

# Open Source Enterprise Architecture Using Apache

### What is Covered

Enterprise Architecture

- Integration
- Modular/Pluggable
- Message
- High Availability
- ESB
- loC
- Spring vs. Struts
- OSB

### Who Am I


OSB Projects

Example: Enterprise Architecture


### Conclusion

hkesler@savoitech.com

### Architecture




### loC



### Example

### Hardware Config



### ESB

Deployment

Scale Down!

Monitor/Logging

- AMQ
- Logging
- AMQ Admins


### Camel CXF

### Aggregational/Enrichment

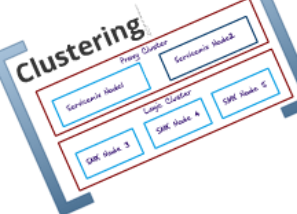
JMS

Testing

### Messaging



### Clustering



# Open Source Enterprise Architecture Using Apache

CALL  
WS CALL  
Bus



EX

Har

SMX Prox

AM

SAN

A

SMX C

af! Servicemix?

Messaging  
Deployment Options

# Who Am I



*SOA Architect*



*Application Fire Fighter*

*Evangelist*

Evangelist





SOA Architect

W  
W  
L





Application Fire Fighter

# What is Covered

Enterprise Architecture

Integration  
Mediation/Routing  
Messaging

High Availability

OSGI

IOC  
spring vs blueprint

ESB

# S Covered

Enterprise Architecture

ility

OSGI

log  
spring vs h.

Integration

Inte

Mediation/Routing



Messaging

m

High Availability

osgi

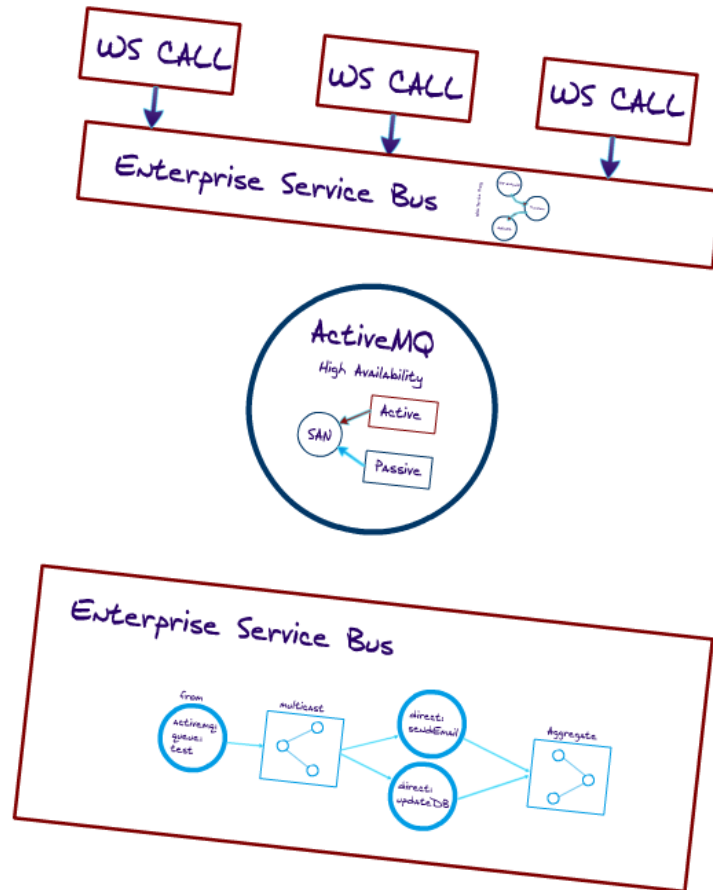
Architecture

IOC

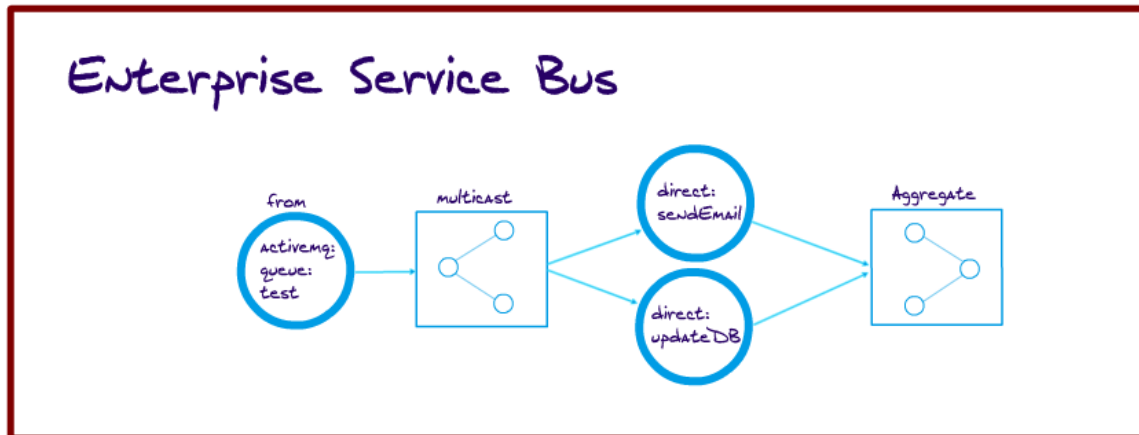
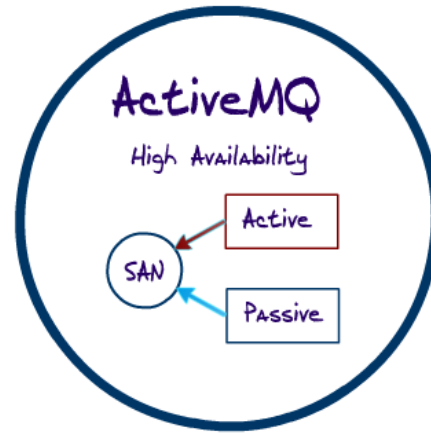
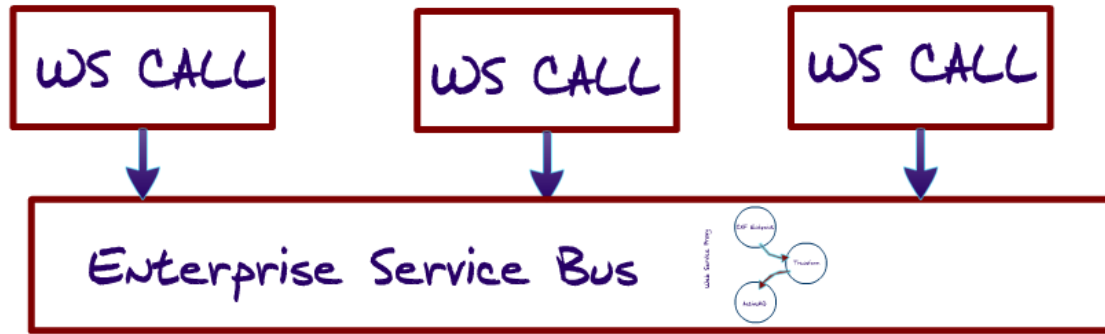
spring vs blueprint

