

Choosing a Message-Oriented Middleware Platform

The next generation of message-oriented middleware platforms have arrived, enabling distributed messaging in the multi-cloud. But which platform is the right one for your business needs?

Message-Oriented Middleware – where are we?

Today’s modern business applications need to be able to harness the increasingly digitized world and seize opportunities in response to ever-increasing agility demands and pressure from customers. They need to be multi-cloud ready and provide the latest security features to be fit-for-purpose.

What is important to your business?

Every client needs a well-built messaging platform, that is scalable, fault-tolerant, with low-latency available on both On-premise and Cloud environments.

There are multiple message-oriented middleware systems that have entered the market recently, providing a wide range of features. The wide range of products available means it is tough to decide which messaging system is the most suitable in each case. Sandhata’s team of middleware and integration experts conducted detailed analysis on the industry-leading tools to find out.

Sandhata’s experience in Middleware and Integration

Sandhata has more than 15 years’ experience delivering complex Integration solutions in the financial services and telecoms sectors. We offer:

- Programme and Transformation management
- Application architecture and development
- DevOps and CICD
- Test automation and service virtualisation
- Infrastructure management and delivery

Summary of Recommendations

Application Type	Throughput	Recommended Platform	Platform Description
Traditional Application	MEDIUM	 RabbitMQ™	Open-source, lightweight, scalable message-broker system
		 TIBCO® EMS	Message-broker system adhering to JMS protocol
Event Streaming / Real-time & Traditional Application	MEDIUM or HIGH	 PULSAR	Open-source, cloud native, distributed streaming and messaging platform
Event Streaming / Real-time ONLY	HIGH	 <small>APACHE</small> kafka® <small>A distributed streaming platform</small>	Open-source, distributed, event streaming platform
Event Streaming / Real-time ONLY	VERY HIGH	 Informatica®	Ultra Messaging: High-performance, real-time streaming
		 TIBCO® FTL	High-performance, cloud-native, real-time streaming
		 CONFINITY SOLUTIONS	Low Latency Messaging: High-performance, real-time streaming

Appendix A

Analysis and Test Outcomes

Recommendations based on our in-depth analysis and testing

Medium throughput, Traditional application

[Rabbit MQ](#) and [TIBCO EMS](#) are both designed for lightweight messaging using request-response, queuing, and publish/subscribe models. Rabbit MQ facilitates efficient delivery of messages covering complex routing scenarios and provides lower end-to-end latency for medium throughput requirements. Rabbit MQ is offered on many platforms including JVM, and is widely supported in both cloud and on premise platforms. TIBCO EMS is part of Java EE and it is typically used in JVM based applications, but there is a lack of support for TIBCO EMS in cloud environments.

High throughput requirements for an Event Streaming / Real-time application

[Kafka](#) is designed for highly distributed workloads - streaming logs and other real-time data feeds at a high scale with low latency. We were very impressed in our tests.

Medium to high throughput, for Event Streaming & Traditional applications

[Pulsar](#) supports both streaming and standard message queuing in the same system. But, it has limited overall support and is not yet as widely deployed as other solutions. We are looking forward to Pulsar getting better support and wider industry-uptake. This will be a great option for applications which need both queuing and event streaming.

Mobile Messaging (Web sockets) / Cloud Messaging

Messaging for web and mobile applications provide bidirectional communications protocol between the client and server. These are mainly used for real-time applications like trading, monitoring, notification, chatting apps, gaming, and are faster than an HTTP connection. There are many Open source options available, using HTML5 and JavaScript. Commercial solutions include [Tibco Cloud Messaging](#), [Kaazing](#), and [Firebase Cloud messaging](#).

Extremely High throughput / Low Latency applications

If you need an extremely low latency and high volume messaging system, consider the commercial solutions available including [TIBCO FTL](#), [Informatica Ultra Messaging](#), and [Confinity Low Latency Messaging](#).

Medium & High Throughput Systems: Performance Highlights

	Kafka	Pulsar	Rabbit MQ	JMS
Throughput capability	High 100,000+ msgs/second	Medium or High 100,000+ msgs/second	Medium 1,000+ to 10,000+ msgs/second	Medium 10,000+ msgs/second
Expected latency	10 Ms	10 Ms	10 Ms	Dependent on deployment architecture and latency of the persistence engine. Can vary from 10s to 100s of milliseconds.
Scalability	Clusters can scale both horizontally and vertically.	Clusters can scale both horizontally and vertically.	Designed for large scale deployment but typically requires large-scale server infrastructure.	Designed for large scale deployment but typically requires larger-scale server infrastructure.
High Availability?	Yes	Yes	Partial	Partial
Global Distribution	Possible with third-party add-ons.	Native support for global distribution and data replication.	Support for complex routing to enable global architectures.	Support for complex routing to enable global architectures.

Appendix B

Medium & High Throughput systems: General Findings and Use Cases

General Findings

	Kafka	Pulsar	Rabbit MQ	JMS
License/Specification	Apache v2	Apache v2	Mozilla Public v2	JMS specification 2.0 (JSR 343)
Components	Kafka + Zookeeper (ZK will be removed)	Pulsar + Zookeeper + Bookkeeper + Rocks DB	Rabbit MQ	JMS Broker + Admin + Registry
Broker model	Pull	Push	Push	Push
Storage Architecture	Log	Log (Ledgers, Fragments and Entries), index	File (Index based approach)	File (index based approach), DB
Open Source?	Yes	Yes	Both open source and commercial solutions.	Both open source and commercial solutions.
Enterprise Support	Support offerings by Confluent among others.	Support offerings by Streamnative among others.	Pivotal	Supported by many vendors including Tibco, IBM, Oracle.
Operational Simplicity	Cluster-based technology with a medium-weight architecture.	Cluster-based technology with a heavy-weight architecture.	Light-weight architecture, requiring only Rabbit's own servers (brokers).	Light-weight Architecture, requiring hub/spoke model with centralised sever.
Cloud offerings	Enterprise cloud offering available with major clouds vendors including Confluent.	Not yet, hopefully soon!	Enterprise cloud offering available on major clouds vendors including Pivotal.	Enterprise cloud offering on major clouds.
Required Skills	OSS knowledge on Apache Zookeeper, Mirror Maker etc.	OSS knowledge on Apache Zookeeper, Bookkeeper etc.,	AMQ and AMQP specification knowledge required.	JMS specification knowledge required.
Documentation	Rich suite of documentation available.	Lightweight documentation available at the moment.	Rich suite of documentation available.	Rich suite of documentation available.
Client Library Languages Supported	Approx. 18 C, C++, Java, Python, Go, Scala, Swift etc.	Approx. 7 Java, Go, Python, C++ etc.	20+ Java, JavaScript, Python, C++, C#, SwiftRuby etc.	3 Java, C/C++, .NET

Use Cases

	Kafka	Pulsar	Rabbit MQ	JMS
Queuing	Not suitable	Queuing API available. This is similar to Rabbit MQ/JMS though it is limited in comparison	Fully Supported	Fully Supported
Publish / Subscribe	Supported	Supported	Supported	Supported
Event Streaming	Strongly Supported	Supported	Not Supported	Not Supported
Mission-critical Applications (99.99% uptime)	Supported	Lack of public use cases.	Supported	Supported
Routing	Support requires third party components.	Support requires third party components.	Native support	Native support