

Apache ServiceMix

Planning

- Overview
- Getting started
- What's in the 'Mix?
 - Karaf
 - Camel
 - CXF
 - ActiveMQ
 - ...

Planning

- Overview
- Getting started
- What's in the 'Mix?'
 - Karaf
 - Camel
 - CXF
 - ActiveMQ
 - ...

Overview

- Apache ServiceMix
 - Open-source integration platform
 - Apache Software License
 - Commercial support available
 - To ESB or not to ESB?
 - Red Hat JBoss Fuse & Talend ESB

Overview

- History
 - ServiceMix 3.x
 - JBI container
 - ServiceMix 4.x
 - OSGi-based container
 - Support for JBI 1.0 through NMR
 - ServiceMix 5.x
 - OSGi-based container
 - Support for JBI removed
 - Added Activiti, Akka, Drools, ...
-

Planning

- Overview
- **Getting started**
- What's in the 'Mix?'
 - Karaf
 - Camel
 - CXF
 - ActiveMQ
 - ...

Getting started

- Download
- Installation
- Running the container
 - interactive
 - background
 - as a service

Getting started

- Download ZIP file
 - from <http://servicemix.apache.org>

Download

The latest release is the [ServiceMix 7.0.0.M2 Release](#). We strongly encourage all users to use this release.

Default assembly

Our default assembly is the best way to get started with Apache ServiceMix.

Source assembly

Use this if you want to build Apache ServiceMix from source yourself

Documentation

Links to the documentation pages for this version

Apache ServiceMix 7.0.0.M2 (zip)

[\[PGP\]](#) [\[SHA1\]](#) [\[MD5\]](#)

Apache ServiceMix Sources 7.0.0.M2 (zip)

[\[PGP\]](#) [\[SHA1\]](#) [\[MD5\]](#)

Documentation for Apache ServiceMix 7.0.x

[\[Quickstart\]](#) [\[ActiveMQ\]](#) [\[Activiti\]](#) [\[Camel\]](#)

Installation

- Requirements
 - 100 MB free disk space
 - JDK 1.7 or higher
- Installation
 - unzip the archive to the local disk

Running the container

- interactive
 - bin/servicemix starts ServiceMix
 - access the command shell console

```
gertv@eowyn:~/opt/apache-servicemix-7.0.0.M2$ bin/servicemix
Please wait while Apache ServiceMix is starting...
100% [=====]

Karaf started in 0s. Bundle stats: 10 active, 10 total

ServiceMix

Apache ServiceMix (7.0.0.M2)

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or 'system:shutdown' to shutdown ServiceMix.

karaf@root>
```

Running the container

- in the background
 - bin/start script starts ServiceMix
 - bin/stop scripts stops ServiceMix
 - access the console
 - using ssh
 - using the webconsole (when installed)

Running the container

- as a service
 - uses Tanuki Java Software Wrapper
 - start ServiceMix interactively
 - in the console
 - feature:install wrapper
 - wrapper:install
 - follow instructions to complete installation

