



---

# OpenESB Monitoring Framework

## Overview

Document version: 1.4

01 June 2017

Please send feedback to [contact@pymma.com](mailto:contact@pymma.com)

## **ABOUT PYMMA CONSULTING**

---

Pymma Consulting is a technical architect bureau. Pymma provides expertise in the conception and execution of integration projects and programs. Working with top 500's company and governmental establishment for many years, Pymma got a unique skill in the integration and service marketplace. Proactive in the open source landscape, Pymma support and sponsor open source projects such as OpenESB, Drools, and Gravitee.io. Pymma was founded in 1999 and is a European company based in the UK (London), Netherland, France and Canada.  
[\(contact@pymma.com\)](mailto:contact@pymma.com) or [\(www.pymma.com\)](http://www.pymma.com)

## **About the Author**

---

## **Contents**

What is OpenESB monitoring framework? .....	4
Why an OpenESB monitoring framework? .....	4
OpenESB monitoring framework installation .....	5
BPEL Events .....	5
Business messages .....	6
Event publisher .....	7
Business message publisher .....	7
External Process .....	8
External process features .....	8
OpenESB monitoring framework scalability .....	8
Publishers implementation .....	9
Event publisher .....	9
Business message publisher .....	9
External process implementations .....	10
External process implementation 01 .....	10
External process implementation 02 .....	10
External process implementation 03 .....	11
License and fees .....	11

## **Introduction to the OpenESB monitoring framework**

You can use OpenESB monitoring framework to extract, at the runtime, technical and business information from the OpenESB BPEL and publish them to create reports, generate alerts or event to monitor your processes and your business.

### **What is OpenESB monitoring framework?**

At a higher level, an organization which wants to monitor its processes needs to extract two types of information

The first type is the technical information on how the processes technically run. It is used to evaluate the service availability, response time, quality of service, SLA. It could also be used to trace the activities at the runtime and understand “How the processes run”.

The second type gives information on the business itself. It is a kind of snapshots of the process whose content is defined at the design time and provides information for business analytics processes. It replies to the question “What my process ‘processes’”

The OpenESB monitoring framework replies to these questions and provides two types of messages named “BPEL event” and “Business Message”.

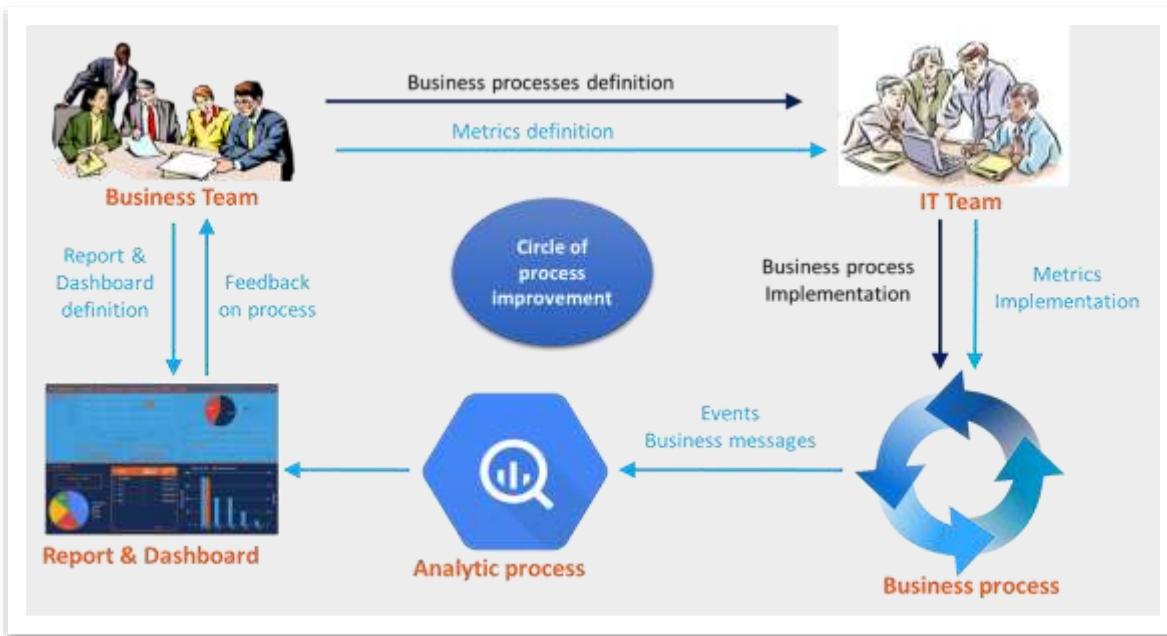
### **Why an OpenESB monitoring framework?**

