Business Activity Management for Service Networks in Cloud Environments

Christian Janiesch
Karlsruhe Institute of Technology (KIT)
76131 Karlsruhe, Germany
+49 721 608 45770
christian.janiesch @kit.edu

Robin Fischer
Forschungszentrum Informatik
10117 Berlin, Germany
+49 30 7017337 332
robin.fischer @fzi.de

Martin Matzner
University of Münster ERCIS
48149 Münster, Germany
+49 251 833 8088
martin.matzner @ercis.de

Oliver Müller
University of Liechtenstein
9490 Vaduz, Liechtenstein
+423 265 13 23
oliver.mueller @uni.li

ABSTRACT

Companies struggle to find ways to manage intra- and interorganizational service networks communicating in a distributed fashion across the globe. We review the state-of-the-art of managing choreographed service networks and put it in relation to process analytics and complex event processing (CEP) against the background of Cloud computing. We present an initial architecture for Event-driven Business Activity Management of service networks which also takes into consideration levels of virtualization. The architecture can serve as a blueprint for flexible business activity monitoring applications as well as closed loop service choreography control solutions. We illustrate the interaction of Cloud infrastructure, services networks, and CEP systems with a number of use cases. In addition, we discuss future research directions based on our experiences from early prototypes.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems – *Decision support (e.g., MIS)*

General Terms

Management, Measurement, Performance, Design.

Keywords

Business process management, complex event processing, business activity monitoring, service networks, architecture.

1. INTRODUCTION

With the advent of Cloud computing and the ongoing shift towards on-demand resource or service consumption, business processes execution becomes increasingly distributed. In consequence, management and measurement of these business intra- and interorganizational choreographies may no longer be based on centralized execution engines, but requires new approaches that address the distributed nature of service networks running in Cloud environments. The application of various measurement and analysis techniques on process-related data has been proposed under umbrella terms such as business process intelligence, business process mining, or business activity monitoring. The aim of these approaches is to provide process owners and participants with evidence-based insights about the performance of operational business processes [1, 2]. As of today, most business process management (BPM) systems provide a centralized log to track the execution of their business processes. The data contained in these logs represents a rich source of information to gain insight and act upon executed processes to improve their performance.

In the context of service networks, however, there is not one BPM system, which executes the process, but a plethora of different systems which execute a choreography of services based on Web technology (e.g., with SOAP/ WSDL): a service network. These service networks form the execution environment for cross-organizational business processes. Each services on the other hand requires resources, which – in the context of Cloud computing – are mostly consumed from virtual resources provided by third-parties as software (SaaS), platform (PaaS) or infrastructure services (IaaS).

Analyses required to manage these networks are complex and costly if possible at all. Furthermore, most existing software solutions are bound to one BPM system and either lack sophisticated capabilities to analyze log data or are limited to pure monitoring functions [2-4]. The main challenge in this distributed scenario is to be able to collect data from the diverse services or service choreographies within the network and return a report or dashboard, which includes all available sources to form a unified picture of the service network's performance. In a Cloud environment, this does not only include the performance of processes and their services but also the performance of the virtualized hardware supporting remote service choreographies.

We propose to apply the concept of Complex Event Processing (CEP) to service networks to allow for a monitoring and active control of the executed choreographies. CEP, in general, comprises a set of techniques for making sense of the behavior of a monitored systems by deriving higher-level knowledge from lower-level system events in a timely and online fashion [5-7]. In the following we will present a proposal for an architecture in which Cloud-based service networks and CEP systems are integrated in a closed control or feedback loop through the exchange of (complex) events. Besides realizing a monitoring and control loop, the proposed event-driven architecture offers further advantages such as real-time processing or loose coupling.