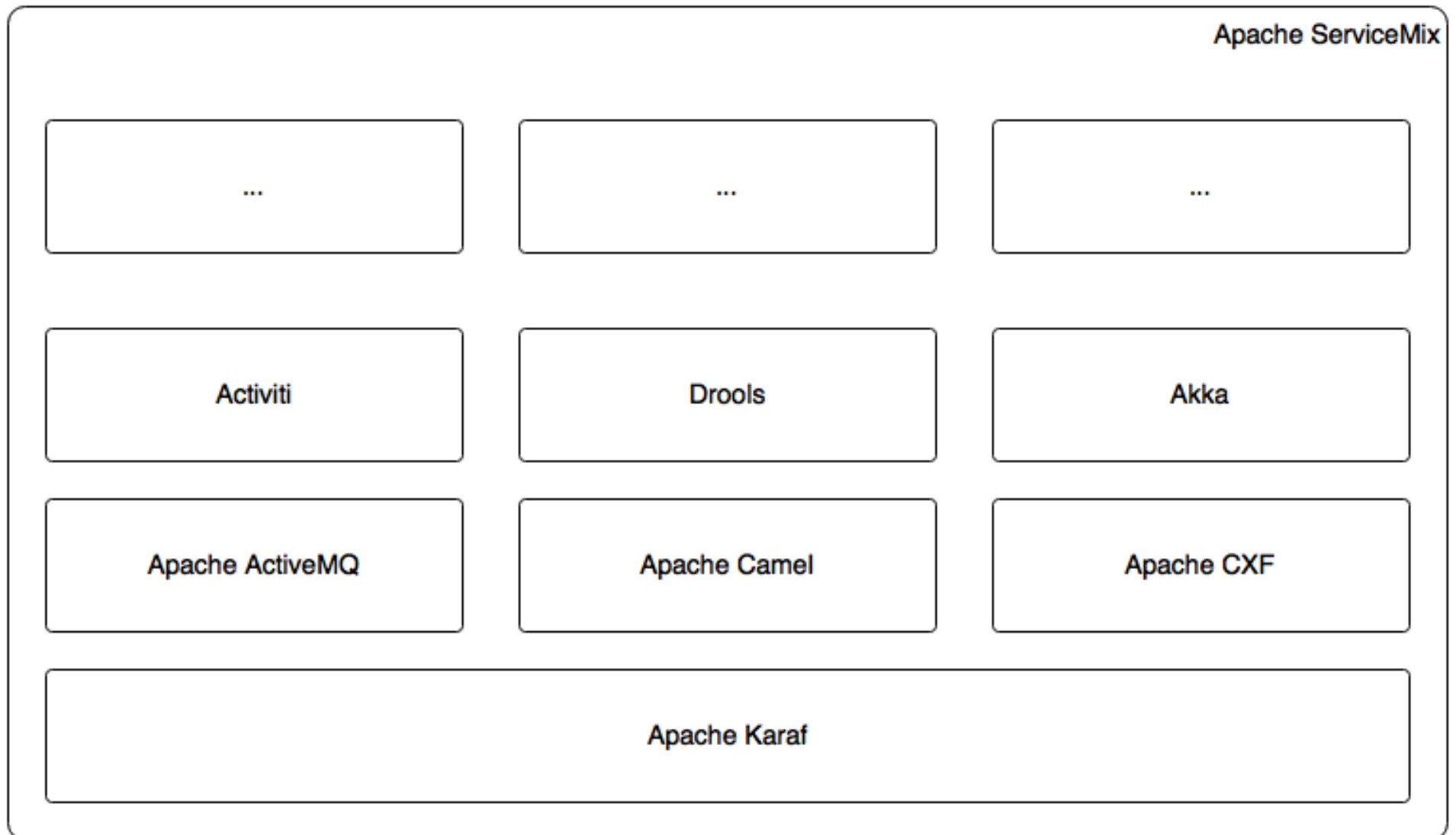
Planning

- Overview
- Getting started
- What's in the 'Mix?
 - Karaf
 - Camel
 - CXF
 - ActiveMQ
 - ...

