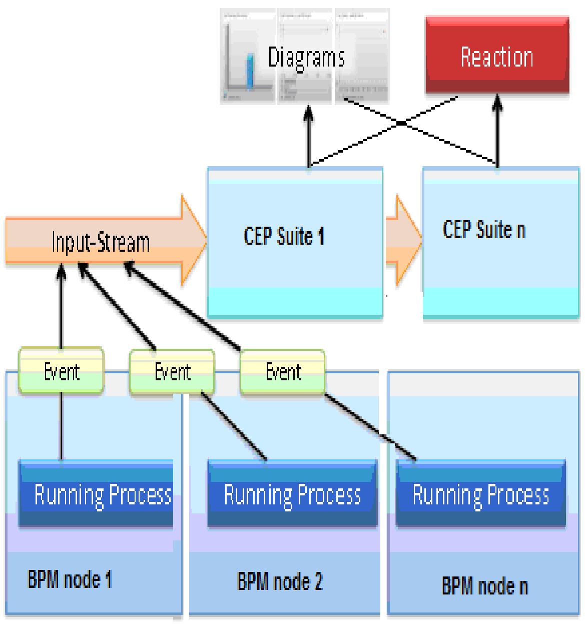

Fig. 3: CEP using distributed BPM data as input

By integrating CEP into BPM in this way, the BPM events can be analyzed in real-time. Consequently, each running decomposed process can be analyzed in real-time in respect to their processing behavior. Additionally, also complex queries including more than one or even all running processes can be realized. This makes also situations dependent on different processes analyzable. Trends can be found very fast, for instance, when all instances are included on each node. As a result, interference options are given if unwished behavior takes place. Current trends can be controlled or redirected as they are recognized while they establish and not afterwards. Another example is that pure system interactions can be controlled directly. These interactions happen too fast for human observers, but coupling these interactions with events allows for continuous observing of the results without diving into the details.

The overall message is: with the usage of CEP in BPM, knowledge is not derived after process executions, but during execution, especially in a hybrid environment where several nodes and instances are involved. This is an absolute necessity in every use case that needs fast and direct interference dependent on special constellations in a BPM server at runtime.

Another implication of using CEP in a distributed BPM environment is the reactions that CEP could originate according to the flows. This approach assumes that a CEP suite is available on each node, and analyzes any kind of external events. These events are produced by an external software or hardware which is not further specified here, but could be also present in the same hybrid environment. In a CEP suite, streams and queries are defined to analyze the incoming events with the intention of searching for patterns that imply a trend or situation that needs to be reacted to.

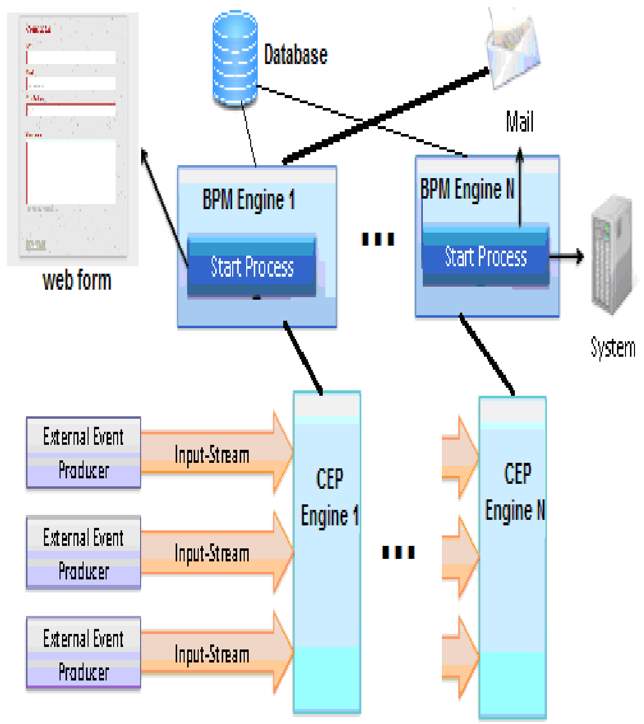

Fig. 4: CEP and distributed BPM in reaction execution

The reactions are considered to be complex and include calls to different systems or humans, or even to BPM engines in other nodes of the architecture. Therefore, the reactions are defined as processes in a BPM suite and pattern detection leads to process starting and execution (Fig. 4). The process executes all actions needed. It can call other systems, send notifications via e-mail or just add process steps to a worklist, for instance. We have to remind that these processes could be decomposed parts of the original scheme, so the integration between events and decomposed processes goes even further deep [1] [2] [17].

