

# Comparative analysis of tools for the integration of IT systems

## Narzędzia do integracji systemów informatycznych - analiza porównawcza

Vladyslav Shkuta\*, Marek Miłośz

*Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland*

### Abstract

This article describes the methods, approaches and technologies of information systems integration, compares the functionality of the selected tools for information systems integration from a selected point of view. In this case, it is an evaluation of the ability to perform the main functions of the Enterprise Service Bus and the resulting benefits.

*Keywords:* integration; backbone; function; comparison

### Streszczenie

Ten artykuł opisuje metody, podejścia i technologie integracji systemów informatycznych, porównuje funkcjonalność wybranych narzędzi do integracji systemów informatycznych z wybranego punktu widzenia. W tym przypadku jest to ocena zdolności do pełnienia głównych funkcji Korporacyjnej Magistrali Usług i wynikających z tego korzyści.

*Słowa kluczowe:* integracja; magistrala; funkcja; porównanie

\*Corresponding author

Email address: [vladyslav.shkuta@pollub.edu.pl](mailto:vladyslav.shkuta@pollub.edu.pl) (V. Shkuta)

©Published under Creative Common License (CC BY-SA v4.0)

## 1. Introduction

System integration (SI) is an IT or engineering process or phase concerned with joining different subsystems or components as one large system. It ensures that each integrated subsystem functions as required [1].

SI is also used to add value to a system through new functionalities provided by connecting functions of different systems [1].

When integrating systems, any objects can be involved, for example: computer networks, various kinds of equipment, integration software, host applications, user screens, ERP-, CRM-, SCM-applications, distributed objects, legacy systems, as well as custom or proprietary applications [2] and many others.

## 2. Methods of system integration

On the one hand, the integration mechanism can be represented by synchronous communication methods with blocking pending a response, based either on remote procedure call (RPC), or on its variations. Direct communication methods based on RPC (such as CORBA, DCOM, RMI) provide tight coupling and require intrusion in the parts to be integrated and also provide point-to-point communication. They may have the advantage of using a binary data exchange format with minimal latency, but the downside of this is the lack of Internet support. Web services, based on RPC, solve the web support problem, but the XML format they use is slower.

Another communication capability is provided by message queues controlled by queue managers. Such a connection is reliable and guarantees the delivery of messages to the recipient under any conditions, even if he is not online, waiting for him and storing the messages until he appears. This method is asynchronous, does not block actions, and allows for loosely coupled

systems; however, it is possible to implement synchronous communication provided by two queues (request and response).

Integration via files and databases is used in cases where other methods are not available. This type of integration can be applied to older systems that cannot be changed.

In UI level integration, two applications are integrated in such a way that a user can perform an operation that includes two different applications - without having to take into account that he is actually launching two applications.

In terms of enterprise integration, enterprise application integration (EAI or A2A) enables the exchange of data and business processes between any connected applications or data sources in the enterprise (in particular, ERP-, CRM-, SCM-applications). The connection between pairs of applications can of course also be direct (so-called point-to-point), for example, using protocol and / or format adapters at one or both ends (using technological APIs such as FTP, IIOP, RPC, remote or batch interfaces), but this is labor-intensive and may require intrusion in these applications. An easier way is to use middleware to minimize interference and ensure reliable data transfer.

It is best of course to use integration tools, because they provide a wide range of capabilities: adapters, message queues, guaranteed message delivery when needed, brokers, business process modeling, application deployment, composite applications, routing, format conversion, security, administration and monitoring capabilities. and many others. They allow to build a service-oriented architecture (using web services), to represent applications, systems, functions as reusable services located on a single backbone, standardizing formats and reducing the number of links (in comparison with a point-to-point topology). The capabilities

depend on the tool (be it a message broker, ESB, application server, or even a BPM tool) and what the manufacturer has put into it, and the tool must be used depending on the needs, because for a small integration (up to three applications) a simple point-to-point solution can be enough (without the need for special tools that only complicate the situation).