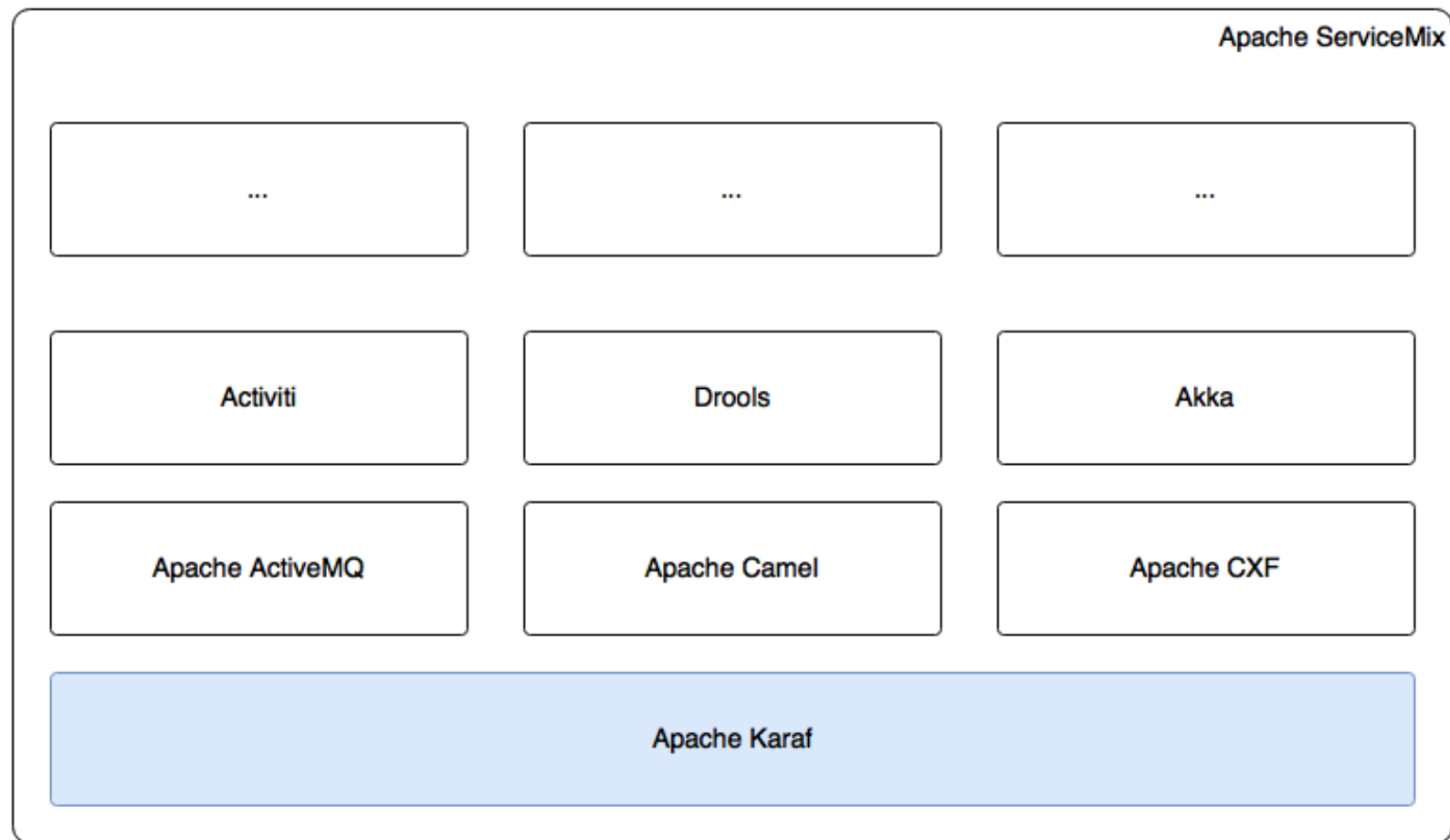
What's in the 'Mix?

- Apache ServiceMix consists of
 - Apache Karaf
 - Apache Camel
 - Apache CXF
 - Apache ActiveMQ
 - Activiti
 - Akka
 - ...