Using processes to define and execute reaction definitions provides several advantages. Processes could have a graphical presentation which enhances understandability; as long as the original decomposed process perspective is present. Therefore, the reactions are easy to maintain. Furthermore, they provide a high degree of flexibility. Processes can be modified or adapted to new environments very easily. Additionally, they can be substituted by other processes in a BPM suite without effects in the CEP suite if both systems are loosely coupled. This loose coupling also enables a CEP suite to be integrated in any other system without affecting the BPM suite. All these features show that processes used to define reaction definitions in CEP can be very useful [18] [19] [20].

4. MODIFICATIONS IN THE ARCHITECTURE

As we have seen previously the architecture that was presented in [6] and [7] has several modules specialized in the execution of decomposed processes, and in the recollection of data for the monitoring application. The main goal during both activities is to preserve the perspective of the original process, in terms of execution and monitoring. The user executing the process should not actually realize which concrete node of the system is executing the current activity, and in terms of monitoring it should be transparent also the node being accessed in order to obtain relevant information for the indicator construction process.

In terms of the concepts introduced in this work, the execution of business processes will remain in the same way. Once the decomposition criterion was fixed, several parts of the process are constructed and deployed in several nodes all over the architecture, where each node contains a BPM engine. Each part of the process will link the next one by using process connectors: once the first one is finished it will start the second one in a remote server and so on.

In terms of monitoring, several modifications should be introduced. As well as for the execution feature where several BPM engines are considered, for monitoring purposes one BAM engine should correspond each execution node. In this way the results of the BAM engines should be merged in order to produce indicators that represent the original process perspective, in despite of the applied decomposition [21] [22].

The monitoring application in charge of the recollection process will interact with each BAM node in the architecture, and should be in charge of merging the results in order to produce indicators related to the original process. This application should also consider the results produced by the CEP suites, which are distributed in the architecture also. In terms of the amount of CEP suites necessary to be deployed, the criterion is not the same compared to BAM. The nature of the data processed by the CEP suite is heterogeneous and could proceed from different kinds of applications, not only BPM engines. In this way, according to the size of the final architecture the system could contain as many CEP suites as necessary to process efficiently the amount of events present in the global system and usefully for the indicator making process [1] [2] [6] [7] [23] [24].

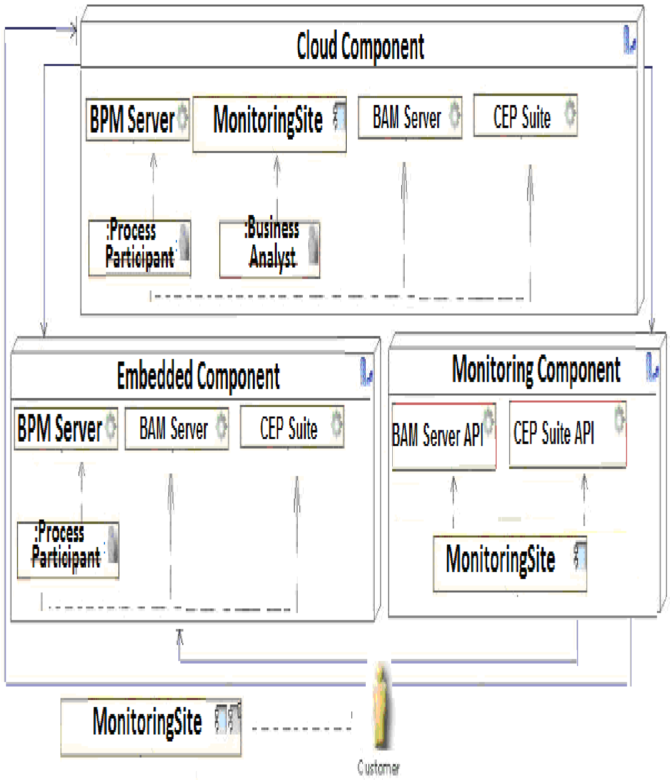

Fig. 5: Architecture components considering BAM and CEP

In Fig. 5 the global architecture is illustrated. It has three main components: the cloud component, the embedded component and the monitoring application. The cloud component has a BPM engine (it could be replicated on several nodes) and the monitoring site which is common for all users; a BAM server (follows the same replication criterion as the BPM engines) and a CEP suite which complements the incomes for the indicator making process [25] [26].