Currently, OpenESB standalone edition (v3.1) inherited a monitoring system from the OpenESB legacy editions. This monitoring relies on a complex mapping process that stores the BPEL Java object in a database. Unfortunately, this process requires so many resources that very few users could use it. In a current configuration, when the legacy monitoring is on, BPEL response times can be increased by up to 10 times. Moreover, BPEL events and business messages can only be stored in a local relational database. It excludes the new persistence systems such as the NoSQL databases or the cloud data lakes.

The difficulty to get metrics from the running processes cuts the links between the business teams and the IT process implementations and due to the lack of feedback, stops the business processes improvement and often generates a loss of income to the companies.

The OpenESB monitoring framework has been designed and developed to solve these issues. It is a light and scalable framework that does not deeply affect the BPEL performances. Events and business messages sending is not limited to the relational databases but targets the latest big data tools such as Kafka, Geode and offer and access to the data lakes on the could.

Pymma relied on the following picture to design, develop and implement the OpenESB monitoring framework.



Our main idea was to create a virtual circle with the following steps:

- Business team: Defines the business processes and the metrics that will be used to evaluate the conformance of the process behaviour with the company's requirements.
- IT team: Implements the business processes and the messages that will be used to generate the metrics.
- Business process: Generates events and business messages
- Analytic process: Collects and improves events and business messages
- Report and dashboard: Provides metrics for the business team

## OpenESB monitoring framework installation

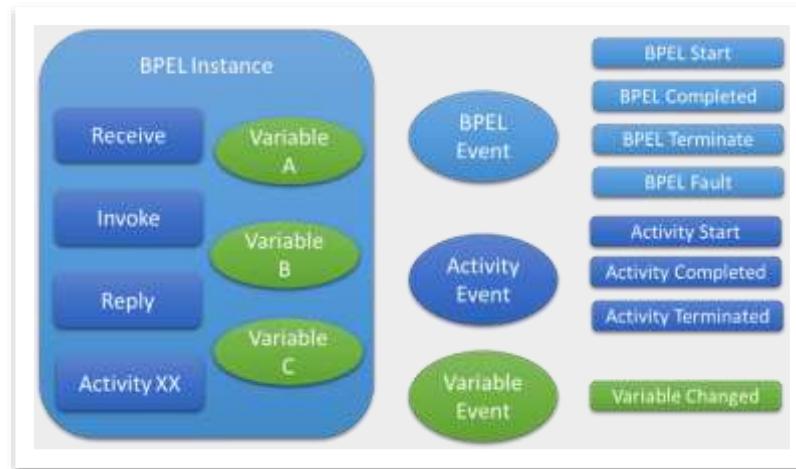
To install the OpenESB monitoring framework, you must deploy on OpenESB, the last version of the BPEL engine developed by Pymma and the publishers you need to use. Pymma's BPEL is backward compatible with the OpenESB Standalone Edition. You don't need to redesign the BPEL processes already deployed on OpenESB to access to the BPEL Events.

Business message definition is concurrent with the process implementation and does not impact it. To publish business messages, you need to add message mappings to the BPEL projects and then redeploy them.

No other component is required to run the OpenESB monitoring framework.

### BPEL Events

Three types of Events are defined, the BPEL event, the Activity event, and the Variable event.