Overview and architecture



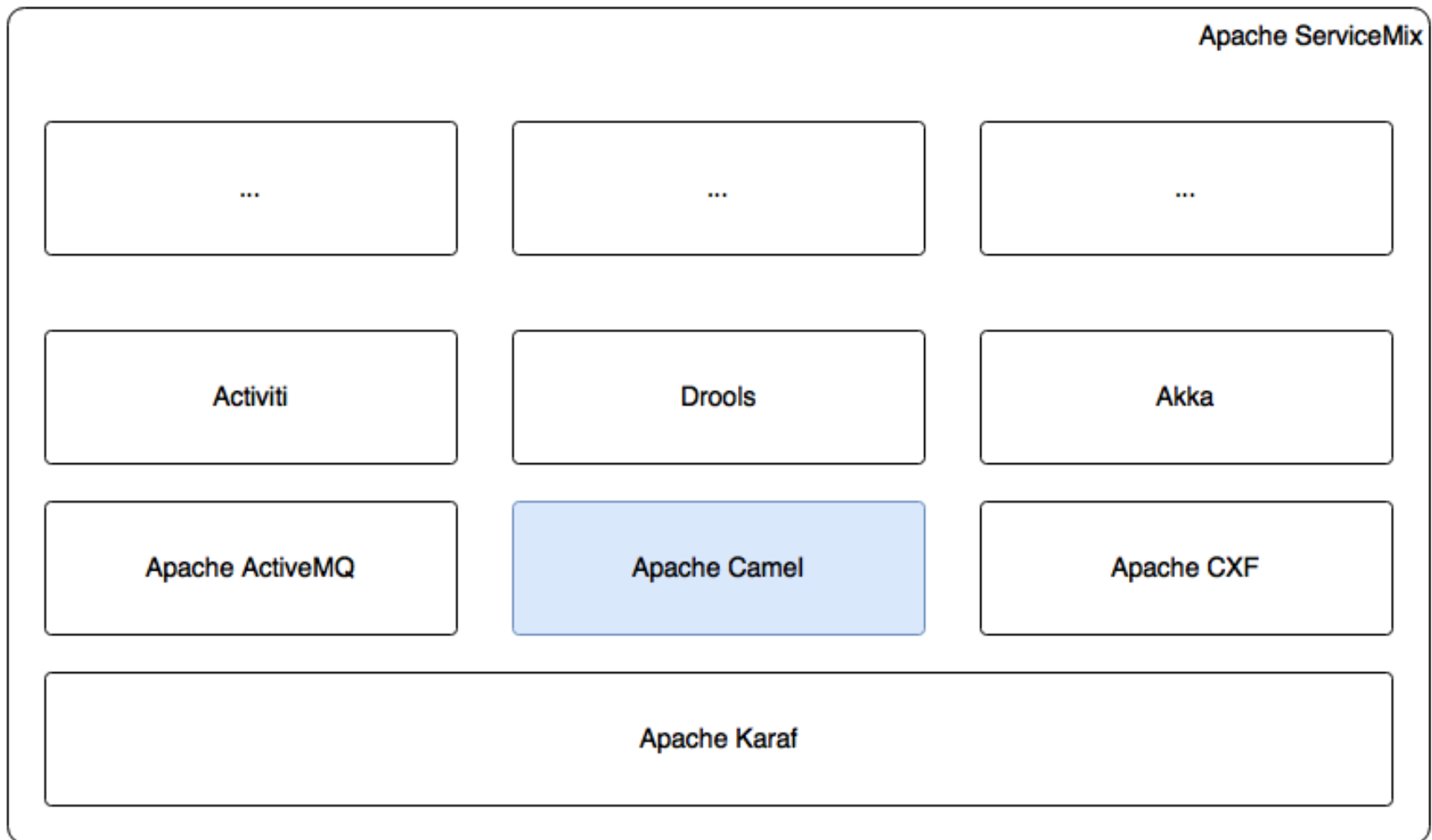
Apache Karaf



Why Karaf?

- OSGi-based container
 - (Remote) management
 - Deployment
 - Modularity
 - Provisioning
 - Centralized logging
 - Dynamic configuration

Apache Camel



Apache Camel

- Installing Camel Components
- Deploying Camel Routes
- Why Camel?

