<https://www.mydbsync.com/blogs/eai-etl-when-you-should-choose-one-over-the-other/>

<https://www.mulesoft.com/resources/esb/enterprise-application-integration-eai-and-esb#origins-of-eai>

<https://blogs.mulesoft.com/dev/mule-dev/to-esb-or-not-to-esb/>

<https://www.confluent.io/blog/apache-kafka-vs-enterprise-service-bus-esb-friends-enemies-or-frenemies/>

Kafka is a message broker. It is comparable with other message brokers like ActiveMQ, RabbitMQ, Azure Service Bus

<https://www.confluent.io/blog/bottled-water-real-time-integration-of-postgresql-and-kafka/>

<https://blogs.mulesoft.com/dev/anypoint-platform-dev/application-vs-data-integration-which-is-better/>

<https://www.linkedin.com/pulse/enterprise-service-bus-vs-message-brokers-eai-soa-anuj-varma/>

http://bytecontinnum.com/2015/12/please-dont-call-kafka-messaging-system/

RabbitMQ, Azure Service Bus

Flink and Spark are in-memory databases that do not persist their data to storage. They can write their data to permanent storage, but the whole point of streaming is to keep it in memory, to analyze current data.

Data Processing

Spark processes data in batch mode while Flink processes streaming data in real time. Spark processes chunks of data, known as RDDs while Flink can process rows after rows of data in real time. So, while a minimum data latency is always there with Spark, it is not so with Flink.

Memory management

Flink pages out to disk when memory is full, which is what happens with Windows and Linux too. Spark crashes that node when it runs out of memory. But it does not lose data since it is fault tolerant.

An ESB is a message broker by default. There are 2 different broker approaches: hub and spoke (old EIA) and bus.

While there is overlap between ESBs and Message Brokers, that is not the case for ESBs and SOA/EAI. Certainly ESBs accommodate most SOA and most EAI architectures, but are not specifically designed to do so

Kafka is distributed system while old EAI was central. Kafka supports both database and application integration as well as transformations.

Kafka is unique because it combines messaging, storage, and processing of events all in one platform.