



LEARN MORE:

## **OUR DIFFERENTIATION**

### **Altworx in contrast with JBoss Fuse**

Altworx is built on top of the powerful foundations of Erlang, which provide Altworx with a distributed and fault-tolerant deployment platform. Altworx therefore does not need to reinvent the wheel and only deals with what is absolutely necessary for a successful integration product.

JBoss Fuse, and other traditional ESB products, on the other hand consists of several huge components in order to achieve the same functionality that is provided by Erlang.

JBoss Fuse consists of several thick layers, while Altworx is really nothing but a simple framework for building an automata network (which is somewhat similar to Camel routes) and providing automats for connecting it to the outside world (REST, SQL / NoSQL, LDAP, etc.), as well as, automats processing the data (Transform/Validate/Enrich).

Altworx offers the best of Erlang combined with the power of Enterprise Integration Patterns in a truly simple, lightweight fashion.

Altworx is so nimble, that it can handle millions of concurrent messages and just as easily fit into super-light embedded IoT appliances, without stripping down any of its power.

### **Altworx in contrast with Splunk**

While Altworx is not a direct competitor of Splunk, one might consider deploying Splunk in order to improve their business awareness and use it to increase their efficiency. Surely using Splunk has numerous benefits and Splunk can certainly be a great tool in addition to actual integrated solutions offered by Altworx.

It is important to understand the difference in paradigm between the two in order to see how they can complement each other. Splunk's main strength is analysis of data back in time, while Altworx is used to architect and operate integrated platforms in order to build new products and solutions.

In other words, one uses Splunk to analyze the operational data while one uses Altworx to create a tool which analyzes the data automatically, so the user no longer has to do it manually.

### **Altworx in contrast with Hadoop**

Hadoop is probably one of the most known big-data products and as such it provides a framework for distributed computing. Altworx is built on top of Erlang, which is an excellent distributed computing runtime and much more generic than Hadoop.

Altworx provides a higher level of abstraction. Hadoop is used to process large input data files by executing a distributed map/reduce algorithm on them to produce output files.

Altworx is used to describe stateful data flows, called automats, which continuously process input services and generate output services. Altworx contains a number of connectors, consumers and producers to be able to plug it into existing systems.

That said, the Hadoop approach requires a single place of input data files and getting any value from them means writing a map/reduce script executed on all of them. To alleviate the problems related to huge amounts of input data, one often combines Hadoop with machine learning and artificial intelligence techniques, these are predictive methods and as such can not be used to build primary systems.

Altworx on the other hand reacts on input events, pushes them through its network of automata and develops new data, only when appropriate.

## Altworx in contrast with traditional BI tools

Similar to comparison with Splunk, which is practically a BI tool for machine data, traditional BI vendors focus on making some sense in the large heaps of already existing data. This approach usually requires engineers with data science backgrounds, business analysts with the domain knowledge and huge amounts of hardware, able to process all the unstructured or semi-structured input data.

Altworx saves the day by looking at the problem from the business perspective first. Altworx first defines inter-department processes, implements them via its automata network, then connects to already existing systems to react on the events coming from the departments themselves. Figure 5 shows a typical deployment scenario for Altworx – the inefficiencies often come from the boundaries between departments as most enterprises focus on efficiency inside departments only – the efficiency paradox<sup>1</sup>.

---

<sup>1</sup> <http://www.digitaltonto.com/2015/the-efficiency-paradox/>

OUR DIFFERENTIATION



Figure 1