Installing Camel Components

- Camel itself is installed by default
- Additional Camel Components
 - can be installed through features

```
karaf@root>feature:list | grep -i camel
```

```
...
```

camel-xmlrpc	2.16.3		Uninstalled	camel-2.16.3	
camel-xmlsecurity	2.16.3		Uninstalled	camel-2.16.3	
camel-xmpp	2.16.3		Uninstalled	camel-2.16.3	
camel-xstream	2.16.3	x	Started	camel-2.16.3	
camel-yammer	2.16.3		Uninstalled	camel-2.16.3	
camel-zipfile	2.16.3		Uninstalled	camel-2.16.3	
camel-zookeeper	2.16.3		Uninstalled	camel-2.16.3	
activemq-camel	5.13.3	x	Started	activemq-5.13.3	

```
karaf@root>feature:install camel-yammer
```

```
karaf@root>feature:install camel-zipfile/2.16.3
```

Deploying Camel Routes

- Options for deployment
 - Plain Spring/Blueprint XML DSL deployment
 - Spring XML DSL in a bundle
 - Blueprint XML DSL in a bundle
 - Java/Scala DSL in a bundle

Deploying Camel Routes

- Plain Blueprint XML DSL deployment
 - Use deploy folder

```
<?xml version="1.0" encoding="UTF-8"?>
<blueprint
  xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.osgi.org/xmlns/blueprint/v1.0.0
    http://www.osgi.org/xmlns/blueprint/v1.0.0/blueprint.xsd">

  <camelContext xmlns="http://camel.apache.org/schema/blueprint">
    <route>
      <from uri="file:camel/input"/>
      <log message="Moving ${file:name} to the output directory"/>
      <to uri="file:camel/output"/>
    </route>
  </camelContext>

</blueprint>
```

Deploying Camel Routes

- Spring XML DSL in a bundle
 - Create a bundle with Spring XML file in META-INF/spring
- Routes can be defined
 - in the Spring XML file itself
 - in Java/Scala DSL RouteBuilders

Deploying Camel Routes

- Blueprint XML DSL in a bundle
 - Create a bundle with Blueprint XML file in OSGI-INF/blueprint
- Routes can be defined
 - in the Blueprint XML file itself
 - in Java/Scala DSL RouteBuilders