B2B integration requires standardization of formats and coordination of partners, because different companies can use different technologies and solutions, therefore consistency is necessary, especially for financial institutions and international partners.

B2C integration is related to e-commerce, which in turn depends on Internet technology. Organizing this type of trading also requires standards and organizational methods that enable the discovery of services provided and the exchange of information in real time and with high reliability. A service-oriented architecture (SOA) with SOAP, WSDL and UDDI formats and the WS-Security specification can help in it.

### 3. Brief overview of selected tools for IT systems integration

From a variety of integration software, the following products were selected for comparison:

- OpenESB (Glassfish) – an open source and developed by the community (and previously Sun Microsystems) ESB;
- MuleESB – one of the most popular ESBs;
- Microsoft BizTalk Application Server – Application Server, which has become an example to follow.

**OpenESB** is a distributed integration middleware infrastructure (a Java-based open source enterprise service bus) providing Web and non-Web services support, transformation, routing, and orchestration [3].

OpenESB is based on various industry standards such as JBI, WSDL, XML and Java. It enables connecting to various heterogeneous systems, gets them together to exchange messages in a standard way and collaborate with each other seamlessly. These standards help in shortening the learning curve, faster development and ease of deployment and managing [3].

OpenESB consists of 5 parts: the framework, the container, the components, the Integrated Development Environment and the development plugins [4].

The framework consists of a lightweight implementation of JBI (Java Business Integration) in Java. This implementation is container-independent and can run on any platform and any container. In addition to being lightweight, the OpenESB framework is also reliable and scales well. The framework implements a virtual bus known as the Normalized Message Router (NMR). This is a powerful asynchronous intelligent communication channel between components [4].

In comparison will participate the Glassfish-based version of OpenESB.

The JBI specification defines 2 component types: The services engine (SE) and the binding component (BC). The SE and BC implement the same interface contract; however, they behave differently [4]:

- Binding components act as the interface between the outside world and the bus, being able to generate bus messages upon receipt of stimuli from an external source or generate an external action/interaction in response to a message received from the bus.
- Service engines receive messages from the bus and send messages to the bus. SE's have no direct contact with the outside world. They rely on the bus for interaction with other components, whether binding components or other service engines.

OpenESB offers a set of graphical tools to ease complex SOA and integration developments. XLM, XML Schema, WSDL, BPEL editor, data mapping and Composition Applications graphical editors are proposed with OpenESB. Similarly, build, deploy, undeploy, run, test and debug tasks are managed by graphical tools. OpenESB provides the best ergonomics for ESB and SOA developments [4].

**MuleESB** is a scalable and distributed object manager from Mulesoft that can handle interactions with services and applications that support a variety of transport and communication technologies.

MuleESB was designed to be lightweight and easy to be embedded into Java applications and application servers, or to act as an independent server. It integrates with a number of frameworks such as Spring, Hivemind and Plexus and supports many transport and service components, including [5, 6]:

- JMS (Java Message Service);
- SOAP (Simple Object Access);
- REST (Representational State Transfer);
- MQ (Message Queuing);
- AQ (Oracle Advanced Queuing);
- Caching (Caching);
- JavaSpaces (a service providing a distributed mechanism for exchanging and coordinating Java objects);
- GigaSpaces;
- IM;
- JCA (J2EE Connector Architecture);
- Data Queues;
- System I / O (System Input-Output);
- JBI (Java Business Integration);
- BPEL (Business Process Execution Language);
- EJB (Enterprise Java Beans);
- AS / 400 (Application System / 400, IBM System i);
- HTTP (HyperText Transfer Protocol);
- JDBC (Java DataBase Connectivity);
- TCP (Transmission Control Protocol);
- UDP (User Datagram Protocol);
- SMTP (Simple Mail Transfer Protocol);
- File (FILE);
- FTP (File Transfer Protocol);
- much more.

The platform is Java oriented but can be a mediator for other platforms such as .NET using web services or sockets [6].