ESB

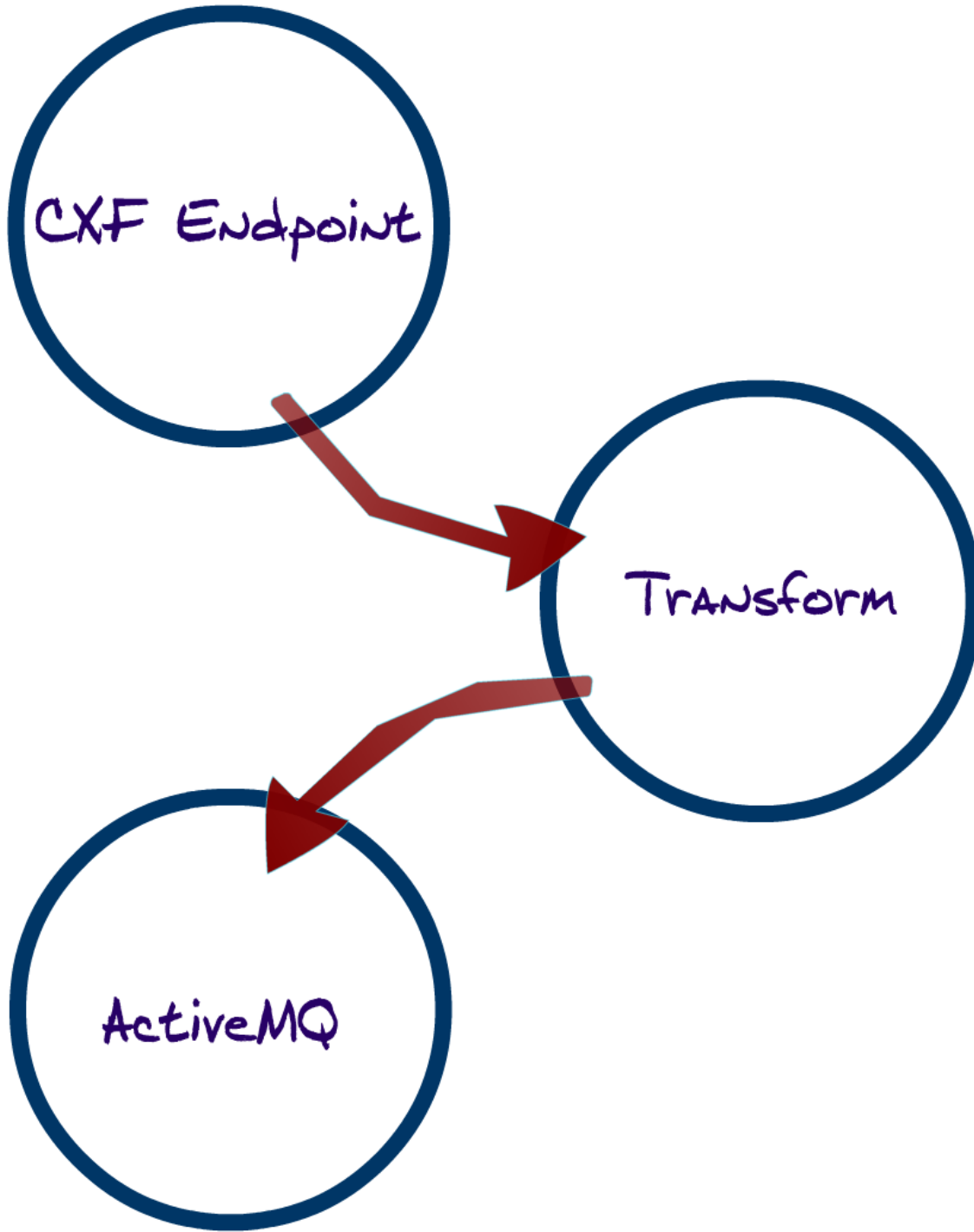
# Architecture *is important*



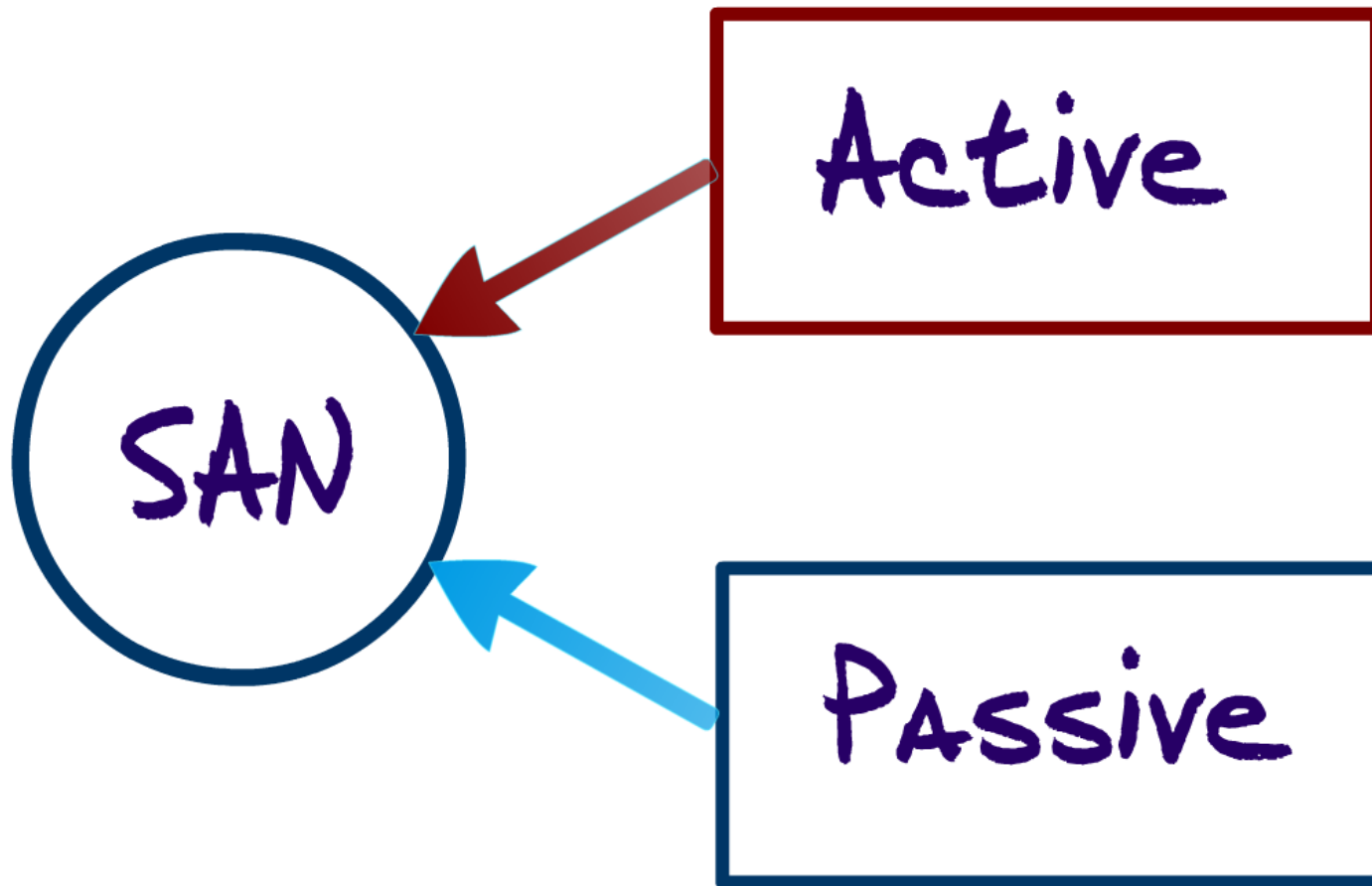




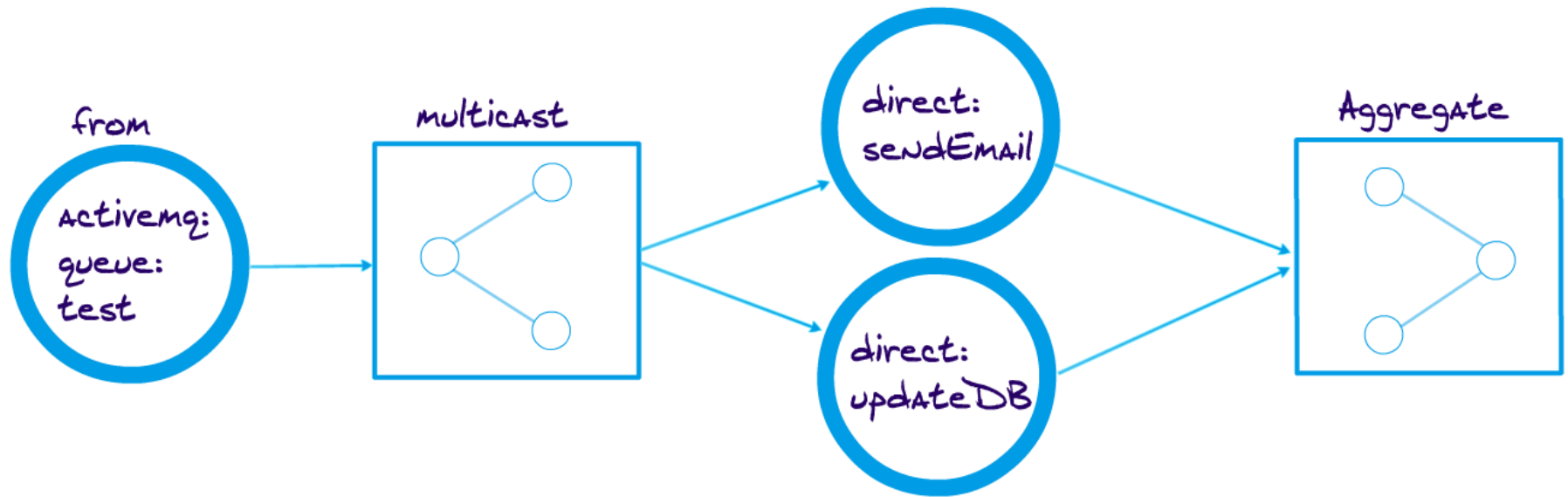
# Web Service Proxy



# High Availability



# Enterprise Service Bus



# Messaging

## Deployment Options

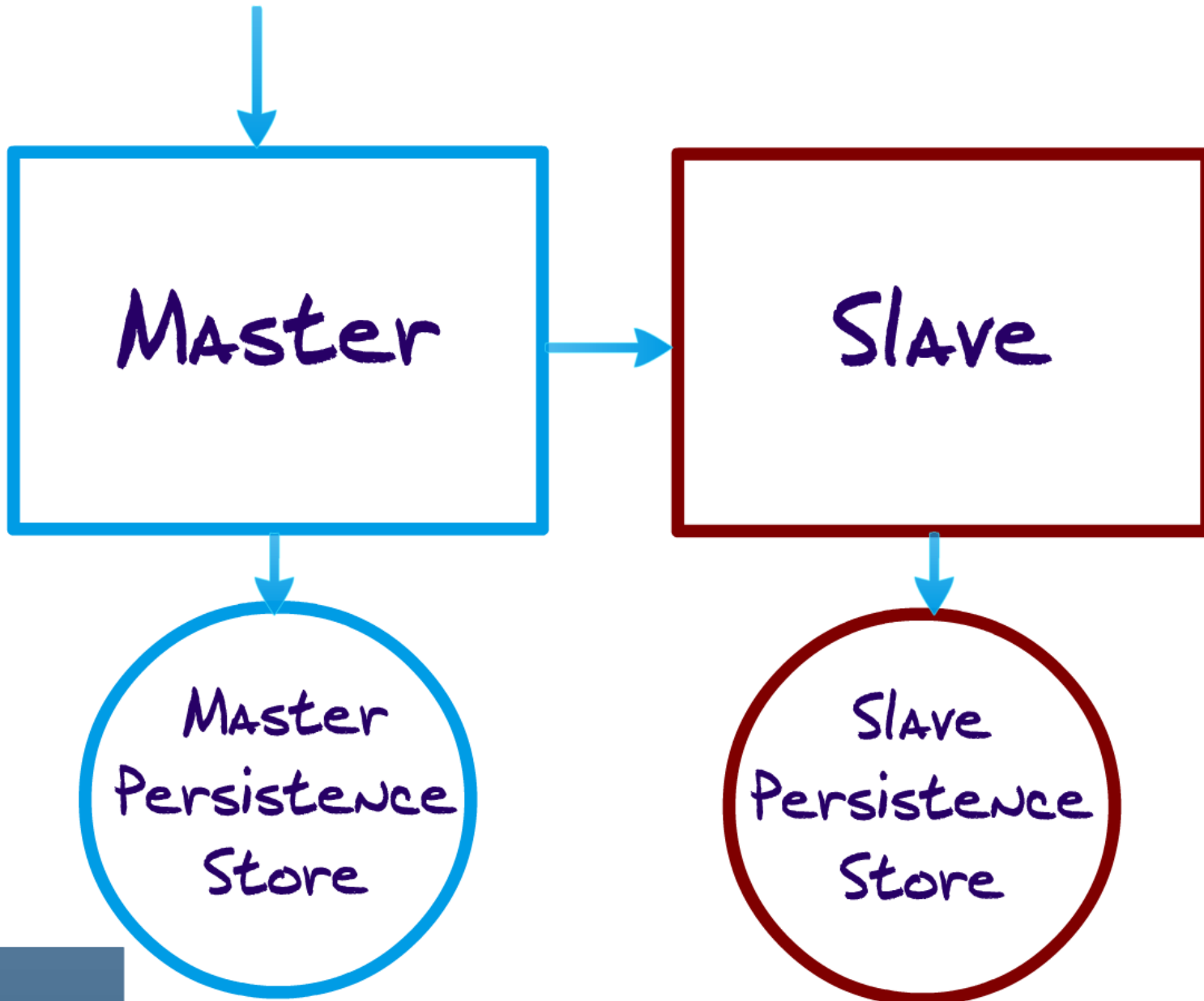
```
<broker masterConnectorURI="tcp://masterhost:61617" shutdownOnMasterFailure="false">
  <persistenceAdapter>
    <journaleJDBC journalLogFiles="5" dataDirectory="${activemq.base}/data/broker2" />
  </persistenceAdapter>
  <transportConnectors>
    <transportConnector uri="tcp://slavehost:61616"/>
  </transportConnectors>
</broker>
```

```
<persistenceAdapter>
  <kahADB directory="${activemq.base}/data/kahadb"
    checksumJournalFiles="true"
    checkForCorruptJournalFiles="true"/>
</persistenceAdapter>
```

```
<broker useJmx="false" brokerName="jdbcBroker" xmlns="http://activemq.apache.org/schema/core">
  <persistenceAdapter>
    <jdbcPersistenceAdapter dataDirectory="${activemq.base}/data" dataSource="#derby-ds"/>
  </persistenceAdapter>
  <transportConnectors>
    <transportConnector name="default" uri="tcp://0.0.0.0:61616"/>
  </transportConnectors>
</broker>

<!-- Embedded Derby DataSource Sample Setup -->
<bean id="derby-ds" class="org.apache.derby.jdbc.EmbeddedDataSource">
  <property name="databaseName" value="derbydb"/>
  <property name="createDatabase" value="create"/>
</bean>
```