Deploying Camel Routes

- Blueprint XML file with Java RouteBuilders

```
<?xml version="1.0" encoding="UTF-8"?>
<blueprint
  xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.osgi.org/xmlns/blueprint/v1.0.0
    http://www.osgi.org/xmlns/blueprint/v1.0.0/blueprint.xsd">

  <camelContext xmlns="http://camel.apache.org/schema/blueprint">
    <package>be.anova.smx.routes</package>

    <routeBuilder ref="myRouteBuilder" />
  </camelContext>

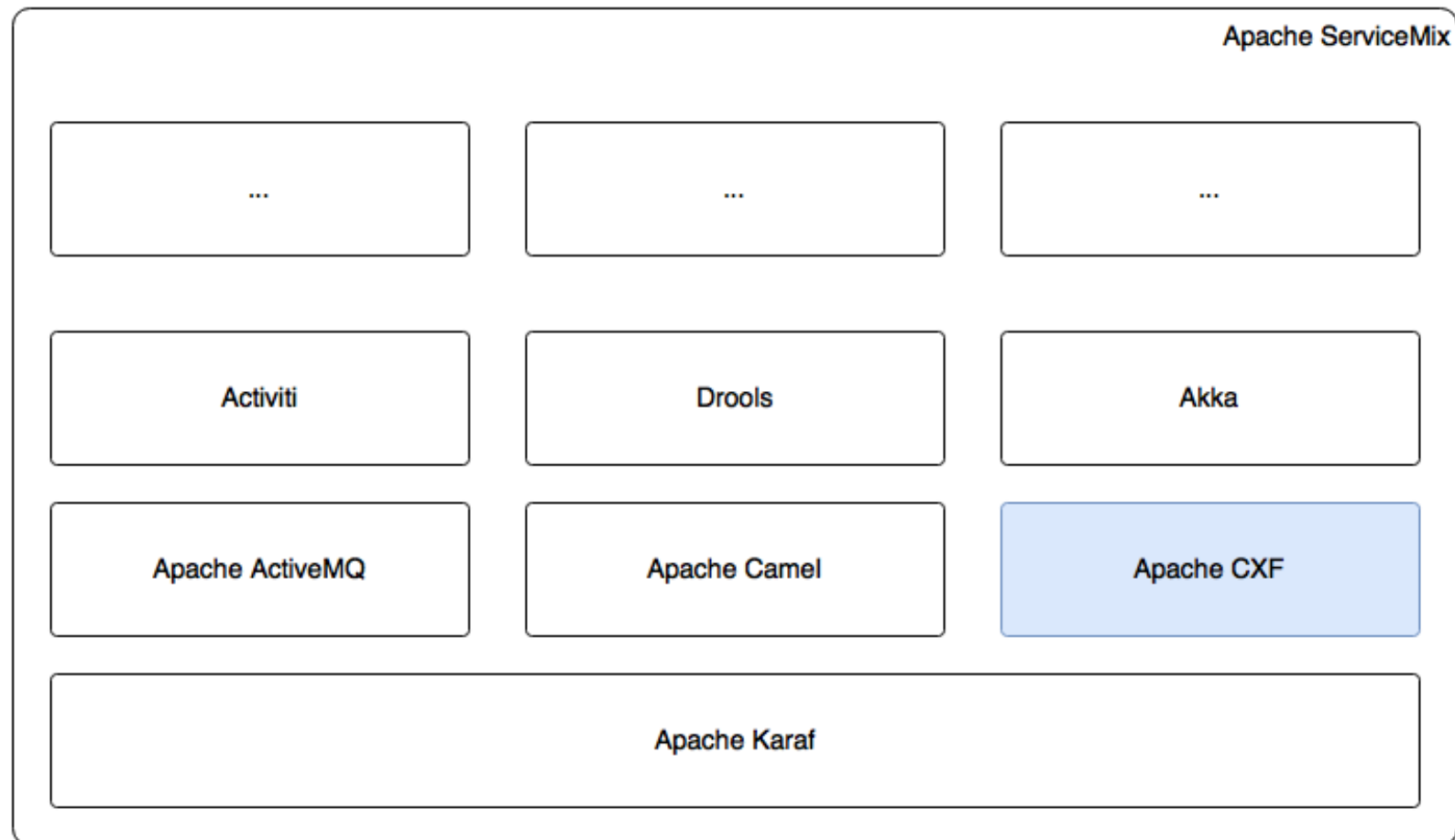
  <bean id="myRouteBuilder" class="b.a.s.course.AnotherRouteBuilder">
    <!-- inject services, beans, ... -->
  </bean>

</blueprint>
```

Why Camel?

- Default choice for integration
 - easy to use
 - feature-rich
 - lots of components available
 - add other technology when necessary
 - ActiveMQ for reliable, async messaging
 - CXF for web services support
 - Akka, Drools, Activiti, ...

