

Service Mix 4 Topologies

Andreas Gies

Principal Architect

<http://www.fusesource.com>

<http://open-source-adventures.blogspot.com>

About the Author

- Principal Architect
PROGRESS - Open Source Center of Competence
- Degree in Computer Science from the University of the German Forces 1992
- Working with middleware like MOMs, CORBA, J2EE, WS and ESBs ever since for Sterling Software, Iona Technologies and PROGRESS
- Specialized on ESB based architectures since 2002



Software used for the examples

- The examples are based on the **FUSE** releases of Apache ServiceMix and Apache ActiveMQ
- The FUSE community provides
 - **Access to Committers** as many Apache committers are employed by the FUSE team
 - **Enterprise support** – Open source adaption in the enterprise requires 24x7 reliable support
 - **Increased testing** on a CI environment maintained by the FUSE team
 - **Enterprise qualities of service** – Ensuring sensible Enterprise deployment and backwards compatibility
 - **Documentation and training** for the Apache projects released under the FUSE brand
 - **Backed by large, enterprise company**

3

FUSE products

- **FUSE ESB 3**
Based on Apache Service Mix 3
- **FUSE ESB 4**
Based on Apache Service Mix 4
- **FUSE Message Broker**
Based on Apache ActiveMQ
- **FUSE Services Framework**
Based on Apache CXF
- **FUSE Mediation Router**
Based on Apache Camel
- **FUSE Integration Designer**
Eclipse tooling for implementing EIP flows
- **FUSE HQ**
Management and Monitoring of the FUSE infrastructure

4

Agenda

- Why look at topologies
- FUSE ESB 4 default topology
- Geographic distribution
- Scaling for large message volumes
- High Availability considerations
- Conclusion & References
- Q & A

5

Why do we look at topologies ?

- Topology – the physical layout of a FUSE ESB 4 deployment
- Requirements of scalability, fault tolerance and reach can be addressed by using multiple FUSE ESB 4 instances
- Patterns are required to achieve an extensible, maintainable and deployable topology
- The patterns involve naming conventions and required configuration elements

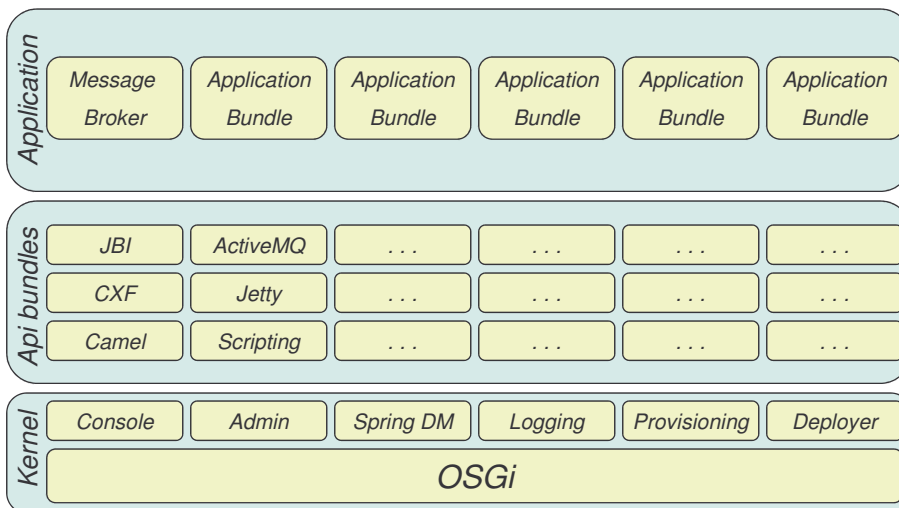
6

Agenda

- Why look at topologies
- FUSE ESB 4 default topology
- Geographic distribution
- Scaling for large message volumes
- High Availability considerations
- Conclusion & References
- Q & A

7

FUSE ESB 4 default topology



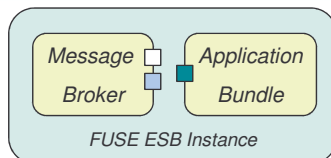
8

FUSE ESB 4 default topology

- OSGi Kernel with standard extensions for logging, Spring support, Administration, Provisioning, Deployment and an extensible shell
- API bundles can be installed if they are required by the applications on the node
 - Typically provide OSGi wrapped API libraries like Camel, CXF, ActiveMQ etc.
- API bundles are typically provided by a provisioning infrastructure based on Maven
- Application bundles contain the communication endpoints and integration flows
- The Message broker configuration is also placed in the application layer as this is one key element investigating different communication topologies