**MS BizTalk Application Server** is exceptionally well-suited to more complex IT landscapes where heavy integration demands [7]:

- Enterprise Application Integration (EAI): Is there a necessity to connect ERP system to a CRM system? Internally or externally? With BizTalk Server it becomes a whole lot easier, thanks to support for standards (JSON, XML, FlatFile) and an extended collection of adaptors that are provided for all kinds of packages and systems: SAP, Oracle, Siebel, MS SharePoint, IBM DB2, AS400.
- Business Process Management (BPM): thanks to integration, a whole slew of business processes can be simplified, which in turn leads to a shorter time-to-market and therefore cost savings. With transparent IT monitoring thanks to the BizTalk Server, it is known from the top what is really happening within an organization and which systems are linked together.
- Business-to-Business integration (B2B): BizTalk Server comes as standard with support for typical integration needs within so-called “industry verticals” such as Healthcare, Transport and Financials, because it supports a range of protocols and report formats like HL7, SWIFT, X12 and EDIFACT.
- Service Oriented Architecture (SOA) and Enterprise Service Bus (ESB): with BizTalk Server one can be adding a versatile platform to the business that one can expand internally to meet the needs using an Enterprise Service Bus (ESB), thanks to the flexibility and scalability of the .NET framework. Given that BizTalk Service includes full support for WCF (SOAP) and Web API (REST) it can also be used as a hosting platform for the services within SOA architecture.
- High Availability and Scalability: thanks to its robust underpinning by Windows and SQL Server Clustering, the BizTalk Server is a platform that provides total support for roll-out in a multi-node architecture, possibly across geographically discrete datacenters, providing support for critical processes of the business.

#### 4. Tools for IT systems integration comparison methodology

The main goal of the article is to evaluate the ability of selected software products for integration of IT systems (integration tools) to provide their main integration functionality, that is the functionality of the integration backbone.

The research will consist of several stages.

The goal of the first stage is to identify the ability of each selected product (integration tool) to provide certain functional capabilities. There are such groups of functional capabilities as the ESB functions described by Falko Menge in [8]. The ESB functions are placed and explained in Table 1.

Table 1: ESB Functions.

Function	Description
<b>Invocation (call)</b>	The ability of the product to send requests and receive responses from integration services and integrated resources.
<b>Routing</b>	The ability to determine the destination of messages during transport.
<b>Mediation</b>	The ability to transform or move non-equivalent resources in message transports.
<b>Adaptability</b>	The ability to adapt to other service providers.
<b>Security</b>	The ability to provide secure messaging at all points of message transport.
<b>Administration</b>	The ability to provide monitoring, logging, auditing, and integration scenarios.
<b>Process Orchestration</b>	The ability to perform complex business processes described in the standard language (BPEL).
<b>Complex Events Processing</b>	The ability to interpret events and the existing relationship between them for further guidance.
<b>Development Tool</b>	The ability to provide tools for designing, developing, testing, and deploying applications for professional development.

The functions presented in Table 1 are divided into functional capabilities.

At the first stage, the functional capabilities will be used as criteria for analyzing the ability of the product to perform the function to which they belong. The products, in turn, are presented in section 3 “Brief overview of selected tools for IT systems integration”, the functional capabilities will be described in the analysis process for each function.

Table 2 shows the scale of possible ratings to assign to the product for each criterion. The evaluation of support for each capability is determined by its ability to perform, completeness of vision, standards monitoring, and ease of realization.

Table 2: The values of characteristics evaluation.

Rating	Characteristic
1	The product initially offers functional capabilities or offers guidance on how to achieve them using certain abilities.
0.75	The product offers functional capabilities, but adapters may be required to meet the requirements in complete.
0.5	The product provides only a manual or requires the use of adapters to achieve some of the required functionality.
0	The product does not offer any documentation or functionality to meet the requirement.

The minimum score is 0, which equals 0% of the ability, and the maximum score is 1, which equals 100%.

The average of ratings for all product’s criteria determines its functional ability.

The goal of the second stage is to identify the overall ability of products to match the reference (benchmark) product of the Total compliance with the enterprise integration backbone class tool.