The structure of this paper is as follows: In the subsequent section (Section 2), we present a review of related work on approaches process analytics, which are relevant for service networks, Cloud-based service execution management, and event-driven architectures based on CEP. We describe the details of the architecture in Section 3 and outline different interaction options between service networks and CEP systems. In Section 4, we discuss use cases, which cover applications from mere monitoring to closed control loops. We close in Section 5 with a discussion of limitations and areas for future research.

2. RELATED WORK

2.1 Process Analytics

Davenport and Harris define the term analytics as the "extensive use of data, statistical and quantitative analysis, explanatory and predictive models, and fact-based management to drive decisions and actions" [1]. Analytics may be both an input for human decisions and a driver for fully automated decisions. In Cloud computing, real-time feedback is even more important to guarantee availability and maintain visibility of virtualized resources. Still, the following applies also to analyzing individual services or choreographies and their resources as part of an overall process.

Following zur Mühlen and Shapiro [2], we distinguish between three types of analytic applications in the context of processes:

Process Controlling and Process Mining. Historical process data can be found in log files of BPM and other service-based information systems [2]. This data can be extracted, transformed, and loaded into an external data warehouse to allow for a detailed analysis. Typically, this data is analyzed on an aggregated level and from an external perspective (focusing on inputs and outputs of services, rather than the structure and behavior of a choreography) [8]. Process mining, on the other hand, strives for the "automatic construction of models explaining the behavior observed in the event log" [9]. Its focus is on concurrent processes rather than on static or mainly sequential structures [8]. To its very nature, process mining is an ex-post analysis of process behavior.

Business Activity Monitoring. Business activity monitoring (BAM) focuses on capturing and processing events with minimum latency, i.e. near real-time [10]. Therefore, such systems are usually tightly integrated with the operational data sources, e.g. BPM engines. BAM foremost strives for transforming process-related events into key performance indicators (KPI) and for detecting changes or trends indicating opportunities or problems which require managers to take proactive or corrective actions [10]. A BAM system must be able to detect process-relevant events, to calculate information for temporal processing, to integrate event and contextual business information on the fly, to execute business rules to set thresholds according to KPIs and other business-specific triggers, and to provide intuitive interfaces for presenting the results [10, 11].

Process Intelligence. The objective of (predictive) process analytics is to assess the impact of changes to process design on the performance of the process [2]. Simulation is used either before a process is deployed for execution or after a process has been executed for a reasonable amount of time. In the former case, simulation can assist in assessing the adequacy of the process design for the designated task. In the later case, variants of a process design can be used for simulation in order to improve or optimize the process design. In both cases, simulation requires a mathematical model of the process at hand, including e.g. data on the availability of resources and probability distributions for the occurrence of events and the duration of activities. Time series analysis and forecasting, on the other hand, offer insight into the longer term horizon of execution and can help identify future bottlenecks in processes or their required resources.

2.2 Service Choreographies in Cloud Environments

Traditionally, BPM and IT Service Management (ITSM) provide approaches towards managing process and system performance. While the focus of BPM is towards managing the overall performance of processes (e.g., improve cycle times, remove bottlenecks, optimize resource allocation, etc.), ITSM is more focused towards the management of organizational IT resources. BAM allows integrating disparate resources (e.g. process engines and ITSM systems), thus enables monitoring of business activities along with required (IT) resources within an organization.

Existing BAM approaches are based on two general assumptions which do not hold in Cloud execution environments: First, BPM (and thus BAM) follows a life-cycle approach that starts with process design, continues through implementation, enactment, and monitoring and closes the loop with process analysis [12]. In consequence, process models are a central prerequisite for BAM efforts. Second, traditional BPM over-emphasize the focus on intra-organizational processes towards organizational operational efficiency. In doing so, a central control or management authority, having exclusive power to reorganize choreographies or reallocate resources is generally assumed. Aspects of autonomy, uncertainty or loss of control (through consumption of remote services or resources) are rarely focused, yet. Consequently, there exists a plethora of tools that support BPM based on these assumptions, i.e. tools that support BPM for intra-organizational process or processes with assigned owners. In contrast, only few works exists that particularly address the needs of managing process execution over organizational boundaries, today:

2.3 Event-driven Architectures and Complex Event Processing

The concept of event-driven architecture (EDA) is a recent addition to the list of architecture paradigms for system landscapes. It emphasizes the orchestration of applications and processes through events which can originate anywhere from external sensors to internal or external business IT systems [7]. The core idea is to enable systems to anticipate and react to normal and abnormal event types and adapt the execution of underlying business processes.

