JMS Design Basics

Coming from an Enterprise Application background, Blended supports connections to one or more external JMS Providers. Normally, the API for external JMS Providers is deployed into Blended as plain API Bundles and just provide the client API for the external JMS provider.

As we can have multiple JMS vendors and potentially have multiple connections for a configured vendor, we have introduced the IdAwareConnectionFactory providing the properties vendor and provider on top of all JMS defined properties and methods.

In EAI seenarios it is a common requirement to tightly control the number of connections to the outside world. Therefore, each ConnectionFactory is wrapped within a BlendedSingleConnectionFactory, which manages exactly one connection on behalf of the container.

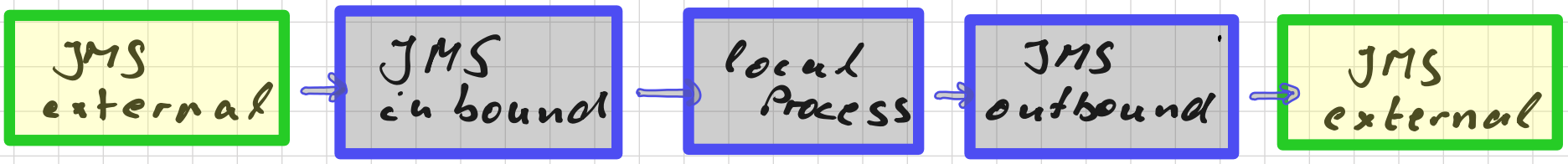
As a result, a bundle realizing a connection to an external provider usually creates one or more Blended Single Connection Factories and registers those with the interfaces ConnectionFactory and IdAware-ConnectionFactory. To identify registered ConnectionFactories the property combination „vendor/provider" must be unique within the container.

The internal Provider

A Blended container may require a local JMS broker, so that container client may leverage JMS to send messages to the container for further processing.
Also, a local broker can be used to realize a store and forward mechanism across unreliable networks.

For example, in the process sketched out below the blue boxes are all local to the blended container, outbound messages will go to a local queue first, then a JMS bridge will forward

the message to the external JMS provider and only acknowledge the message locally upon a successful send.



Blended supports exactly <u>one</u> internal JMS provider (which is currently realized with an embedded ActiveMQ broker). The associated provider configuration will have the "internal" property set to "TRUE".

Some bundles such as the JMS Bridge or the generic Dispatcher require an internal CF to work with.

As a general rule: if the container is to be deployed remotely, i.e. in a shop of a retail network, the container should leverage a local JMS broker and only connect to external providers with an instance of the JMS bridge

The next Blended release will have an implementation of stream based integration flows. This also brings an abstraction of JMS messages, the FlowMessage. For each technology in use a mapping between the FlowMessage and the technical external message is defined. As a result, the resulting streams are technology gnostic.

BlendedSingleConnectionFactory

Under the covers a BlendedSingleConnectionFactory will try to establish the connection as soon as possible and maintain it for as long as the container exists.
Anytime, the connection is not currently established, the „create" method will yield a JMSException.
Once disconnected, the BSCF will wait for an amount of time before a reconnect is attempted. This prevents the external provider to be flooded with connection requests after it has been restarted.

For connections that are used only to consume messages, it is sometimes hard to detect connection failures. Even though most modern JMS providers have some form of keep alive support, this is a feature not defined in the JMS specification and usually works a bit differently for each provider. Therefore, the BSCF has an embedded ping support to perform a JMS request-reply at regular intervals.

In case a ping fails, the interval is shortened, so that the ping happen more rapidly. If the ping has failed a defined number of times, the connection is considered dead and will be restarted.

The state of all BSCF is published and maintained via JMX.

Closing the connection beneath a BSCF will terminate all Streams that are using Sinks, Sources or Integration steps based on this BSCF. These streams can be configured with automated restarts, so that they will pick up the new connection eventually.