The reference product is an integration tool that can perform 100% of the Total compliance with the enterprise integration backbone functions. The more percentages the tool scores, the closer it is to the reference

product. This means that such a tool can be used in more complex scenarios and for broader integration needs and is perfectly suited for enterprise environments and developments.

At this stage, the criteria will be ESB functions, and the evaluation of the ability of products to meet the reference product of the Total compliance with the enterprise integration backbone class tool will be determined by the arithmetic mean between the ratings for all the criteria for the product.

### 5. Comparison of selected tools for IT systems integration

Based on the methodology described in section 4 “Tools for IT systems integration comparison methodology”, OpenESB, MuleESB, and Microsoft BizTalk are compared.

First, a description of the function and its capabilities is provided, and then is provided a comparison.

General evaluation of all functions and average of ESB functions capabilities for every tool is in the end.

**Invocation (call)** is the ability of a product to send requests and receive responses from integration services and integrated resources. It includes the following capabilities:

- Asynchronous call. These are non-blocking actions that allow the main program to continue processing.
- Synchronous messaging. Ability to simulate synchronous communication when a program call waits for a result before continuing processing.
- Translation of protocols. Ability to transfer from one type of communication Protocol to another (for example, TCP/IP to HTTP).
- FTP support. It is a Protocol commonly used for file exchange over a TCP/IP-based network to manage files on a computer on that network.
- Support for SFTP. FTP Protocol with additional built-in security.
- HTTP support. A communication Protocol used for transmitting information in intranets and WWW. Hypertext transfer protocol.
- Support for POP3. Protocol for sending mail messages to the server on request.
- SMTP support. Simple Protocol for sending mail messages.
- IMAP support. Mail Protocol of the Internet.
- EDI support. Ability to exchange electronic files.
- Support for JMS. Ability to process messages from Java services.

The ability to comply with the call rules is shown in Table 3.

Table 3: Compliance with invocation rules.

	Characteristic	BizTalk	OpenESB	MuleESB
Invocation (call)	Asynchronous invocation	1	1	1
	Synchronous messaging	1	1	1
	Translation of protocols	1	0	0.75
	FTP support	1	1	1
	Support for SFTP	0	0	1
	HTTP support	1	1	1
	Support for POP3	1	0	1
	SMTP support	1	0.5	1
	IMAP support	1	0.5	1
	EDI support	1	1	1
	Support for JMS	0	1	1
<b>Ability</b>	<b>0.81</b>	<b>0.64</b>	<b>0.97</b>	

**Routing** is the ability to determine the destination of messages during transport. It includes the following capabilities:

- Content-based Routing. Searches for message routing based on the current content of the message itself, rather than on a specific destination.
- Endpoint independence. Localizing a service that doesn't depend on the running application.
- Delivery guarantee. Describes a Protocol that allows messages to be reliably delivered between distributed applications, despite failures of software components, systems, or networks.
- Load balancing. Ability to deploy multiple instances of the service and use a load balancer to send requests and release the required traffic.

The ability to comply with routing rules is shown in Table 4.

Table 4: Compliance with message routing rules.

	Characteristic	BizTalk	OpenESB	MuleESB
Routing	Content-based Routing	1	0.5	1
	Endpoint independence	1	0.5	1
	Delivery guarantee	1	0.5	1
	Load balancing	1	0.5	0.5
	<b>Ability</b>	<b>1</b>	<b>0.5</b>	<b>0.87</b>

**Mediation** is the ability to transform or move non-equivalent resources in message transports. It includes the following capabilities:

- Message Verification. Validation is a simple check that the incoming message contains a well-formed XML format and matches a specific schema or WSDL document that describes the message.
- Displaying diagrams. A tool for making it easier to display message schemas.
- Routing messages with errors. When a message fails to be received, it is sent to another location to perform additional actions.

- Provision and registration of services. Ability to accompany new services and register them in a configuration-based style.

The ability to comply with mediation rules is shown in Table 5.