Due to the high volume of simple events, events may need to be correlated and aggregated to complex events to provide meaningful insight. This requires scalable processing technology such as CEP. CEP in general, comprises this set of techniques for making sense of the behavior of a system by deriving higher-level knowledge from lower-level system events in a timely and online fashion [5-7].

The central concept is the event. An event is defined as a data object that is a record of an activity that has happened, or is thought of happening in a system [7]. The event signifies the activity. Such objects comprise, among others, activities, actors, data elements, applications, or entire processes. Another key idea of CEP is the aggregation of events. Through the definition of abstraction relationships it is possible to define complex events by aggregating a set of low-level events. Hence, a complex event signifies a complex activity which consists of all the activities that the aggregation of individual events signified [7]. Orthogonal to the concept of aggregation is the concept of causality. Low-level events might happen in dependence of each other [7]. Causality plays an important role in any kind of root cause analysis, both online and ex-post.

3. EVENT-DRIVEN BUSINESS ACTIVITY MANAGEMENT FOR SERVICE NETWORKS

Event-based or event-driven Business Intelligence (BI) approaches have emerged. These systems are mostly referred to as operational BI. While they shorten decision latency and overcome the lack of event support in traditional BI systems, event-driven BI falls short when it comes to the dynamic modeling and use of business rules to define actions upon event occurrences and especially in correlating events into metrics over time [10]. They fall short of real event-driven architectures as these are considered to be an addition to or an augmentation of but not a replacement for service-oriented architectures (SOA). EDA can not only used for the federation of events for alerting and notification purposes but for complex interactions with the associated systems.

In the context of Cloud-based service networks, the abilities to define business rules dynamically and correlate events from a variety of resources are crucial. Events in service networks root from (geographically) distributed IT-systems (e.g. database and middleware in the U.S., application server in Europe) which are out-of-control for consumers and potentially federated amongst multiple Clouds. Consequently, it is difficult or even impossible for involved actors to apply direct control over all resources. In order to counter the threat in Cloud computing of loss of control and re-gain control over remote resources, new methods, architectures and tools are required that support BAM for service networks.

Here, CEP emerges as a valuable instrument for a timely analysis of process and service network-related events. The interaction of CEP and service network can be bi-directional, as other systems can act as (a) a producer and as (b) a consumer of events [6].

In the following, we describe how CEP can be applied to monitor and control service networks which run in Cloud environments in real-time. We extend upon the real-time concept of BAM by leveraging the capabilities of CEP to include decision-making and feedback mechanisms to the originating service network and other systems. This effectively means that the CEP engine, in parts, manages rather than observes the execution of the service networks. Hence, we also speak of *Business Activity Management* rather than of monitoring.

Figure 1 depicts the proposed Business Activity Management architecture for service networks. The architecture is based on the typical tri-partition in CEP. In most CEP applications, there is a distinction between the entities that input events into the actual CEP system (event producers), the entities that process the events (event processors), and the entities that receive the events after processing (event consumers).