# Non-shared/Pure



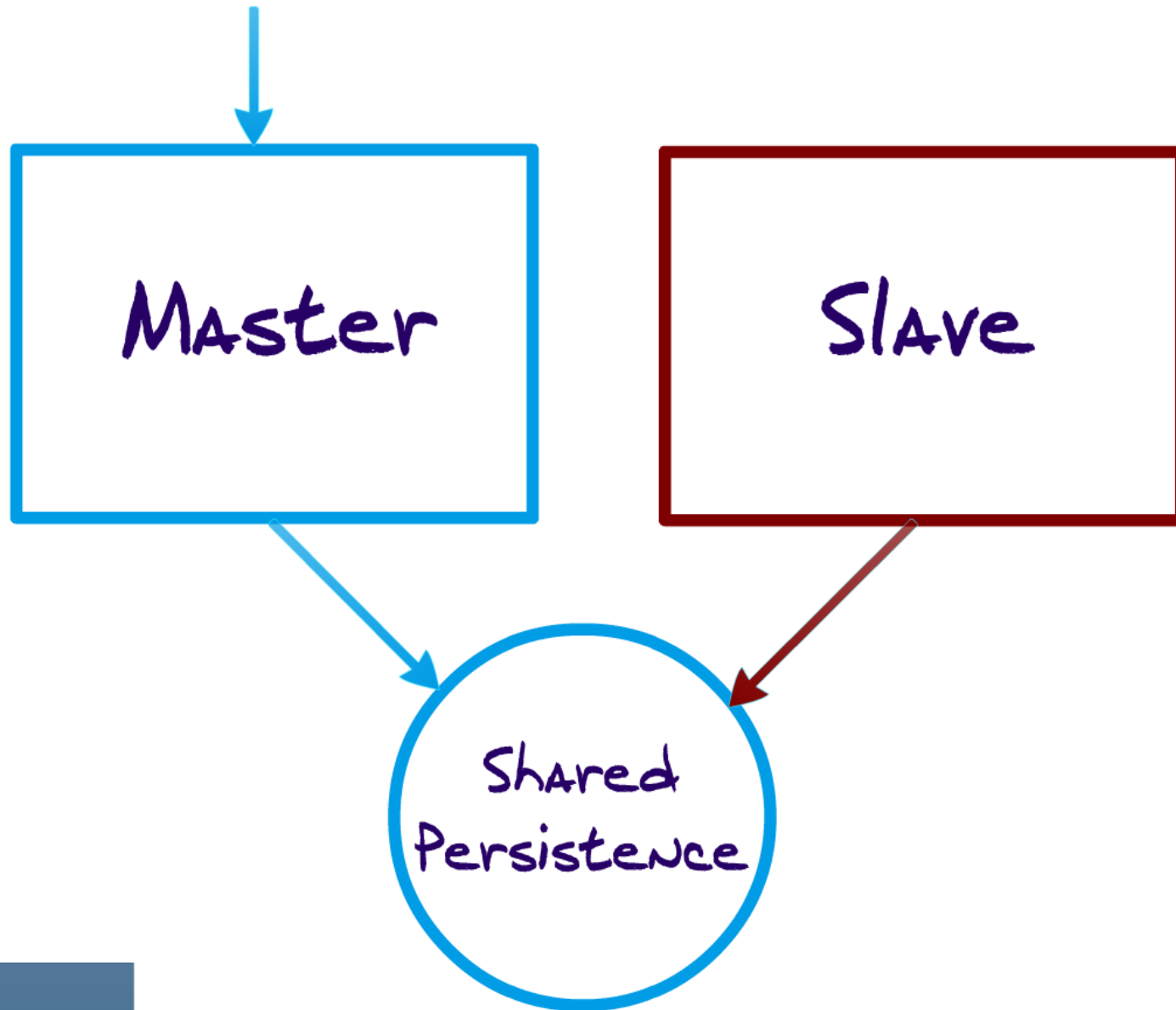


# YMENT OPTIONS

```
<broker masterConnectorURI="tcp://masterhost:61617" shutdownOnMasterFailure="false">
  <persistenceAdapter>
    <journaledJDBC journalLogFiles="5" dataDirectory="{activemq.base}/data/broker2" />
  </persistenceAdapter>

  <transportConnectors>
    <transportConnector uri="tcp://slavehost:61616" />
  </transportConnectors>
</broker>
```

# Shared File System / JDBC



```
<persistenceAdapter>
  <kahaDB directory="${activemq.base}/data/kahadb"
    checksumJournalFiles="true"
    checkForCorruptJournalFiles="true" />
</persistenceAdapter>
```

```
<broker useJmx="false" brokerName="jdbcBroker" xmlns="http://
  <persistenceAdapter>
    <jdbcPersistenceAdapter dataDirectory="${activemq.base}
  </persistenceAdapter>
```

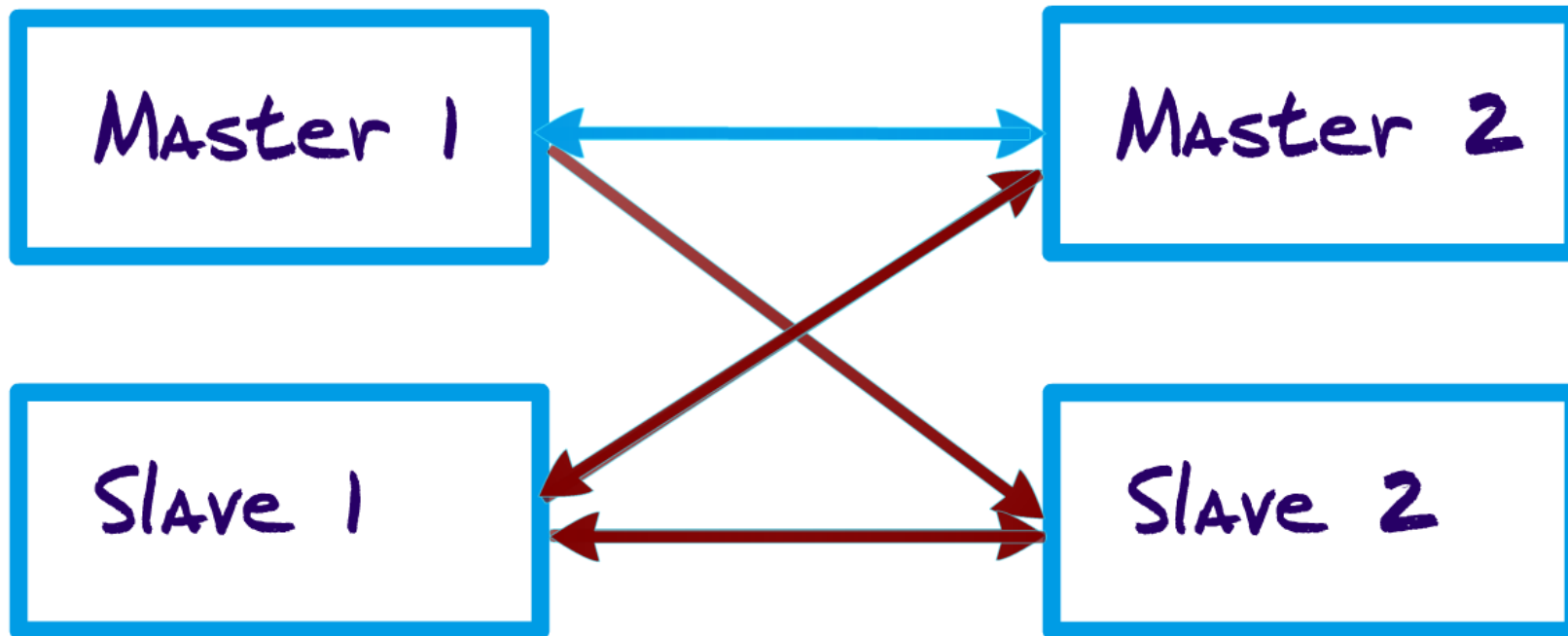
```
<persistenceAdapter>
  <kahaDB directory="${activemq.base}/data/kahadb"
    checksumJournalFiles="true"
    checkForCorruptJournalFiles="true" />
</persistenceAdapter>
```

```
<broker useJmx="false" brokerName="jdbcBroker" xmlns="http://activemq.apache.org/schema/core">
  <persistenceAdapter>
    <jdbcPersistenceAdapter dataDirectory="${activemq.base}/data" dataSource="#derby-ds" />
  </persistenceAdapter>

  <transportConnectors>
    <transportConnector name="default" uri="tcp://0.0.0.0:61616" />
  </transportConnectors>
</broker>
```

```
<!-- Embedded Derby DataSource Sample Setup -->
<bean id="derby-ds" class="org.apache.derby.jdbc.EmbeddedDataSource">
  <property name="databaseName" value="derbydb" />
  <property name="createDatabase" value="create" />
</bean>
```