On the other hand an embedded component is present. This is representing the nodes which should remain in traditional conditions according to reasons like application portability or data sensibility. A BPM engine is present in this component (also it could be replicated), and the BAM and CEP suite follow the same role as it was described for the cloud component. The third element is the Monitoring Component which is in charge of producing relevant results about the distributed processes for the decision making process. An important detail to remark in this context is the access that the process participant could have to the distributed BAM and CEP suites on the one hand, and the relationship between them and the Monitoring Component on the other. If a process participant (or one with administration permissions) decides to access the BAM component or the CEP suite in an individual node, the provided results will correspond to a single part of the decomposed process. This could be useful for performance or debugging issues while the process is being designed and refined. The Monitoring site presents a completely different situation. This application is in charge of displaying results according to the original perspective of the decomposed process. So, it will interact with the APIs associated to the BAM and CEP suites in order to recollect the results and join them in order to show the information seamlessly. The partial results correspond to each partition of the original process, while the final results correspond to the global process once the intermediate results were merged [27] [28].

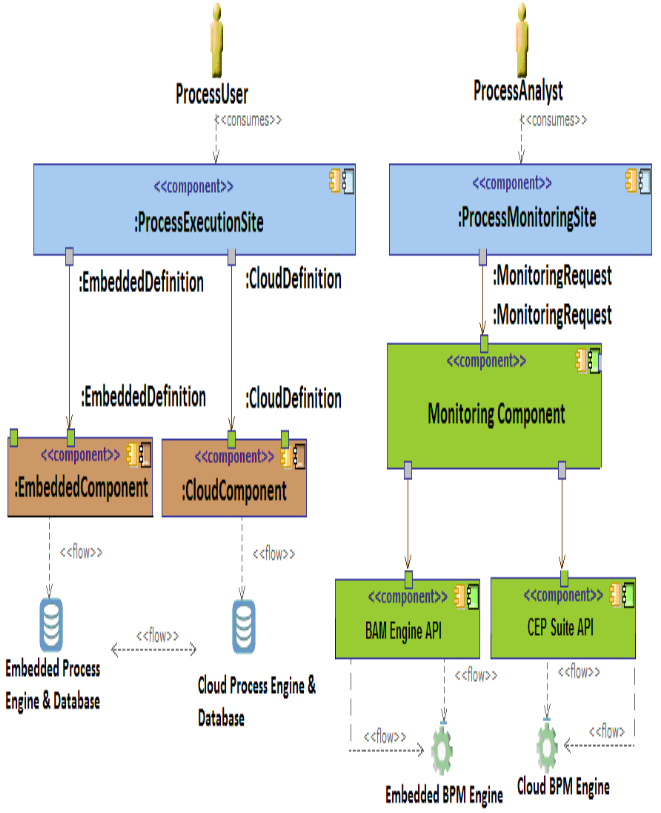

Fig. 6: Application communication diagram using CEP and BAM

In Fig. 6 the communication between the different components and the actors is illustrated. This particular aspect of the architecture remains mostly invariable as in the previous contributions, except for the inclusion of the BAM and CEP suites' API in order to obtain the results from the different portions of the process and merge them seamlessly according to the original process perspective [1] [2] [29].

5. INDICATORS: BATCH AND SPEED LAYERS IN DISTRIBUTED BPM

There are several differences between the results thrown by BAM and CEP. Even though both participate in the indicator making process and provide relevant information for the decision making process, the nature of the incomes and the outcomes of each one tends to be quite different.

In the case of BAM, the incomes proceed exclusively from the BPM engines distributed all over the architecture. The data recollected is absolutely dependent on the decomposed processes. On the other hand, CEP monitors events in general, and in particular they could belong to business processes or, instead of that, even proceed from different applications deployed in the cloud, related to the business also, but not specifically to the BPM engines [5] [10] [30] [31].