Event Producer. The major event producers in service network scenarios are individual systems (application servers) in a service network providing orchestrated or atomic services such as BPM engines (BPE) or enterprise application integration (EAI) platforms. In Cloud-based service networks process execution is distributed amongst independent service providers having varying levels of autonomy and potentially a multitude of BPE or EAI components. Moreover, it is very likely, that there is no central management instance but choreographies may start in one system, transition to other systems and may or may not return to the first system. For a consolidation of the occurring events into events that stem from corresponding executions of a service network, the following components are of main interest:

Typically, every BPM and EAI engine provides an environment for the execution and interaction of service instances, based on existing service choreography definitions. Theses system’s logging components offer functions to track the progress of execution and log corresponding audit data including data on occurred state transitions of choreography-relevant objects (e.g. processes, services, resources, work items). This audit data constitutes a rich source of service execution events for subsequent analysis, but is usually not accessible from outside the individual execution engine. Hence, an event publisher – output adapter – is required to make the data available for participating parties in a service network. The main functions of this output adapter is sniffing, formatting, and sending of events [7].

In the sniffing step, the adapter detects events in the local audit data (service orchestration) and other relevant sources (e.g. work item lists or Cloud resources). Typically, the sniffer maintains a list of filters that defines which events should be observed and subsequently extracted to prevent sensible data from being published or to simply reduce the volume of events. Such filters can be defined on type level or instance level. It is important that event sniffing must be benign, i.e. it must not change the service network’s behavior. After detecting an event, the corresponding data has to be extracted and transformed to a common event format.

However, not all system events can be easily mapped into a standard format. BPAF [13], for example, deals only with state changes in local processes on the activity level. Execution-related events such as lower-level events of consumed services (e.g., SaaS, PaaS, IaaS) or even related system level events of required resources are not considered. Efficient or even standardized monitoring of Cloud resources is still in its infancy. As of now, only basic health information on cloud resources is made available for service consumers. Amazon’s CloudWatch API, for example, allows for the monitoring of basic information like minimum, maximum or average CPU load, memory usage or request latency through a standardized API. Correlation of resource events to service network events (or vice-versa) is not supported as of now, as existing Cloud resource management services do not incorporate the notion of a processes or service choreographies. Through the application of our concept of event output adapters, existing cloud resources can be integrated into our event-driven Business Activity Management infrastructure allowing for an aggregation of events towards higher-level network events like potential performance slowdown as system metrics reach thresholds.

Besides BPM and Cloud infrastructure systems, there are a number of other systems that can act as event sources in the context of a service networks. Sensors, for instance, can produce events that signify the actual execution of a physical service activity or report on real-world events that are happening in the context of the execution (e.g. weather change, traffic conditions, position and movement of objects). Transactional systems, such as