# Scaling - Network of Brokers





```
<networkConnectors>
  <networkConnector uri="static:(failover:(tcp://master:61616,
                                         tcp://slave:61617))" duplex="true"/
</networkConnectors>

<persistenceAdapter>
  <kahaDB directory="{activemq.base}/data/static-broker2/kahadb" /
</persistenceAdapter>
<!-- The transport connectors ActiveMQ will listen to -->
<transportConnectors>
  <transportConnector name="openwire" uri="nio://master1:61618"
    updateClusterClients="true" rebalanceClusterClients="true"/
</transportConnectors>
```



OXF



# CXF

```
<camelcxf:cxfEndpoint id="reportDomain"  
    address="http://localhost:9080/webservices/domain"  
    wsdlURL="META-INF/wsdl/report_domain.wsdl"  
    serviceClass="com.poc.blueprint.ws.reportdomain.ReportDomainEndpoint"/>
```

```
from("cxf:bean:reportDomain")  
    .choice()  
    .when().simple("${in.header.operationName} == 'ReportDomain'")  
        .convertBodyTo(InputReportDomain.class)  
        .to("activemq:queue:domain")
```

# Aggregation/ Enrichment

```
from("activemq:queue:domain").setHeader(PROCESS_ID,constant(1)).m
```

```
from("seda:external1").to("bean:spin1").to("seda:domainAggregator
```

```
from("seda:external2").to("bean:spin2").to("seda:domainAggregator
```

```
from("seda:external3").to("bean:spin3").to("seda:domainAggregator
```

# AGGREGATION/ Enrichment

```
from("activemq:queue:domain").setHeader(PROCESS_ID,constant(1)).multicast().to("seda:ext

from("seda:external1").to("bean:spin1").to("seda:domainAggregator");
from("seda:external2").to("bean:spin2").to("seda:domainAggregator");
from("seda:external3").to("bean:spin3").to("seda:domainAggregator");

// aggregate the return values - the processId is a unique identifier used
// to associate return values with each other.
from("seda:domainAggregator")
    .aggregate(header(PROCESS_ID), new AssembleAggregationStrategy())
    .completionSize(3).to("log:dummy")
    .setHeader(Exchange.FILE_NAME, constant("response-${date:now:yyyy-MM-dd-HH:mm:ssSSS}")
    .wireTap("file://target/outbox");
```

```
public class AssembleAggregationStrategy implements AggregationStrategy {

    private static final Logger LOG = Logger.getLogger(AssembleAggregationStrategy.class.getPack

    public Exchange aggregate(Exchange oldExchange, Exchange newExchange) {
        // the first time we only have the new exchange
        if (oldExchange == null) {
            return newExchange;
        }
    }
}
```

```

from("seda:domainAggregator")
    .aggregate(header(PROCESS_ID), new AssembleAggregationStrategy())
    .completionSize(3).to("log:dummy")
    .setHeader(Exchange.FILE_NAME, constant("response-${date:now:yyyy-MM-dd-HH:mm:ssSSS}")
    .wireTap("file://target/outbox");

```

```

public class AssembleAggregationStrategy implements AggregationStrategy {

    private static final Logger LOG = Logger.getLogger(AssembleAggregationStrategy.class.getPack

public Exchange aggregate(Exchange oldExchange, Exchange newExchange) {
    // the first time we only have the new exchange
    if (oldExchange == null) {
        return newExchange;
    }
    // add the newExchange to the old exchange
    oldExchange.getIn().setBody(oldExchange.getIn().getBody(String.class) + " , " + newExcha
    String tempResponseQueueName = newExchange.getIn().getHeader(Constants.REPLY_TO_PROP, St
    String correlationId = newExchange.getIn().getHeader(Constants.CORRELATION_PROP, String.
    oldExchange.getIn().setHeader(Constants.REPLY_TO_PROP, tempResponseQueueName);
    oldExchange.getIn().setHeader(Constants.CORRELATION_PROP, correlationId);

    LOG.info("-----");
    LOG.info("[EXCHANGE MESSAGE]:[ " + newExchange.getIn().getHeader("processId") + " ] " +
    LOG.info("-----");
    return oldExchange;
}

```

UIMS

# JMS

-->

```
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cm="http://aries.apache.org/blueprint/xmlns/blueprint-cm/v1.0.0"
  xmlns:jaxws="http://cxf.apache.org/blueprint/jaxws"
  xmlns:cxf="http://cxf.apache.org/blueprint/core"
  xmlns:camel="http://camel.apache.org/schema/blueprint"
  xmlns:camelcx="http://camel.apache.org/schema/blueprint/cxf"
  xsi:schemaLocation="
    http://www.osgi.org/xmlns/blueprint/v1.0.0 http://www.osgi.org/xmlns/blueprint/v1.0
    http://cxf.apache.org/blueprint/jaxws http://cxf.apache.org/schemas/blueprint/jaxws
    http://cxf.apache.org/blueprint/core http://cxf.apache.org/schemas/blueprint/core.x
  ">

<bean id="activemq" class="org.apache.activemq.camel.component.ActiveMQComponent">
  <property name="brokerURL" value="failover:(nio://0.0.0.0:61616,nio://0.0.0.0:61617)" />
</bean>
```

```
from( "cxf:bean:reportDomain" )
  .choice()
  .when().simple( "${in.header.operationName} == 'ReportDomain'" )
    .convertBodyTo( InputReportDomain.class )
    .to( "activemq:queue:domain" )
```

```

xmlns:cxf="http://cxf.apache.org/blueprint/core"
xmlns:camel="http://camel.apache.org/schema/blueprint"
xmlns:camelcxf="http://camel.apache.org/schema/blueprint/cxf"
xsi:schemaLocation="
    http://www.osgi.org/xmlns/blueprint/v1.0.0 http://www.osgi.org/xmlns/blueprint/v1.0
    http://cxf.apache.org/blueprint/jaxws http://cxf.apache.org/schemas/blueprint/jaxws
    http://cxf.apache.org/blueprint/core http://cxf.apache.org/schemas/blueprint/core.x
">

```

```

<bean id="activemq" class="org.apache.activemq.camel.component.ActiveMQComponent">
    <property name="brokerURL" value="failover:(nio://0.0.0.0:61616,nio://0.0.0.0:61617)" />
</bean>

```

```

from( "cxf:bean:reportDomain" )
    .choice()
    .when().simple( "${in.header.operationName} == 'ReportDomain'" )
        .convertBodyTo( InputReportDomain.class )
        .to( "activemq:queue:domain" )
    .end();

```

# Testing

```

context.getRouteDefinitions().get(0).adviceWith(context, new RouteBuilder()
@Override
public void configure() throws Exception {
    // intercept sending to mock:foo and do something else
    interceptSendToEndpoint("mock:foo")
}