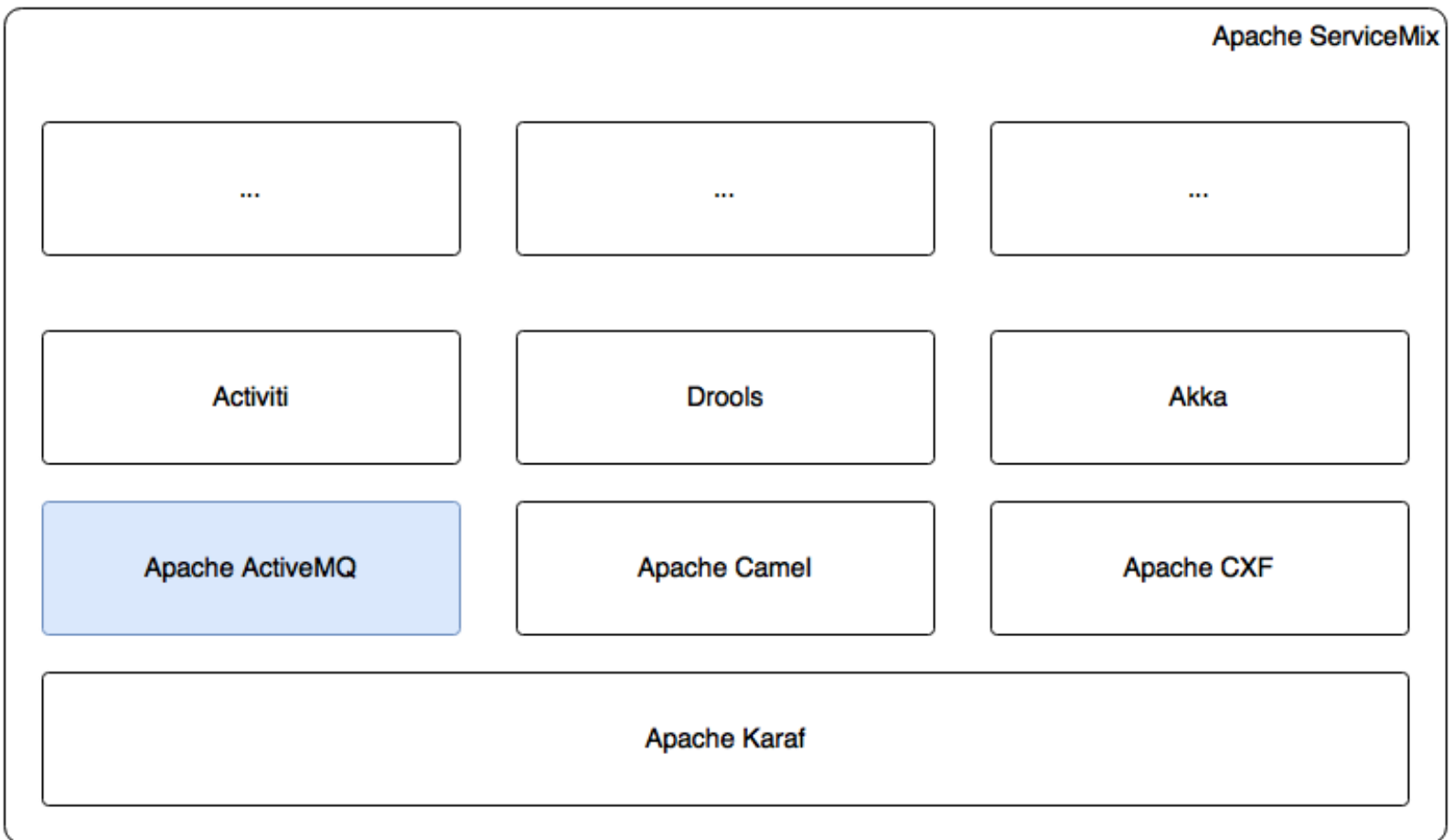
Apache CXF



Why CXF?

- All things webservises
 - JAX-WS for WS-* services
 - WS-Addressing, WS-Policy, WS-Security, ...
 - WS-Notification
 - JAX-RS for REST services
- Good integration with Camel