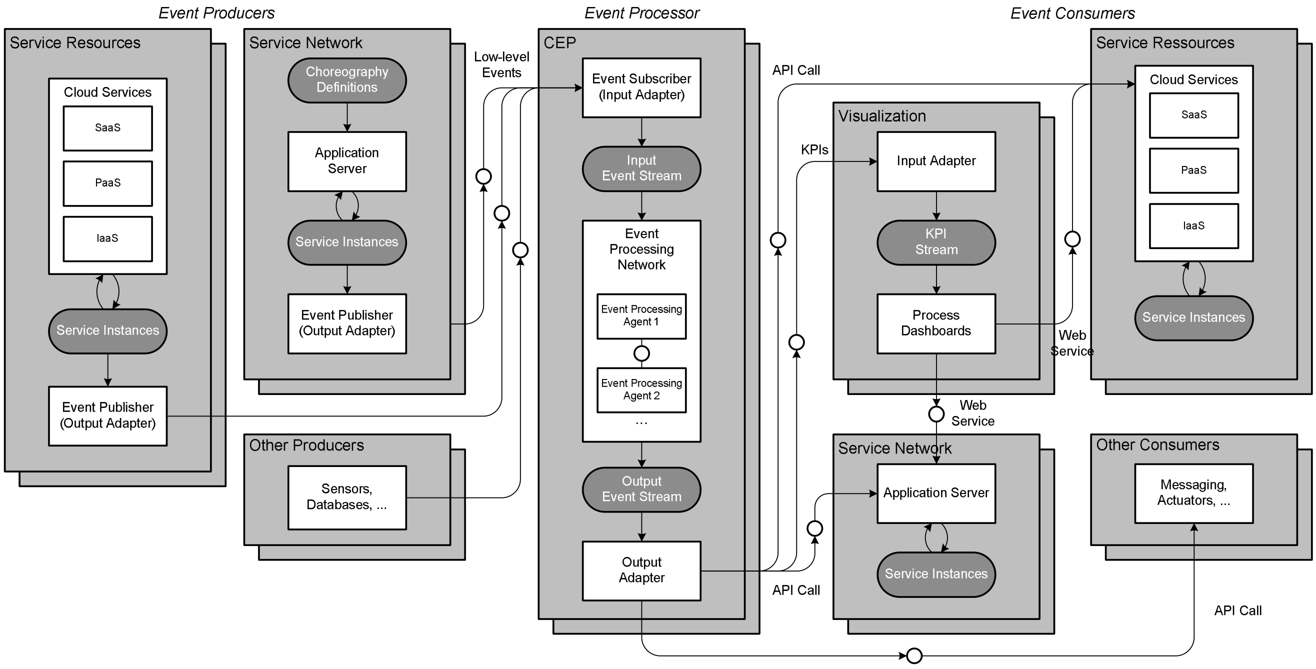

Figure 1. Architecture for Event-driven Business Activity Management of Service Networks.

ERP or CRM systems, can produce events that are related to a choreographies but lie outside of the control sphere of a service network (e.g. low stock alert, creation of a master data record).

After an event has been detected and formatted, a corresponding message has to be sent to the CEP system. Communication can be done in a direct point-to-point fashion or via message-oriented middleware (for reasons of complexity not depicted in Figure 2). Either way, the use of a publish/ subscribe messaging mechanism is a promising way to realize loosely coupled and scalable communication as required in Cloud environments.

Event Processor. The raw events sent out by the various event producers are received and processed by an event processor, i.e. the actual CEP system. Positioning CEP as the central component in this event-driven measurement and management architecture is crucial.

CEP systems are typically deployed in a centralized fashion, however, there are early versions of CEP engines which can be hosted in the cloud or in a dynamic distributed fashion. In any case, queries can be staged and distributed so that the risk of a single point of failure within the service network can be minimized. Through virtualization, placement of the event processor (in any of its occurrences illustrated below) in the Cloud can reduce its susceptibility. However, bandwidth can become a problem as event traffic is of high volume. Cf. Figure 2 for staging options from service networks via CEP engines to a dashboard for visualization. Option 1 depicts the simplest version of BAM for service networks where all parties signify events to one CEP engine. Option 2 is a more sophisticated setup of staged CEP engine which communicate with each other to load balance and avoid a single point of failure. The setup may – however – be transparent to the user. More recently dynamic Cloud CEP solutions have been proposed which relieve the user from manually separating queries solely for performance reasons into multiple models.

Events are processes in an event-processing network as laid out in Section 2.3. After processing, events are queued into an output event stream, ready to be forwarded to possible event consumers. An event output adapter is responsible for transforming the outgoing events into external formats that can be processed by the connected event consumers. After adaptation, the outgoing data does not necessarily have to represent as an event anymore; it can, for example, be in the form of KPIs, messages, or remote function calls (see below).

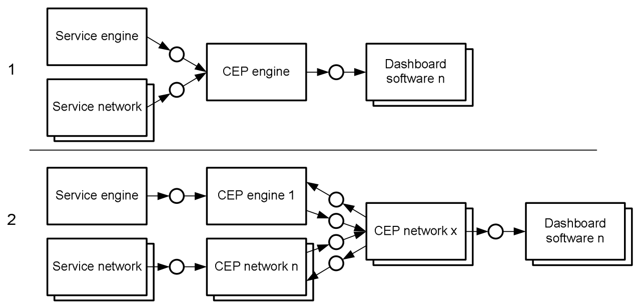

Figure 2. Conceptualization of Service Networks in Cloud Environments.