Table 5: Compliance with mediation rules.

	Characteristic	BizTalk	OpenESB	MuleESB
Mediation	Message Verification	1	0.75	0.75
	Displaying diagrams	1	0.75	1
	Routing messages with errors	1	0.5	0.5
	Provision and registration of services	1	0	1
	<b>Ability</b>	<b>1</b>	<b>0.66</b>	<b>0.81</b>

**Adaptability** is the ability to adapt to other service providers. It includes the following capabilities:

- Microsoft SharePoint Services. Integration with Microsoft SharePoint Services.
- Custom adapters. Examples and existing documentation for creating custom adapters.
- MS SQL Server. Integration with MS SQL Server.
- Sybase ASE Server. Integration with the Sybase ASE Server.
- Oracle Databases. Integration with Oracle databases.
- .NET Adapter. Adapter for .NET.
- IBM MQ support. Support for the IBM MQ Series.
- MSMQ support. Integration with the MSMQ queue manager.
- Sockets support. Adaptation to communications via Sockets.
- MS Windows systems. Integration with Microsoft Windows systems.
- Linux/Unix systems. Integration with Linux/Unix systems.
- Other OSs. Integration with other operating systems.

Compliance with the adapter support requirements is shown in Table 6.

Table 6: Compliance with adapter support requirements.

	Characteristic	BizTalk	OpenESB	MuleESB
Adaptability	Microsoft SharePoint Services	1	0	0.75
	Custom adapters	1	0	0.5
	MS SQL Server	1	0.75	0.5
	Sybase ASE Server	1	0.75	0.5
	Oracle Databases	1	1	1
	.NET Adapter	1	0.5	1
	IBM MQ	1	0.5	0.5
	MSMQ support	1	0.5	1
	Sockets support	1	0	0.5
	MS Windows systems	1	1	1
	Linux/Unix systems	0	1	1
	Other OSs	0	1	1
	<b>Ability</b>	<b>0.83</b>	<b>0.58</b>	<b>0.77</b>

**Security** is understood as the ability to provide secure messaging at all points of message transport. It includes the following capabilities:

- Notification. The ability to report incidents based on certain parameters.
- High availability. Constant availability of the service regardless of the state of the server it is hosted on or the dependent servers it runs on.
- Crash recovery. Ability to restore all resources after a disaster.
- Security in web services. WS-Security describes improvements to SOAP messages to provide quality protection through message integrity, message confidentiality, and single-message authentication.
- Encryption/decryption of content. Support for message content encryption.
- Content authentication and authorization. Authentication and authorization based on the message content.
- Digital signature. The ability to use digital signatures to authorize access.
- Keys syncing. Mechanisms for ensuring key synchronization (if the ESB connects to other systems that can synchronize credentials in multi-transition systems).
- The impossibility of renouncing authorship. The ability to guarantee that the transmitted message was sent and received by the parties claiming to send or receive such a message.
- A Single Login (Single Sign-On). A service that allows administrators to map a Windows or non-Windows user account.

The ability to comply with safety requirements is shown in Table 7.

Table 7: Compliance with security requirements.

	Characteristic	BizTalk	OpenESB	MuleESB
Security	Notification	1	0.75	1
	High availability	1	1	1
	Crash recovery	0.75	0.75	1
	Security in web services	1	0.5	0.75
	Encryption/decryption of content	1	0.5	1
	Content authentication and authorization	1	0.75	1
	Digital signature	1	0	1
	Keys syncing	1	0.5	1
	The impossibility of renouncing authorship	1	0	0
	Single Sign-On	1	0	0.5
<b>Ability</b>	<b>0.9</b>	<b>0.39</b>	<b>0.82</b>	

**Administration** is the ability to provide monitoring, logging, auditing, and integration scenarios. It includes the following capabilities:

- Prohibit or restrict messages (throttling). A configuration that allows only a certain number of messages

to reach the service before a certain time period expires.