```



# IoC

## Blueprint

```
<?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"
  xmlns:cm="http://aries.apache.org/blueprint/xmlns/blueprint-cm/v1.0.0"
  xmlns:jaxws="http://cxf.apache.org/blueprint/jaxws"
  xmlns:cxf="http://cxf.apache.org/blueprint/cxf"
  xmlns:camel="http://camel.apache.org/schema/blueprint"
  xmlns:camelcxf="http://camel.apache.org/schema/blueprint/cxf"
  xsi:schemaLocation="
    http://www.osgi.org/xmlns/blueprint/v1.0.0 http://www.osgi.org/xmlns/blueprint/v1.0
    http://cxf.apache.org/blueprint/jaxws http://cxf.apache.org/schemas/blueprint/jaxws
    http://cxf.apache.org/blueprint/cxf http://cxf.apache.org/schemas/blueprint/core.x
">
  <camelcxf:cxfEndpoint id="reportDomain"
    address="/web-services-blueprint-domain/webservices/domain"
    wsdlURL="META-INF/wsdl/report_domain.wsdl"
    serviceClass="com.poc.blueprint.ws.reportDomain.ReportDomainEndpoint"/>
  <bean id="domainRoutes" class="com.poc.blueprint.ws.routes.DomainRoutes"/>
  <camelContext id="camel" xmlns="http://camel.apache.org/schema/blueprint">
    <routeBuilder ref="domainRoutes"/>
  </camelContext>
</blueprint>
```

## Spring

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:camel="http://camel.apache.org/schema/spring"
  xmlns:cxf="http://camel.apache.org/schema/cxf"
  xmlns:p="http://www.springframework.org/schema/p"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans
    http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spr
    http://camel.apache.org/schema/cxf http://camel.apache.org/schema/cxf/camel-cxf.xsd">
  <import resource="classpath:META-INF/cxf/cxf.xml"/>
  <import resource="classpath:META-INF/cxf/osgi/cxf-extension-osgi.xml"/>
  <cxf:cxfEndpoint id="reportDomain"
    address="http://localhost:9090/domain"
    wsdlURL="META-INF/wsdl/report_domain.wsdl"
    serviceClass="com.poc.spring.cxf.reportDomain.ReportDomainEndpoint"/>
  <bean id="reportDomainRoutes" class="com.poc.spring.cxf.routes.DomainRoutes"/>
  <camel:camelContext id="camel-spring">
    <camel:routeBuilder ref="reportDomainRoutes"/>
  </camel:camelContext>
</beans>
```

# Blueprint

```
<?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"
  xmlns:cm="http://aries.apache.org/blueprint/xmlns/blueprint-cm/v1.0.0"
  xmlns:jaxws="http://cxf.apache.org/blueprint/jaxws"
  xmlns:cxf="http://cxf.apache.org/blueprint/core"
  xmlns:camel="http://camel.apache.org/schema/blueprint"
  xmlns:camelcxf="http://camel.apache.org/schema/blueprint/cxf"
  xsi:schemaLocation="
    http://www.osgi.org/xmlns/blueprint/v1.0.0 http://www.osgi.org/xmlns/blueprint/v1.0
    http://cxf.apache.org/blueprint/jaxws http://cxf.apache.org/schemas/blueprint/jaxws
    http://cxf.apache.org/blueprint/core http://cxf.apache.org/schemas/blueprint/core.x
">

  <camelcxf:cxfEndpoint id="reportDomain"
    address="/web-services-blueprint-domain/webservices/domain"
    wsdlURL="META-INF/wsdl/report_domain.wsdl"
    serviceClass="com.poc.blueprint.ws.reportdomain.ReportDomainEndpoint"/>

  <bean id="domainRoutes" class="com.poc.blueprint.ws.routes.DomainRoutes"/>

  <camelContext id="camel" xmlns="http://camel.apache.org/schema/blueprint">
    <routeBuilder ref="domainRoutes"/>
  </camelContext>

</blueprint>
```

```
http://www.osgi.org/xmlns/blueprint/v1.0.0 http://www.osgi.org/x
http://cxf.apache.org/blueprint/jaxws http://cxf.apache.org/sche
http://cxf.apache.org/blueprint/core http://cxf.apache.org/schem
">
```

```
<camelcxf:cxfEndpoint id="reportDomain"
    address="/web-services-blueprint-domain/webservices/c
    wsdlURL="META-INF/wsdl/report_domain.wsdl"
    serviceClass="com.poc.blueprint.ws.reportdomain.Repor
```

```
<bean id="domainRoutes" class="com.poc.blueprint.ws.routes.DomainRoutes"/>
```

```
<camelContext id="camel" xmlns="http://camel.apache.org/schema/blueprint">
    <routeBuilder ref="domainRoutes"/>
</camelContext>
```

```
</blueprint>
```

# Spring

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:camel="http://camel.apache.org/schema/spring"
  xmlns:cxf="http://camel.apache.org/schema/cxf"
  xmlns:p="http://www.springframework.org/schema/p"

  xsi:schemaLocation="
    http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans
    http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring
    http://camel.apache.org/schema/cxf http://camel.apache.org/schema/cxf/camel-cxf.xsd">

  <import resource="classpath:META-INF/cxf/cxf.xml" />
  <import resource="classpath:META-INF/cxf/osgi/cxf-extension-osgi.xml" />

  <cxf:cxfEndpoint id="reportDomain"
    address="http://localhost:9080/domain"
    wsdlURL="META-INF/wsdl/report_domain.wsdl"
    serviceClass="com.poc.spring.cxf.reportdomain.ReportDomainEndpoint" />

  <bean id="reportDomainRoutes" class="com.poc.spring.cxf.routes.DomainRoutes" />

  <camel:camelContext id="camel-spring">
    <camel:routeBuilder ref="reportDomainRoutes" />
  </camel:camelContext>
```

# ESB



## Deployment



## Monitoring/Logging

- JMX
  - JConsole
  - JVisualVM
  - JMX Console
  - JMX API
  - Web Console
- Logging

Camel logging: `set(loglevel)`  
 Framework Logging: `log4j`  
 Karaf Console Logging: `logset DEBUG "package.level"`

- AMQ Advisories

Camel Routes  
 Consumer  
 Producer  
 Connection  
 TQLQ  
 Memory Consumption  
 more.....

## Scale Down

```

Remove...
=====
Licensed to the Apache Software Foundation, ASF, under one or more
contributor licenses. See the NOTICE file distributed with this work
for additional information regarding copyright ownership. ASF
licenses this file to you under the Apache License, Version 2.0
(https://www.apache.org/licenses/LICENSE-2.0), with the exception of
the license, you may obtain a copy of the license at
https://www.apache.org/licenses/LICENSE-2.0
=====
License required by applicable law or agreed to in writing, requires
written permission for redistribution of any code, either in source or
binary form, without specific language governing permissions and
restrictions in writing.
=====
Some required list of features requested or required by default.
features:org.apache.karaf.features:karaf-core:1.2.4/maven:amino
features:org.apache.karaf.features:karaf-core:1.2.4/maven:amino
features:org.apache.karaf.features:karaf-core:1.2.4/maven:amino
=====

```

# KARAF!



# ServiceMix?



# KARAK!

Console

Logging

Deployer

Provision

Admin

OSGI

# Service Mi

JMS

JAX-WS

JAX-RS

Camel

Spring

JB1

KARAF



```
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
```

```
#
# http://www.apache.org/licenses/LICENSE-2.0
```

```
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

```
#####
```

```
#
# Comma separated list of features repositories to register by default
```

```
#
featuresRepositories=mvn:org.apache.karaf.assemblies.features/standard/2.2.4/xml/features,mvn:or
```