Event Consumer. An event consumer is an entity that receives and visualizes or acts upon events from a CEP system. It is the logical counterpart of the event producer. As depicted in Figure 1, dashboards, messaging services, and choreography participants of the service networks are the main event consumers in this scenario.

Dashboards provide high-level summaries of information in a concise, clear, and intuitive way and often in real-time [14]. This real-time aspect makes dashboards especially appealing in context of Event-driven Business Activity Management. Typical information to be displayed in a process dashboard include KPIs (e.g. runtime, idle time), occurred or anticipated exceptions (e.g. when thresholds of KPI will likely be exceeded), or visualizations of the actual flow/ path of service instances. The visualized information is often averaged on the basis of the last *n* events of an event output stream or on the basis of a certain time window.

Similarly, messaging services can be leveraged to send alert messages to inform service or infrastructure owners about exceptions or declining KPIs. A CEP engine can basically use all available channels of communication such as mobile phone short messaging services (SMS), instant messaging, e-mail, or mass messaging services such as Twitter to actuate reactions.

Apart from being an event producer, service providers in services networks also represent a key event consumer in the architecture. Based on the insights derived from processing events produced by a service network, an event processor might initiate actions that influence services in that very network (or other networks respectively).

An event output adapter of a CEP system might, for instance, initiate new service choreographies. However, as of today, most BPE can only be called via proprietary Web service or through proprietary SOAP or REST message.

Likewise, manipulating or updating the conditions of a choreography execution is next to impossible today, although corresponding resource allocation might have major impact and change decisions which control the flow of a choreography instance [15].

An API call to allocate more resources for a particular service (e.g., in case of dropped availability or responsiveness) or suspend, resume, or cancel running service instances (which could be anything from a single service to a service choreography) could be the increase of computational power (e.g., add another virtual machine to an existing service hosted on AWS EC2).

4. USE CASES

The architecture introduced in the previous section facilitates various analytics use cases for managing service networks. These range from mere real-time monitoring of services and virtualized hardware to orchestrating service networks in an event-driven fashion and closed loop automated insight to action. So, on the one hand it can simply be used to replace custom built legacy applications for service monitoring with a standardized backend in order to make them more maintainable. On the other hand, it can be used to facilitate an even bigger picture and include multiple input event sources, multiple output options, and multiple processing alternatives.

The architecture is flexible enough to not only serve one service type or BPM system, but can work with a whole service network. In addition, real-time information on the performance of virtualized hardware can be factored in the equation. Moreover, multiple CEP systems can be used in order to distribute work due to performance or for security reasons. Furthermore, the event processing (as well as service execution) can take place in the Cloud and escalation adaptations can take place on both, the level of resources and business logic.

In the following, we elaborate on a selected number of use cases for this architecture. Some of the cases have been dealt with individually already but so far there has not been a proposal for a unifying architecture as we introduced it above. Consequently, vendors have only made a fraction of the functionality available so far.

Event Aggregation to Live Dashboards of Service Networks. As more and more information relevant to operating a business and making decisions is available in near real-time, the desire to populate operational dashboards instantly rises. As a consequence, critical decisions can be made using the actual data instead of the static data produced by batch jobs.

In a service network or even within a single company, multiple services and BPM systems exchange data with each other and form an overall process flow. Sensor data or resource data from virtualized hardware can further add information. Technically, it does not matter to the proposed architecture which systems are connected as long as they can either produce events or incorporate services which handle event sending for them. Furthermore, the dashboard and event processing may reside in the Cloud to scale with the current size of the event streams and to provide the dashboard frontend online at any time. Future work will focus on introducing Cloud-based dashboard services.

Common use cases involve a supply chain or governance process moderator, who is unable to see the current state of a service choreography if it is not within his control. The architecture enables services to publish status events to the event processor which can be aggregated and plotted according to an overall choreography definition so that a better overview is achieved. This of course entails, that the choreography models are at least known at a high level or standardized operations are performed and propagated.