Events contain raw information such as

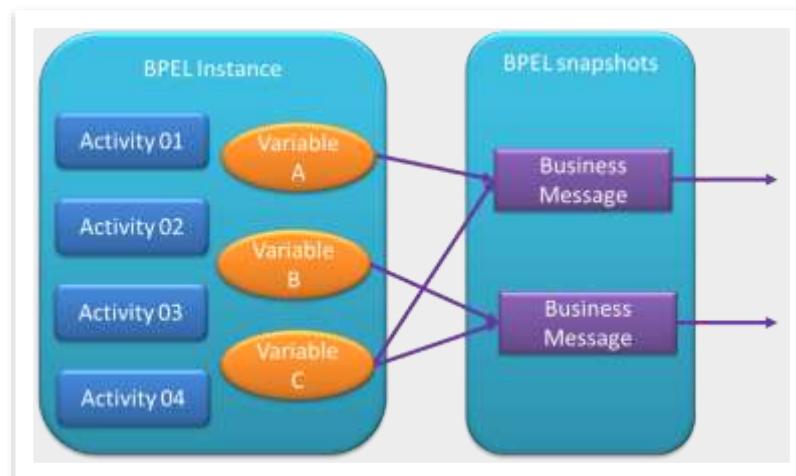
- What is the event?
- When it happens
- Where it occurs

It is up to the analytic process to use this raw data and extract useful information for the users.

Using BPEL event does not impact the business process design, and no additional development is required.

### **Business messages**

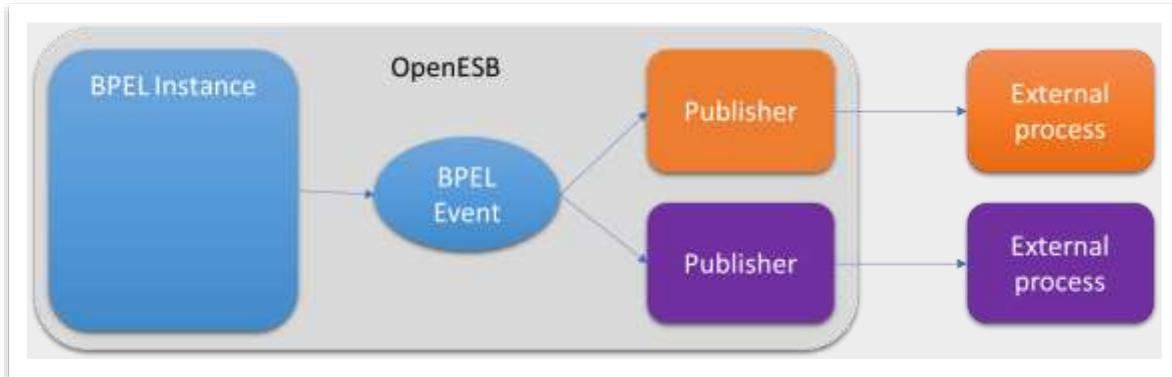
A business message is a makeup of variable embedded in the process context. At the design time, The OpenESB user defines concurrently to the BPEL development, the element they want to extract from the BPEL context and compose a new structure often defined by the business teams.



In a process, many business messages can be implemented and published concurrently. Generating business message does not impact the process development.

## Event publisher

Once the event is issued, it is sent to an external analytic process not embedded in OpenESB.