9

FUSE ESB 4 default topology - simplified



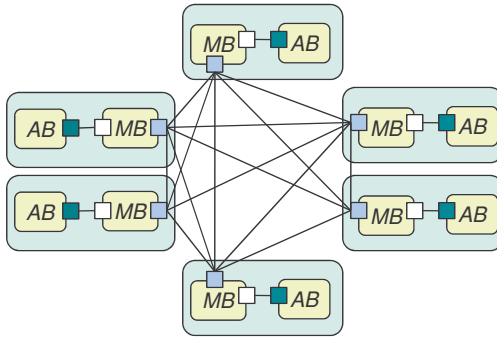
Symbol	Usage
□	The <u>Transport Connector</u> of the message brokers used by messaging clients
■	The <u>Network Connector</u> of the message brokers are used to interconnect brokers
■	The <u>Client Connectors</u> are JMS connections that provide JMS connectivity to the application bundles

We will concentrate on JMS topologies with the FUSE ESB

10

FUSE ESB 4 default topology – multiple nodes

Network of Nodes



- Most simple setup
- Established with a simple Multicast Network connector
- Requires a multicast connection for dynamic discovery

11

FUSE ESB 4 default topology – multiple nodes

Network of Nodes

Within broker configuration on each node

- *define a discoveryUri for the transportConnector*
- *define a multicast network connector for that uri*

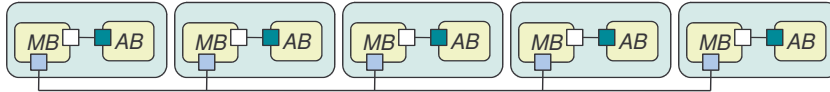
```
<transportConnectors>
  <transportConnector
    uri="tcp://scrooged:61616"
    discoveryUri="multicast://default" />
</transportConnectors>

<networkConnectors>
  <networkConnector uri="multicast://default" />
</networkConnectors>
```

12

FUSE ESB 4 default topology – multiple nodes

List of Nodes



- Requires a specific setup for each node to connect only to the peer broker
- Connector can be a normal tcp or multicast connection
- Requires a multicast connection for dynamic discovery

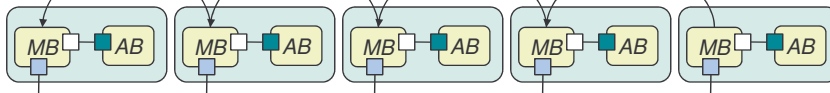
13

FUSE ESB 4 default topology – multiple nodes

List of Nodes

Easiest to establish with static network connection from each successor to it's predecessor in the list

Important to set the duplex property to true

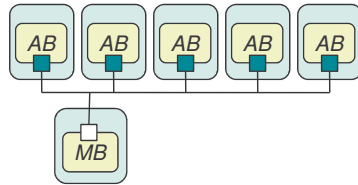


```
<transportConnectors>
  <transportConnector
    uri="tcp://scrooged:61616"/>
</transportConnectors>

<networkConnectors>
  <networkConnector
    uri="tcp://donald:61616" duplex="true"/>
</networkConnectors>
```

FUSE ESB 4 lean messaging set up

Lean Messaging



- Requires a message broker setup in only one node
- Application Bundle containers are not longer autonomous in terms of the communication infrastructure
- Does not require a network connector setup
- Is a classical “hub and spoke” topology