- Logging and auditing. The possibility of logging of messages and ease of implementation of these audit logs.
- Exception log processing. Logging exceptions related to server operations.
- Performance monitoring. A tool for monitoring system performance.
- Compilation of statistical data. Ability to save data for statistics on the use of server services and resources.
- Processing of Poison messages (correction, auto-forwarding). "Poison message" is a message that has exceeded the number of delivery attempts to the target application. This situation can occur when a queue-based application cannot process a message due to errors.
- Integrated development environment (IDE). A project-oriented graphical tool for specific developments.

The ability to comply with administrative requirements is shown in Table 8.

Table 8: Compliance with administrative requirements.

Characteristic		BizTalk	OpenESB	MuleESB
Administration	Prohibit or restrict messages	0	0	1
	Logging and auditing	1	0.75	1
	Exception log processing	0.75	0.5	1
	Performance monitoring	1	0.75	1
	Compilation of statistical data	0.75	0.75	1
	Processing messages with errors (correction, auto-forwarding)	0.5	0	0.5
	Integrated development environment (IDE)	1	0.5	1
	<b>Ability</b>	<b>0.71</b>	<b>0.46</b>	<b>0.92</b>

**Process Orchestration** is the ability to perform complex business processes described in the standard language (BPEL). It includes the following capabilities:

- Separation of rules. The ability to provide rules that will then be reused.
- Reuse rules between processes. Ability to reuse the provided rules.
- Dynamic reconfiguration. Dynamically adds a new service producer and consumer to the stage (orchestration).
- Exception handling. Mechanism for handling exceptions that occur during orchestration.
- Long-term transactions. Ability to handle orchestrations that take a long time to complete.
- Generation of web services. Ability to publish or create web services from orchestrations.
- Atomic transactions. Ability to support short-lived centralized operations or processes when a failure occurs and needs to be identified quickly.

- Coordination of web services. An advanced framework that provides protocols that coordinate the actions of distributed applications.
- Extended service support. Ability to interact programmatically with external services. These services are web services published with the ESB.
- Message tracking. A tool for tracking messages as they pass through the service layer.
- Publish and subscribe. Subscriptions to events that occur in orchestrations. Ability to publish events for interested parties.

Compliance with the requirements for process orchestration is shown in Table 9.

Table 9: Compliance with process orchestration requirements.

Characteristic		BizTalk	OpenESB	MuleESB
Process Orchestration	Separation of rules	1	0	0
	Reuse rules between processes	1	0	0.5
	Dynamic reconfiguration	0	0	0
	Exception handling	1	0.5	0.75
	Long-term transactions	1	0.5	0.75
	Generation of web services	1	1	1
	Atomic transactions	1	0.5	0.5
	Coordination of web services	1	0.75	1
	Extended service support	1	0	1
	Message tracking	1	0.5	0.5
Publish and subscribe	1	1	1	
<b>Ability</b>	<b>0.9</b>	<b>0.43</b>	<b>0.63</b>	

**Complex Events Processing** is the ability to interpret events and the existing relationship between them for further guidance. It includes the following capabilities:

- Dynamic resource allocation. Dynamic allocation of resources on the server.
- Data protection and cleaning. Data storage mechanisms, as well as a set of parameters for clearing data.
- Publish and subscribe to business events. Subscriptions to events that occur in orchestrations. Ability to post events for interested parties.
- Managing the publication of business events. A tool for managing business event publications.
- Business events subscription management. A tool for managing subscriptions to business events.

Compliance with complex events support requirements is shown in Table 10.

Table 10: Compliance with complex events support requirements.