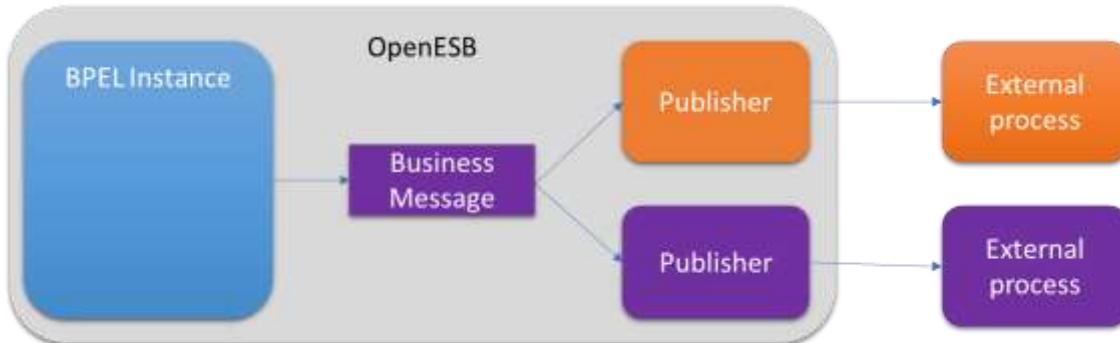


The OpenESB architecture based on multi-instances deployment proved its capability to daily process billions of messages on production. Since OpenESB resources are dedicated to run and manage the business process, the external processes that will manage events and business messages must be delegated to an external system that releases OpenESB from this overload.

The event publishers are multithreaded and optimized to provide the best performances with few resources. Regarding the protocol used, they publish from hundreds to thousands of messages per second.

## Business message publisher

The business message publisher has mainly the same features than the event publisher. Nevertheless, its design is often a bit more complex than the event publisher since it deals with messages without a predefined structure.



The business message publishers are multithreaded and optimized to provide the best performances with few resources. Regarding the protocol used, they publish from hundreds to thousands of messages per second.

## External Process

The external process must not share the same resources as the OpenESB. It could be designed, developed and deployed by the users or by Pymma on-demand. It could be any software such as alerts management, event processing, data lake, statistic or analytic tools, etc.

Consequently, the publisher acts as an “external process” client. Also, it transforms and adapts the event and the business messages to match the message format expected by the external system. It must not be used to execute a part of the external process and then affects BPEL performances.

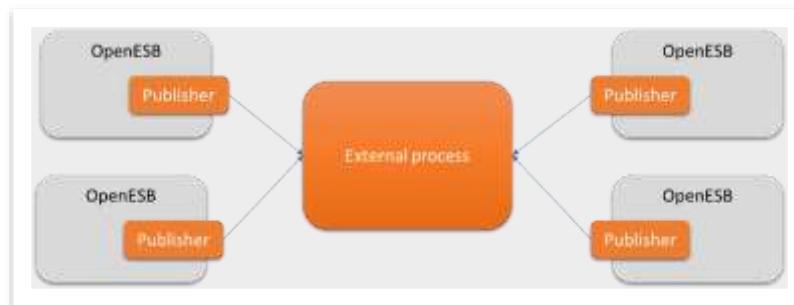
### External process features

The external process must fulfil some critical requirements. It must be an asynchronous system with a high capability of ingestion since OpenESB monitoring framework relies on a “fire and forget” policy and does not wait for any acknowledgment from the External process to release its resources.

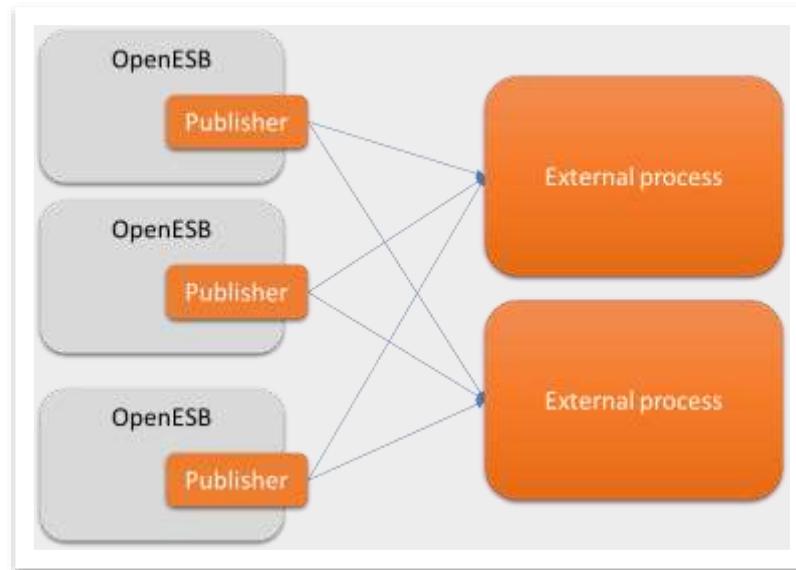
Some configurations working with multiple OpenESB instances generate up to billions of messages on a daily basis. It is up to the user to calibrate the capability of ingestion of the external process and keep safe and secure the messages published by OpenESB.