In terms of indicators, the data monitored using BAM in general are more static compared with events, especially those which could be considered complex. The period of latency necessary in order to provide a reliable indicator is major compared with the event log considered by CEP. There are several standard indicators for BAM developed for traditional BPM and they could be extrapolated for distributed BPM:

- Average execution time: it could be considered for each part of the decomposed process, or a summarization in case of the global process definition.
- Finalization percentage: determines the process' participation inside the organization. It considers the amount of ended processes divided by the total amount of them. In terms of decomposition this indicator could be expressed for each individual node, or globally. To consider the second choice it is necessary to consider the amount of ended instances grouped by process definition, and then proceed to make the calculation.
- Failed tasks percentage: this indicator shows the percentage of failed tasks grouped by process definition. It is useful to see the most problematic tasks in the environment. In terms of distributed BPM, the instances should be grouped firstly by the original process definition, and after this make the final calculation.
- Started instances: it shows the started cases divided by process definition, and it could consider also process versions. In terms of distributed BPM the instances should by grouped firstly by the original process definition, and after this make the final calculation.
- Activation percentage: shows the relation between the total amount of cases activated by process definition and the global amount of instances. Once again, in a distributed environment, cases should be grouped first.
- Accumulated percentage: results from dividing the accumulated process activations and the started cases of all processes. The grouping tasks should be done once again in a distributed environment.

In this way it is possible to see how the standard BAM indicators extracted from traditional BPM are capable of being extrapolated to distributed BPM, in terms of some extra calculations introduced by the fact of using process decomposition [4] [5] [32] [33].

CEP also is very important in terms of indicators for the decision making process, but the data's nature considered by this discipline is much more dynamic than the one considered in BAM. Also the latency period in CEP is shorter than in BAM; several CEP indicators are often calculated in real time. There are also several standard indicators for CEP applied to traditional BPM. They are also applicable to distributed BPM and mostly take data from cases currently in execution. Examples of these indicators are:

- Active cases' execution time: this indicator shows the elapsed time and the pending time for the deadline applied to active instances. The indicator also shows an alert in case the deadline has been reached. In terms of distributed BPM it is important to merge the source events for all the intermediate instances in order to show a global result with the original process perspective.

- Failed tasks control: shows in real time the failed tasks with some control information. On several occasions this indicator results useful to detect the status of a certain case.

This list is not restricted; any kind of new indicators could be developed considering the real time events emerged from a BPM system. This type of indicators results very useful during the design and implementation phase in terms of quick modification and bug fixing. Considering the different layers composing the business infrastructure, BAM corresponds to the batch layer and CEP to speed layer.

	BAM	CEP
Incomes' nature	Data considered for BAM indicators is mostly related to ended cases. Since the discipline is related to processes exclusively, the source is the BPMS's log.	Data considered for CEP indicators proceed mostly from active cases and real time events. These events could belong both to BPM engines and traditional applications deployed in the cloud environment.
Indicator's type	Static.	Dynamic.
Latency window	Latency is wide. The processing type is associated with the batch mode.	Is very narrow. The real time factor is essential in terms of efficiency and error correction.

Table I: Comparison between BAM and CEP

In Table I we make a very brief comparison of BAM and CEP. It can be also applied both to traditional or distributed BPM.

These concepts correspond to traditional BPM, but are the same in the case of distributed BPM. Analyzing the BAM indicators, the information involved is mostly related to ended cases, so the latency window is wide [10] [5] [34] [35].

The type of processing needed by BAM indicators is batch; results are not shown in real time. In opposition to this, CEP indicators use information of current running activities and cases. The results needed are related to the efficiency and error correction. This information must be released in real time and processed on the fly. That is the main reason why BAM and CEP are complementary in terms of the type of indicators managed. If an organization only uses BAM, it takes the risk of making delayed decisions, because of the batch feature. In opposition to this, if an organization only uses CEP it will lack all that information useful to know the characteristics of the ended cases and to correct some wrong behaviors [36] [37] [38].

6. CONCLUSIONS

The irruption of BPM in the cloud has caused several changes in business process execution. These changes have been studied deeply in current bibliography, especially when it comes to process execution in a decomposed environment involving private or public nodes. One aspect poorly explored by the current bibliography is the monitoring of decomposed processes. In previous contributions we proposed architectures and concrete implementations for them using a BPMS currently present in the industry (BonitaOS) [35] and a monitoring application in charge of collecting data from each distributed node using web services.

Some concepts already known from traditional BPM environments, like BAM and CEP are considered in the present work as mechanisms to improve the monitoring application. In order to make this, it was necessary to detect the substantial differences between both disciplines, in terms of incomes, results and nature. The final result is the architecture from previous works now arranged according to the new technologies, and a set of traditional indicators used in dashboards modified under the perspective of distributed BPM.