		BizTalk	OpenESB	MuleESB
Complex Events Processing	Dynamic resource allocation	0	0	0
	Data protection and cleaning	0	0.75	0.5
	Publish and subscribe to business events	1	0.5	1
	Managing the publication of business events	1	0.5	1
	Business events subscription management	1	0.5	1
	<b>Ability</b>	<b>0.6</b>	<b>0.45</b>	<b>0.7</b>

**Development Tool** includes tools for designing, developing, testing, and deploying applications for professional development. It includes the following capabilities:

- Streaming compilation. A tool for streaming compilation.
- Errors reporting. A tool for reporting and managing flow errors.
- Real-time monitoring of processes. Real-time monitoring of processes.
- Monitoring and debugging threads. A graphical tool that allows to monitor and debug the tech. process.
- Easy application deployment. A tool to help deploy services, trace, and so on.
- Ease of applications migration. A tool for facilitating service migration between endpoints (instances).
- IDE. A tool for developing integrated applications.
- Authorship and defining of business rules. A tool for creating business rules.
- Versions control. Ability to deploy new versions of business rules. The ability to have multiple versions that can be deployed.
- Development and implementation of business rules. Published API for interacting with business rules from external applications.

Compliance with the requirements for development tools is shown in Table 11.

Table 11: Compliance with the requirements for development tools.

		BizTalk	OpenESB	MuleESB
Development Tool	Compilation of streams	1	0.5	1
	Errors reporting	1	0.5	1
	Real-time monitoring of processes	1	0.5	1
	Monitoring and debugging threads	1	0.5	1
	Easy application deployment	1	0.5	1
	Ease of migration of applications	1	0	1
	IDE	1	0.5	1
	Authorship and defining of business rules	0.75	0.5	1
	Versions control	0.5	0.5	0.5
	Development and implementation of business rules	1	0	1
	<b>Ability</b>	<b>0.92</b>	<b>0.4</b>	<b>0.95</b>

To summarize, it is necessary to sum up the evaluation results for each function in a single table. And find the overall compliance of each tool with the enterprise integration backbone (average of ESB functions capabilities).

Compliance with the reference product is indicated in Table 12.

Table 12: General evaluation of product characteristics.

		BizTalk	OpenESB	MuleESB
Functions	Invocation	0.81	0.64	0.97
	Routing	1	0.66	0.81
	Mediation	1	0.66	0.81
	Adaptability	0.83	0.58	0.77
	Security	0.9	0.39	0.82
	Administration	0.71	0.46	0.92
	Process Orchestration	0.9	0.43	0.63
	Complex Events Processing	0.6	0.45	0.7
	Development tool	0.92	0.4	0.95
	<b>Average of ESB functions capabilities</b>	<b>0.85</b>	<b>0.52</b>	<b>0.83</b>

## 6. Conclusions

Although the evaluation is quite subjective, it gave a General idea about the compared tools and demonstrates the difference that exists between the Microsoft BizTalk Application Server, MuleESB and OpenESB (Glassfish).

The evaluation revealed that Microsoft BizTalk Application Server and MuleESB are tools that can be used in more complex scenarios and for broader integration needs, so they are great for enterprise environments and development.

## References

- [1] What is System Integration (SI)?, <https://www.techopedia.com/definition/9614/system-integration-si>, [10.08.2020]
- [2] D. S. Linthicum, Enterprise Application Integration, Addison-Wesley, 2000, [https://www.academia.edu/8531349/Types\\_of\\_EAI\\_21](https://www.academia.edu/8531349/Types_of_EAI_21), [2.09.2020]
- [3] Welcome to OpenESB An ESB based on industry standards, <https://www.logicoy.com/openesb/>, [5.10.2020].
- [4] OpenESB, <https://en.wikipedia.org/wiki/OpenESB>, [27.10.2020].
- [5] Mule (software), [https://es.wikipedia.org/wiki/Mule\\_\(software\)](https://es.wikipedia.org/wiki/Mule_(software)), [20.10.2020].
- [6] MuleESB, <https://ru.wikipedia.org/wiki/MuleESB>, [20.10.2020].
- [7] Microsoft BizTalk Server Enterprise Integration with Microsoft, <https://www.realdolmen.com/en/solution/microsoft-biztalk-server>, [27.10.2020].
- [8] F. Menge, Enterprise Service Bus. In: FrOSCon: Free and Open Source Software Conference, University of Applied Science Bonn-Rhein Sieg, 2007.