```
#
# Comma separated list of features to install at startup
```

```
#
featuresBoot=karaf-framework,config,cxf-specs,camel,camel-activemq,camel-cxf,camel-blueprint,war
```

# Deployment

SMX Proxy 1

SMX Proxy 2

AMQ 1

AMQ 2

SMX Logic 1

SMX Logic 2

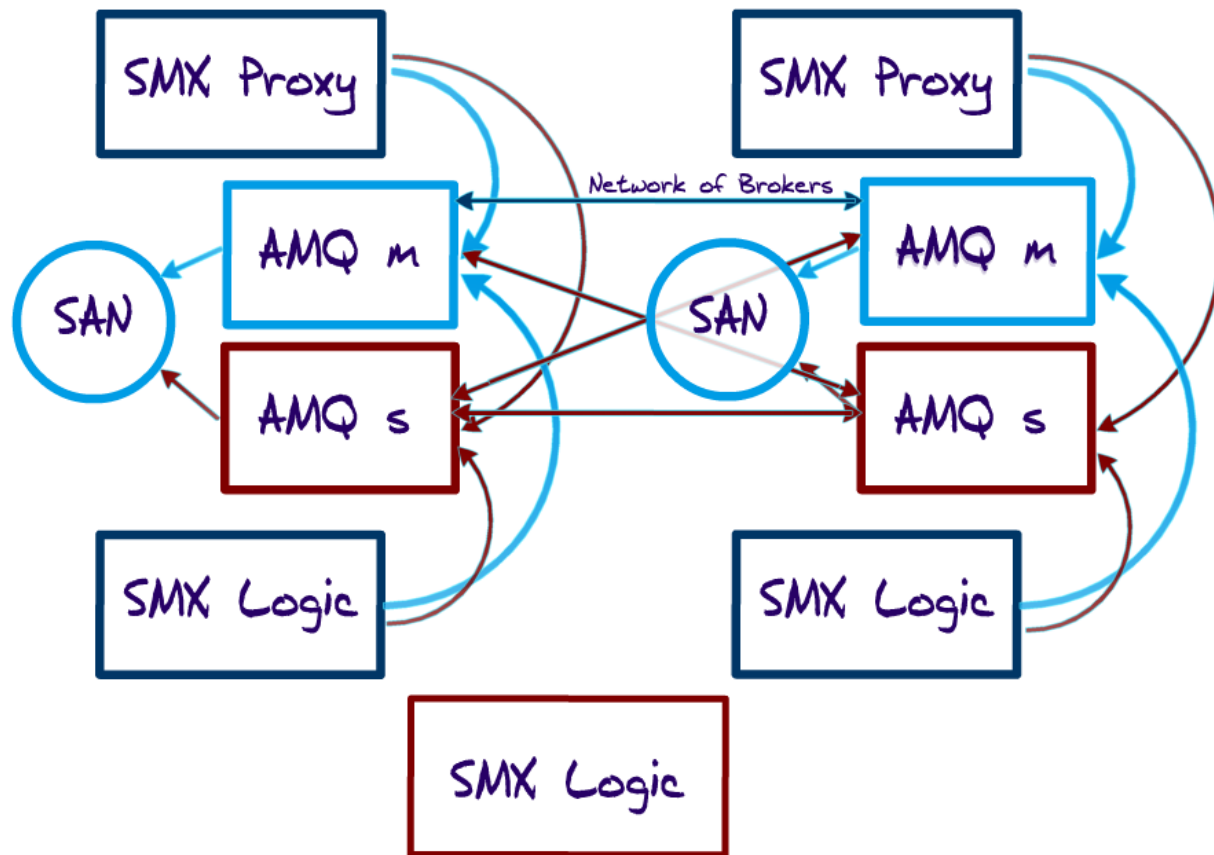
SMX Logic 3

Proxy Layer

Messaging Layer

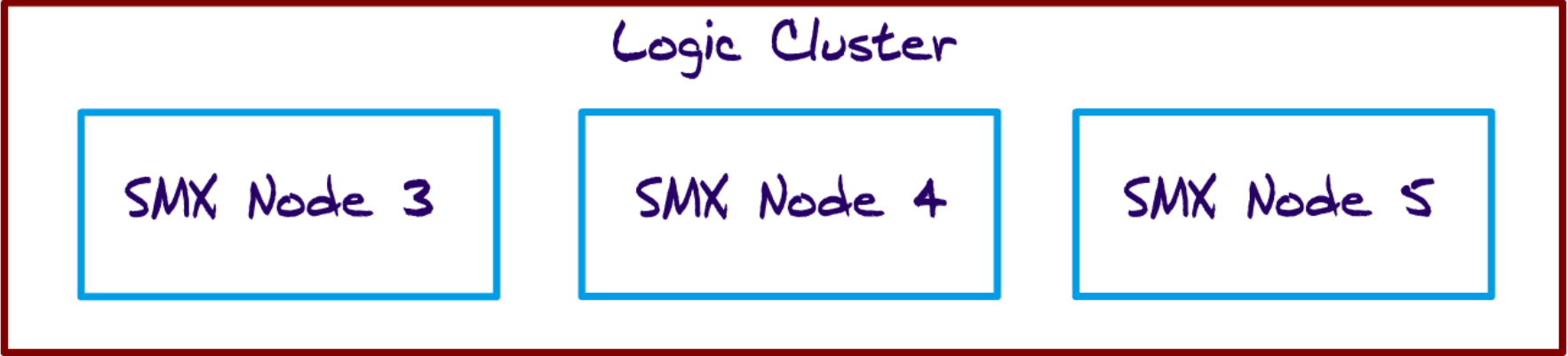
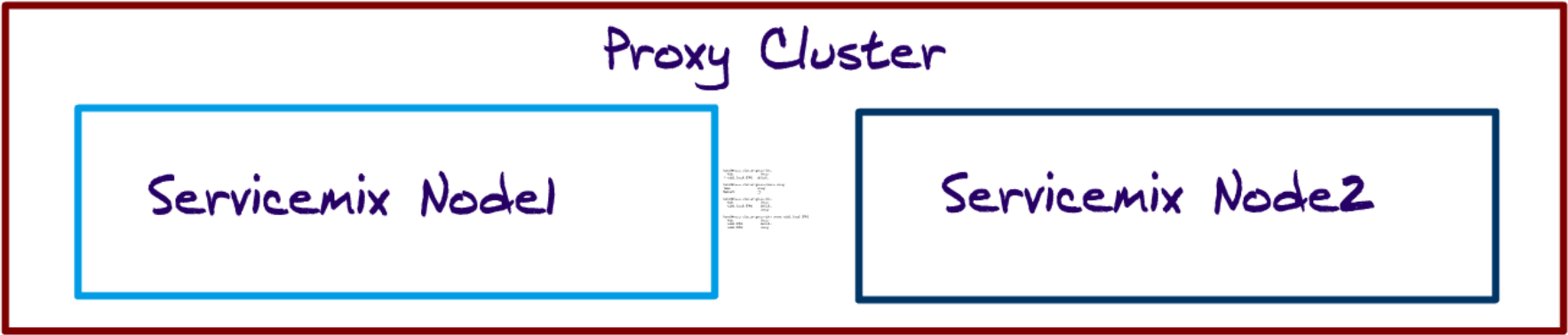
Business Logic Layer

# Hardware Config



# Clustering

© 2005 Sun Microsystems, Inc. All rights reserved. Sun, the Sun logo, Java, the Java logo, and J2SE are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.





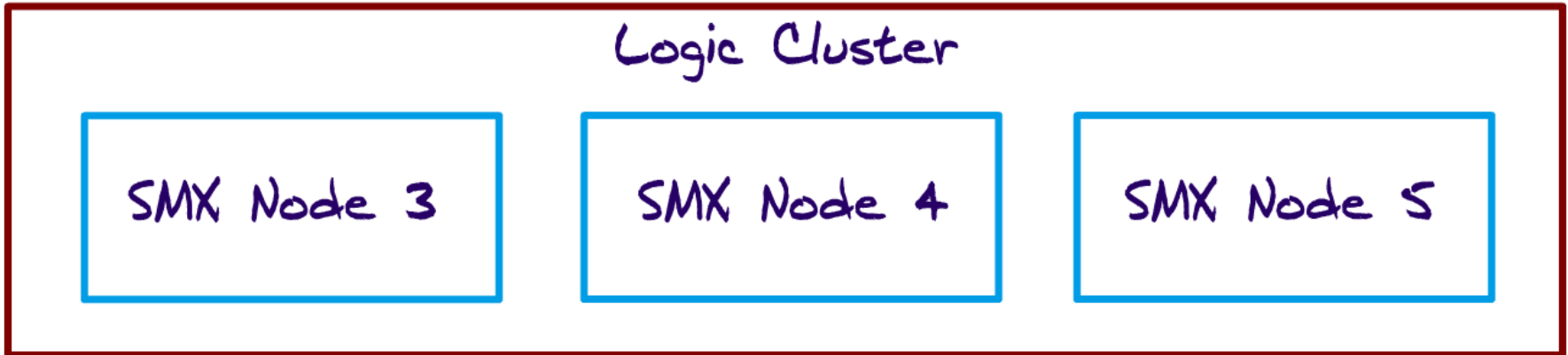
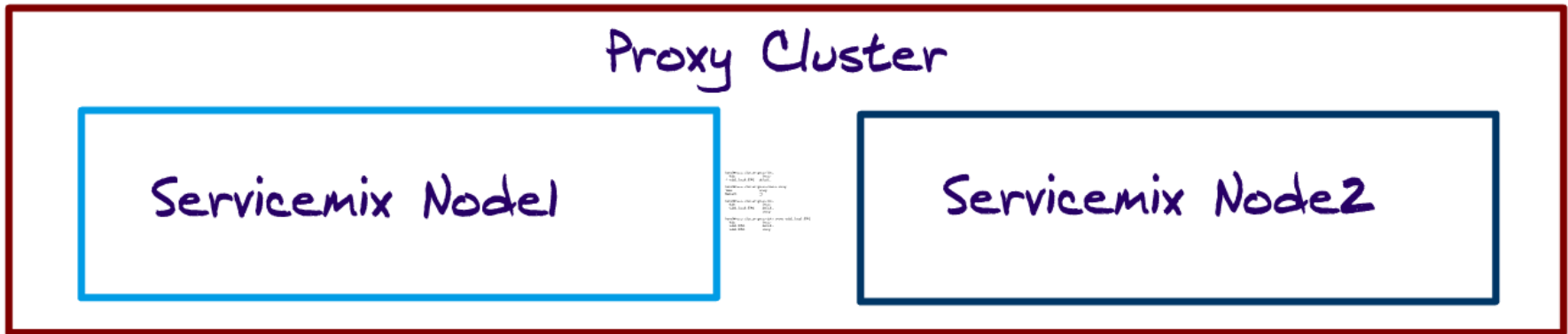
---