In terms of future work some lines can be expressed. There are several tools in the present market for the implementation of BAM and CEP, like WSO2BAM and WSO2CEP [39]. They can be deployed in a cloud environment and be integrated with several open source tools, like Bonita Open Solution. A future proposal regarding this is to adapt the concrete implementation of distributed business processes made before in several works to these tools, and enhance in this way the monitoring application by adding some indicators properly managed in BAM and CEP traditionally.

REFERENCES

[1] N. Herzberg, A. Meyer, M. Weske. "An Event Processing Platform for Business Process Management". Business Process Technology Group, Hasso Plattner Institute at the University of Potsdam. Potsdam, Germany. June 2013.

[2] S. Bulow, M. Backmann, N. Herzberg, T. Hille, A. Meyer, B. Ulm, T. Y. Wong, M. Weske. "Monitoring of Business Processes with Complex Event Processing". Business Process Technology Group, Hasso Plattner Institute at the University of Potsdam. Potsdam, Germany. July 2013

[3] C. Zeginis, K. Kritikos, P. Garefalakis, K. Konsolaki, K. Magoutis and D Plexousakis. "Towards Cross-Layer Monitoring of Multi-Cloud Service-Based Applications". Institute of Computer Science Foundation for Research & Technology – Hellas. Grece. August 2013.

[4] M. Goetz. "Integration of Business Process Management and Complex Event Processing". iTransparent GmbH, IT Consulting, Bergstraße 5, 90403 Nuremberg, Germany. November 2010.

[5] J. Kollar. “Business Activity Monitoring”. MASARYK UNIVERSITY FACULTY OF INFORMATICS. July 2010.

[6] J Martinez Garro, P Bazán. “Monitoreo de procesos en el cloud. Una propuesta arquitectónica”. JCC 2013. Universidad de Temuco. Chile. November 2013.

[7] J Martinez Garro, P Bazán. “Constructing and monitoring processes in BPM using hybrid architectures”. IJACSA Journal. Londres. Febrero 2014.

[8] J Martinez Garro, P Bazán, J Diaz. “Decomposed processes in Cloud BPM: techniques for monitoring and the use of OLC”. WORLD COMP 2014. Las Vegas, USA, July 2014.

[9] J Martinez Garro, P Bazán, J Diaz. “OLC y Monitoreo de procesos en el cloud: un caso de estudio”. JCC 2014. Chile. November 2014.

[10] WebMethods, “Business Activity Monitoring (BAM) The New Face of BPM”. WebMethods Inc. June 2006.

[11] D W. McCoy. “The Convergence of BPM and BAM”. Gartner. SPA-20-6074. January 2004.

[12] R Confortia, M La Rosaa, G Fortinoc, A H. M. ter Hofstedea, J Reckera, M Adamsa. “Real-Time Risk Monitoring in Business Processes: A Sensor-based Approach”. Queensland University of Technology, Brisbane, Australia. May 2013.

[13] E Mulo, U Zdun, S Dustdar. “Domain-Specific Language for Event-based Compliance Monitoring in Process-driven SOAs”. Distributed Systems Group Institute of Information Systems Vienna University of Technology, Vienna, Austria. April 2013

[14] P Szwed, W Chmiel, S Jedruzik, P Kadluczka, “Business Process in a Distributed Surveillance System integrated through workflow”. Automatika. Vol 17. No 1. November 2013.

[15] V Stavrou, M Kandias, G Karoulas, D Gritzalis. “Business Process Modeling for Insider Threat Monitoring and Handling”. Information Security & Critical Infrastructure Protection Laboratory Dept. of Informatics, Athens University of Economics & Business. Greece. May 2014

[16] C Cabanillas, A Baumgrass, J Mendling, P Rogetzer, B Bellovoda. “Towards the Enhancement of Business Process Monitoring for Complex Logistics Chains”. Institute for Information Business at Vienna University of Economics and Business, Austria. July 2013.

[17] K Jander, W Lamersdorf, “Jadex WfMS: Distributed Workflow Management for Private Clouds”. Distributed Systems and Information Systems University of Hamburg, Hamburg, Germany. January 2013.

[18] P Goyal,R Mikkilineni. “Implementing Managed Loosely-coupled Distributed Business Processes: A New Approach using DIME Networks”. Kawa Objects Inc. Los Altos, USA. July 2012.