Explorative and Predictive Process Analysis. Data processing in an event processing agent does not need to be restricted to filtering, transformation, and static pattern detection through time-based correlation. It is possible to execute a broad variety of analyses on the stream of data. Once data is pushed to an event processing agent, explorative data analysis such as clustering, classification or association analysis can be implemented as well as simulation to predict imminent bottlenecks or excess capacities before they occur. In doing so, the architecture may contribute to the emerging field of Data-as-a-Service, providing a platform for data and data analysis in the Cloud.

As a benefit, unexpected spikes service networks transaction volumes demanding more computational resources for knowledge data discovery operations could still work on real-time events rather than logs to predict the amount and allow for an early verification of proper resourcing. In terms of clustering and classification these scenarios only add value to established batch processing of data if there is a clear requirement for early warnings and it is not possible to wait for the next day. Examples span from the association analysis of medical data to isolate infectious diseases quickly to the prediction of transport capacities for express goods or the prediction of resource need for virtualized hardware in order to avoid unnecessary lead times when adding or even lost opportunities.

This use cases can use events from the service network and sensors as well as from the underlying resources. It depends on the scenario if resource load or (the lack of) service activity would trigger a complex event. The same applies for the following which emphasizes the importance of auxiliary events and automated response channels to the respective consumers/ producers.

Context-aware Business Activity Management and Automated Insight to Action. Context-aware Business Activity Management makes service networks more flexible in relation to external events which change the context of the service execution. This can include weather (e.g. hurricane warnings) or traffic information (e.g. port or airport closures) for logistics use cases, currency exchange rates for financial use cases, or server/ network maintenance downtimes and failures in more technical use cases.

This entails that the interaction between the CEP and service networks is not only a unidirectional measurement but bidirectional communication. Only if a closed loop is established between the two, as in the above architecture, instant and automated insight to action is possible. This entails, that the event processor must not only be aware of the overall process (as in the first use case) but must also know the actors, which deliver the specific parts of it, or – at least – needs a repository where it can set global variables for the service network. A context engine could be such a global system, which collects data from different context event sources. It understands events in relation to the service choreography so that it can indicate to the service network to adapt service choreographies accordingly (e.g., allocate additional resources to counteract anticipated utilization spikes). A secondary CEP system can act as a basic context engine.

Accordingly, this architecture could be further developed to monitor system or even machine level events in Cloud environments and correlate these into information on service or system status. In the example, of an outage of infrastructure services, a context-aware BAM system for service networks could have invoked a network reconfiguration (e.g. exchange of services) to accommodate failure and maintain the choreography's operationality automatically based-on low-level system events.

5. CONCLUSIONS

Timely insight into a company's value creation is of great importance in order to be able to monitor and improve its operations. Services are the core building blocks of a company's process operation and provide a plethora of information that can be tapped into. In the past, custom applications have been developed for this purpose. Recently, the concept of CEP gained attention. However, its dedicated application to BPM and service networks is only in its infancy. Nevertheless, it promises cost saving due to the ability to react in a reasonable timeframe to errors, exceptions, or other indicators such as workload, waiting times, etc.

We reviewed the current state of the art of Cloud-based service execution, and analytics in BPM and CEP. Based on the existing work, we developed an architecture for Event-driven Business Activity Management. Following the tri-partition in CEP, we introduced an architecture that is capable of delivering blueprints for applications ranging from real-time service monitoring to context-aware adaptation of service networks and their resources. The architecture is a software independent conceptual model that can be instantiated with basically any combination BPM and CEP systems which are capable of communicating with each other.