```
features:addurl mvn:org.apache.karaf.cellar/apache-karaf-cellar/2.2.1/xml/fe
```

```
features:install cellar
```

# Clustering

© 2008 Oracle and/or its affiliates. All rights reserved. Oracle, Java, and the Java logo are trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.



```
karaf@root> cluster:group-list
```

Node	Group
* node1.local:5701	default

```
karaf@root> cluster:group-create proxy
```

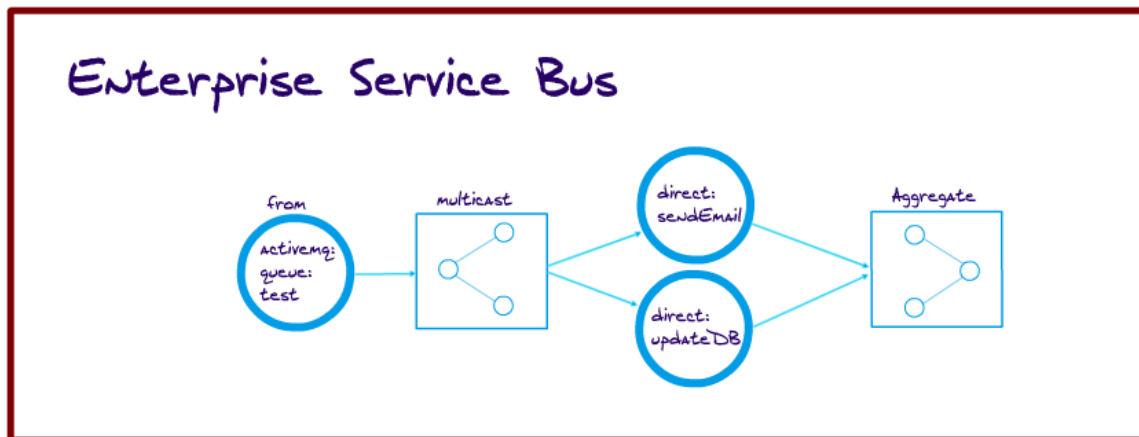
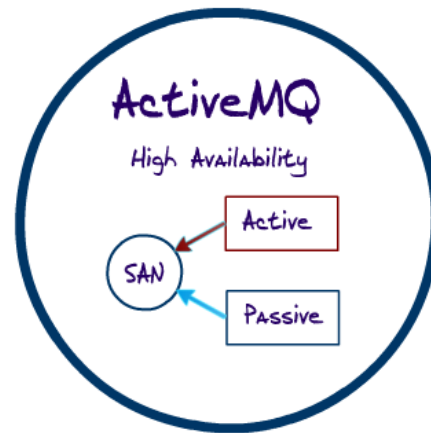
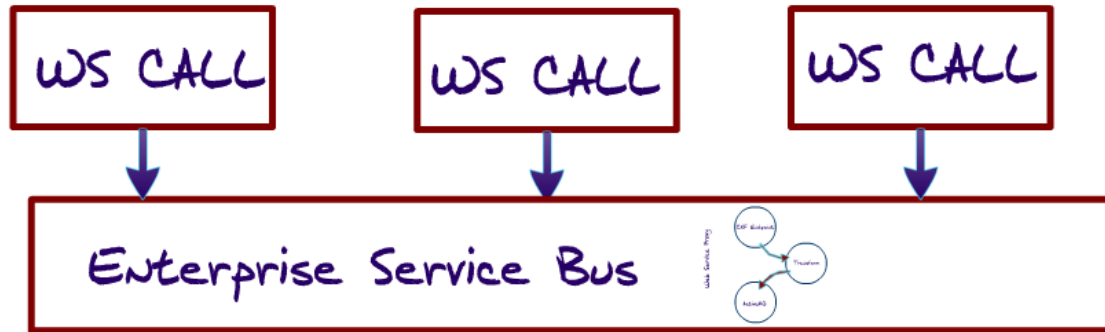
Name	proxy
Members	[]

```
karaf@root> cluster:group-list
```

Node	Group
node1.local:5701	default
	proxy

```
karaf@root> cluster:group-join proxy node1.local:5701
```

Node	Group
node1:5701	default
node1:5701	proxy





# Example

*Let's run it*



**Conclusion**

hkesler@savoirtech.com

**Exam**