[19] Z Fang, C Yin, “Intelligent Information Management” (http://www.SciRP.org/jou rnal/iim). IIM “BPM Architecture Design Based on Cloud Computing”. May 2010.

[20] G Srdić, M B. Jurič. “BPM and iBPMS in the Cloud Aleš Frece”. Proceedings de la Conferencia Internacional de Cloud Assisted ServiceS Bled. October 2012.

[21] L Ferreira Pires, L O. Bonino da Silva Santos. “Towards a BPM Cloud Architecture with Data and Activity Distribution”. IEEE 16 Conferencia Internacional de Computación sobre objetos distribuidos.. Evert F. Duipmans. May 2012.

[22] N Herzberg, A Meyer. “Improving Process Monitoring and Progress Prediction with Data State Transition Events”. En ZEUS. 2013. p. 20-23. Plattner Institute at Potsdam University. May 2013.

[23] N Herzberg, A Meyer, O Khovalko, M Weske. “Improving Business Process Intelligence with Object State Transition Events”. En Conceptual Modeling. Springer Berlin Heidelberg, 2013. p. 146-160. Plattner Institute at Potsdam University. July 2013.

[24] M Mevius, R. Stephan, P. Wiedmann, "Innovative Approach for Agile BPM", eKNOW 2013: Quinta Conferencia internacional en Información, Procesos y Gestión del Conocimiento. July 2013.

[25] H Sakai, K Amasaka. "Creating a Business Process Monitoring System “A-IOMS” for Software Development". Chinese Business Review, ISSN 1537-1506, Vol. 11, No. 6, 588-595. June 2012

[26] 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: Tercera Conferencia Internacional en Cloud Computing, GRIDs, y Virtualización. July 2012.

[27] E Duipmans, Dr. L Ferreira Pires, "Business Process Management in the cloud: Business Process as a Service (BPaaS)". Trabajo de finalización de especialización. University of Twente, April 2012.

[28] JP Friedenstab, C Janieschy, M Matzner, O Mullerz. "Extending BPMN for Business Activity Monitoring". 2012 45th Hawaii International Conference. University of Liechtenstein, Hilti Chair of Business Process Management, Vaduz, Liechtenstein. September 2011.

[29] M Reichert, J Kolb, R Bobrik, T Bauer. "Enabling Personalized Visualization of Large Business Processes through Parameterizable Views". 2011 44th Hawaii International Conference. Hochschule Neu-Ulm, Neu-Ulm, Germany. September 2011.

[30] J Kolar, T Pitner, "Agile BPM in the age of Cloud technologies”, Scalable Computing: Practice and Experience Journal, April 2012.

[31] A Lehmann and D Fahland , "Information Flow Security for Business Process Models - just one click away", En BPM (Demos). 2012. p. 34-39. University of Rostock, Germany, April 2012.

[32] R Accorsi, T Stocker, G Müller, "On the Exploitation of Process Mining for Security Audits: The Process Discovery Case". En Proceedings of the 28th Annual ACM Symposium on Applied Computing. ACM, 2013. p. 1462-1468. Department of Telematics, University of Freiburg, Germany, June 2012.

[33] A Frece, G Srdić, MB. Jurič, "BPM and iBPMS in the Cloud", En Proceedings of the 1st International Conference on Cloud Assisted ServiceS, Bled, Octubre 2012.

[34] S Zugal, J Pinggera and B Weber. "Toward enhanced life-cycle support for declarative processes". Journal of Software: EVOLUTION. Marzo 2012

[35] Bonita Open Solution http://es.bonitasoft.com/. March 2015.

[36] D Müller, M. Reichert, J Herbst (2007). “Data-driven modeling and coordination of large process structures”. De On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS (pp. 131-149). Springer Berlin Heidelberg. August 2007

[37] D Müller, M Reichert, J Herbst, (2006, January). “Flexibility of data-driven process structures”. De Business Process Management Workshops (pp. 181-192). Springer Berlin Heidelberg. October 2007.

[38] L. Karabagosian, J. Martinez Garro, P. Bazan, “Ejecución y monitoreo de procesos de negocios distribuidos entre diferentes motores de Bonita OS”. XVI Workshop de Investigadores en Ciencia de la Computación 2014 : WICC 2014. Usuahia, Tierra del Fuego, Argentina. May 2014.

[39] WSOS Products. WSO2.com. March 2015.