We have implemented a proof of concept of this reference architecture using multiple SAP NetWeaver BPM servers as BPM engines, the Sybase Aleri Streaming Platform as a CEP engine, and SAP BusinessObjects Xcelsius Enterprise as a visualization frontend running in a private cloud environment. The software prototypes can be downloaded for trial and evaluation purposes and represents state-of-the-art BPM and CEP technology. The implementation is able to showcase the majority of the described use cases. Due to limitations in paper length we are not able to include details on the successful implementation of the architecture in this publication, but will make it available during the presentation. Next steps will focus on the sophistication of the proof-of-concept and its validation and the transfer of the implementation to other engines. In particular, we will create a comprehensive reference scenario which can showcase the abilities mentioned above in one integrated demonstration.

In addition to that, we observed several shortcomings which have to be worked on in the future. These include but are not limited to the lack of common format(s) for business process and resource events, the lack of a standardized set of logged events, and the lack of standardized operations to trigger individual services or services choreographies for automated insight to action. On the conceptual side, there is a clear need for an integrated or at least synchronized modeling paradigm for service networks and monitoring applications, on the one hand, and service networks and their interaction with event processing networks, on the other hand. It must be possible to specify KPIs for a service network which are linked to an executable CEP model underneath. Otherwise, the dependencies between the service network and the Business Activity Management system will create a significant management overhead.

6. REFERENCES

[1] T. Davenport and J. Harris. *Competing on Analytics*. Harvard Business School Press, Boston, MA, 2007.

[2] M. zur Muehlen and R. Shapiro. Business Process Analytics. In J. vom Brocke and M. Rosemann (eds.) *Handbook on Business Process Management: Strategic Alignment, Governance, People and Culture*, pages 137-157. Springer, Berlin, 2010.

[3] D. Ting-Yi Ho, Y. Jin, and R. Dwivedi: Business Process Management: A Research Overview and Analysis. In *15th Americas Conference on Information Systems (AMCIS)*, San Francisco, CA, 2009.

[4] M. Weske, W. M. P. van der Aalst, and H. M. V. Verbeek. Advances in Business Process Management. *Data & Knowledge Engineering*, 50(1):1-8, 2004.

[5] M. Eckert. *Complex Event Processing with XChange EQ: Language Design, Formal Semantics, and Incremental Evaluation for Querying Events*. LMU München: Faculty of Mathematics, München, 2008.

[6] O. Etzion and P. Niblett. *Event Processing in Action*. Manning Publications, Cincinnati, OH, 2010.

[7] D. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Professional, Boston, MA, 2002.

[8] H. M. W. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst: XES Tools. In *22nd International Conference on Advanced Information Systems Engineering (CAiSE), CEUR Workshop, vol. 592*, pages 1-8, Hammamet, 2010.

[9] W. M. P. van der Aalst, H. A. Reijers, and A. Weijters. Business Process Mining: An Industrial Application. *Information Systems*, 32(5):713-732, 2007.

[10] D. Nesamoney. BAM: Event-driven Business Intelligence for the Real-Time Enterprise. *DM Review*, 14(3):38-40, 2004.

[11] C. Costello and O. Molloy: Towards a Semantic Framework for Business Activity Monitoring and Management. In *AAAI 2008 Spring Symposium*, pages 1-11. AAAI Press, Stanford, CA, 2008.

[12] M. Weske. *Business Process Management*. Springer, Berlin, 2007.

[13] Workflow Management Coalition (WFMC). *Business Process Analytics Format Draft Version 2.0*. 2008, http://www.wfmc.org/index.php?option=com_docman&task=doc_download&gid=437&Itemid=72.

[14] S. Few. *Dashboard Confusion*. 2004, http://intelligent-enterprise.informationweek.com/showArticle.jhtml;jsessionid=BJXFCGOZCMUE3QE1GHPCKHWATMY32JVN?articleID=18300136.

[15] J. Becker, M. Matzner, O. Müller, and M. Walter: A Review of Event Formats as Enablers of Event-driven BPM. In *5th International Workshop on Event-driven Business Process Management (EdBPM)*, pages 41-52, Clermont-Ferrand, 2011.