## OpenESB monitoring framework scalability

The OpenESB monitoring framework has a powerful horizontal scalability. So, it matches the new architecture based on virtualization and cloud. Messages generated by the framework are stateless and atomic and allow many OpenESB instances to push messages to the same external system.



For better performances, the external module must scale horizontally as well.



## Publishers implementation

### Event publisher

The list of event publishers is not exhaustive.

- *File event publisher*
- *MongoDB event publisher*
- *Elasticsearch event publisher*
- *Apache Geode event*
- *Apache Kafka event publisher*
- *RDB event publisher*
- *Google Big Query event publisher*

On-demand, Pymma develops event publisher for its customer

### Business message publisher

This list of business message publishers is not exhaustive.

- *File business message publisher*
- *MongoDB business message publisher*
- *Elasticsearch business message publisher*
- *RDB business message publisher*
- *Google Big Query business message publisher*

On-demand, Pymma develops event publisher for its customer

## External process implementations

It is up to the OpenESB user to design and implement an external process that fulfills its requirements. As an example, we detail three cases that could be chosen as external process design implementation.

### External process implementation 01

This implementation is dedicated to the configuration with a limited number of messages.



OpenESB publishers put the events, and the business messages in a database which acts as a buffer and a container of data and the BI tools query the database to generate reports and dashboards.

If the configuration is simple and easy to deploy, but the OpenESB user must consider that the events and the business messages are recorded in a raw format and require a complex query to provide the right data for reports and dashboards. Database ingestion capability could be a point of contention since relational databases demonstrate low performance in high rate ingestion.

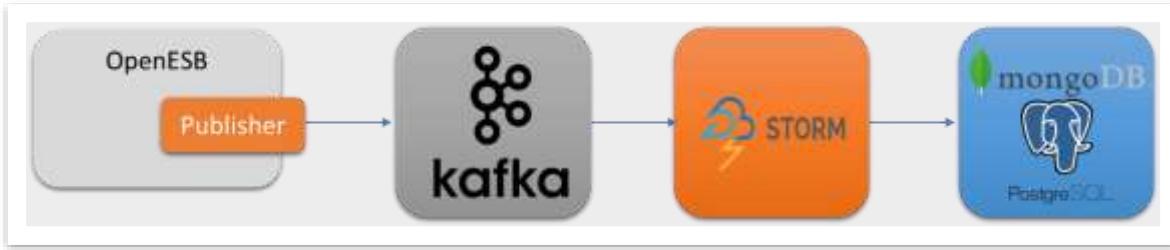
### External process implementation 02

Cloud application performances can be used to generate reports and dashboard.



OpenESB publishers send the event and business messages to a cloud database. Then, online BI editors access the database to generate reports and dashboard. In this implementation, we rely on the cloud's power to provide a powerful ingestion and execute complex requests. The cloud streaming databases such as Google big query are more suitable for this type of processes than the classical relational databases often dedicated to transaction and consistency.

## External process implementation 03



This implementation is made up of three elements:

- A buffer: Kafka receives the messages from OpenESB in an asynchronous mode
- An engine: Storm processes the raw events and messages and generates usable records
- A persistence system: MongoDB stores the record generated by the engine and makes them available for any analytic tool

This implementation is powerful and scales horizontally and matches the most demanding configurations, but it requires a large knowledge to be deployed and maintained.

### License and fees

Use of the OpenESB monitoring framework is subject to annual support subscription that gives you the right to use it in your configuration. The cost depends on the project, the number of instances and many other parameters. Please feel free to contact us for any further information on this subscription.