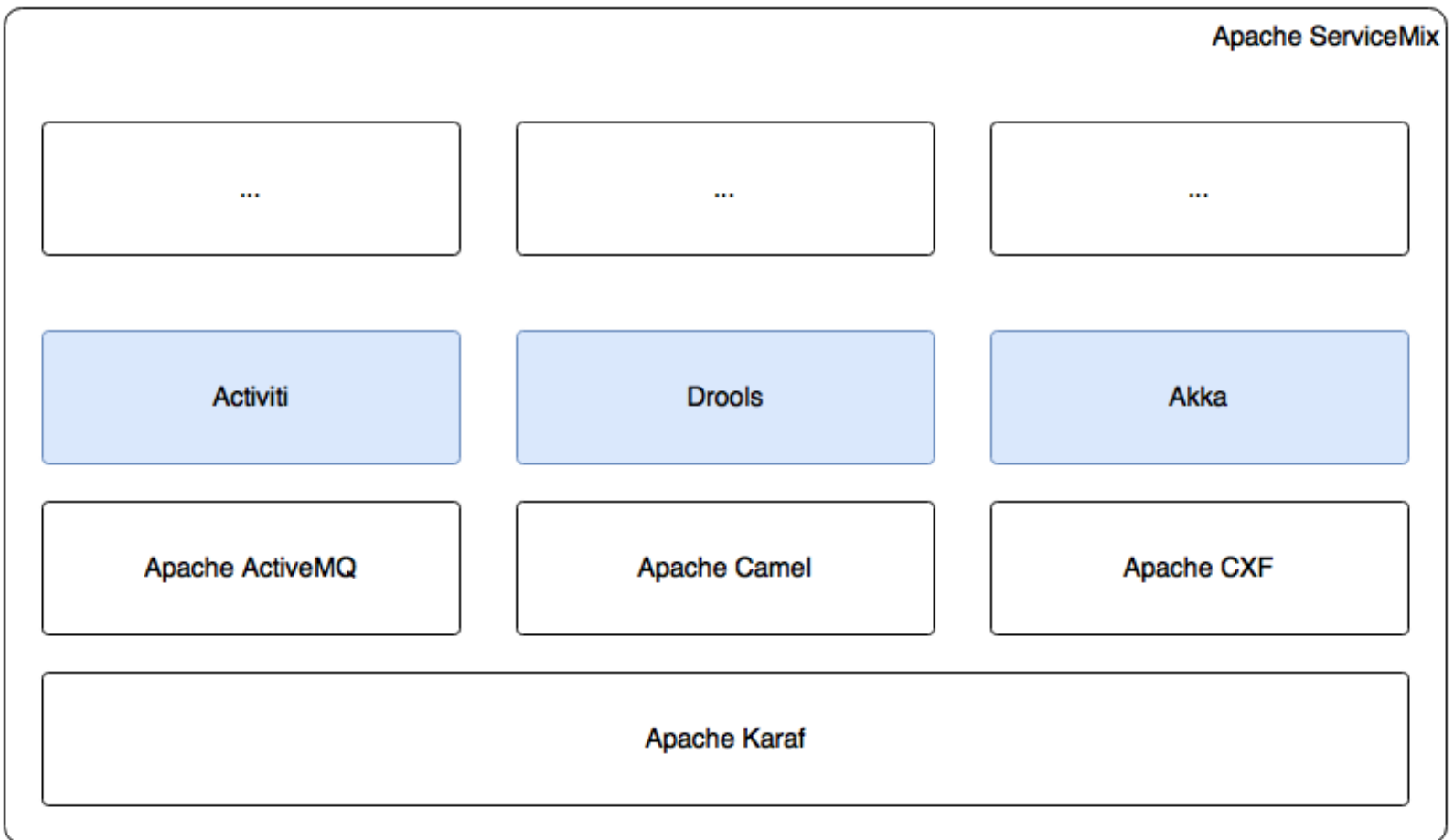
Apache ActiveMQ



Why ActiveMQ?

- Reliable and asynchronous messaging
 - Message persistence
 - Master/slave configuration
 - Load-balancing
 - Clustering
 - High availability
 - Scalability
 - Network of brokers

... what else is there?



... what else is there?

- Optional components
 - All of them are integrated with Camel
 - Activiti
 - Workflow and BPM
 - Akka
 - Highly concurrent/resilient applications
 - Drools
 - Business rules engine