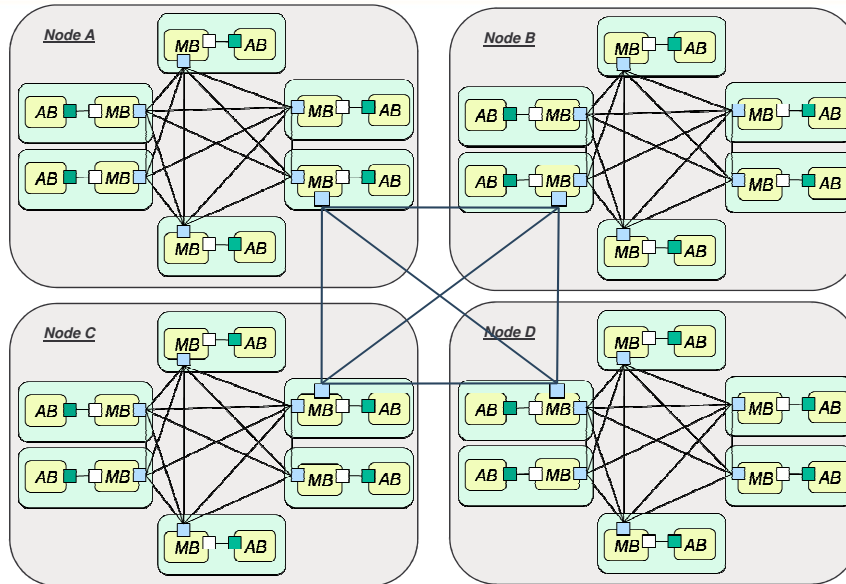
15

Agenda

- Why look at topologies
- Service Mix 4 default topology
- Geographic distribution
- Scaling for large message volumes
- High Availability considerations
- Conclusion & References
- Q & A

16

Geographically distributing FUSE ESB Nodes



17

Geographically distributed FUSE ESB nodes

- Within each node configure the brokers and restrict multicast discovery to a group corresponding to the node

```
<transportConnectors>
  <transportConnector
    uri="tcp://scrooged:61616"
    discoveryUri="multicast://224.1.2.3?group=NodeA" />
</transportConnectors>

<networkConnectors>
  <networkConnector uri="multicast://224.1.2.3?group=NodeA" />
</networkConnectors>
```

18

Geographically distributed FUSE ESB nodes

- Within the connecting Brokers add a transport connector with a global discoveryUri
- Also add a network connector connecting to that discoveryUri

```
<transportConnectors>
  <transportConnector
    uri="tcp://scrooged:61616"
    discoveryUri="multicast://224.1.2.3:6256?group=NodeA" />
  <transportConnector
    uri="tcp://scrooged:61618"
    discoveryUri="multicast://224.1.2.3:6255?group=Global" />
</transportConnectors>

<networkConnectors>
  <networkConnector
    uri="multicast://224.1.2.3:6256?group=NodeA" />
  <networkConnector
    uri="multicast://224.1.2.3:6255?group=Global" />
</networkConnectors>
```

19

Agenda

- Why look at topologies
- FUSE ESB 4 default topology
- Geographic distribution
- Scaling for large message volumes
- High Availability considerations
- Conclusion & References
- Q & A

20

Scaling Services

- Normally services can work as concurrent consumers on the same JMS queue
- The message brokers will distribute the messages across all service instances
- Message ordering might be important
 - For all messages within a given destination
=>Only one service is allowed to be active at any given time
 - Only for groups of messages sent to a given queue
=>Use JMS Message groups in that case

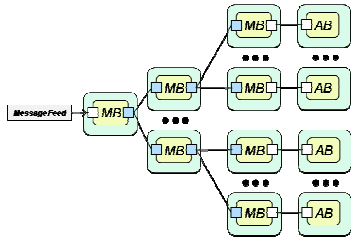
21

Scaling the messaging layer

- A normal Network of Nodes or List of Nodes can be used to address the messaging throughput
- Messaging throughput and latency might be concurrent goals
- Normally a Network of Nodes yields in lower message latency
- A List of Nodes is more a bus like Architecture with a minimal number of connections and low message volumes

22

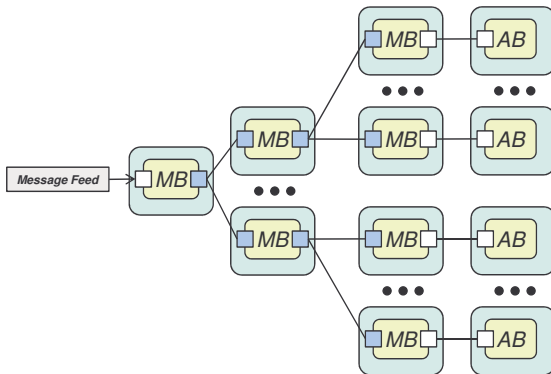
Fan out strategy



- Used to push out a large number of events to a large number of consuming services
- A typical fan out factor is 1:50
- The configuration should be done using static routes from the successors to their predecessors
- The message broker in the last tree level is optional

23

Fan out strategies



24

Agenda

- Why look at topologies
- Service Mix 4 default topology
- Geographic distribution
- Scaling for large message volumes
- High Availability considerations
- Conclusion & References
- Q & A

25

Highly available services

- Normally multiple instances of a service can be deployed to achieve fault tolerance
- Exclusive consumers may be used in cases where only a single Service is allowed at any point in time
- JMS Message groups will guarantee that all messages within one group are processed by the same Service instance

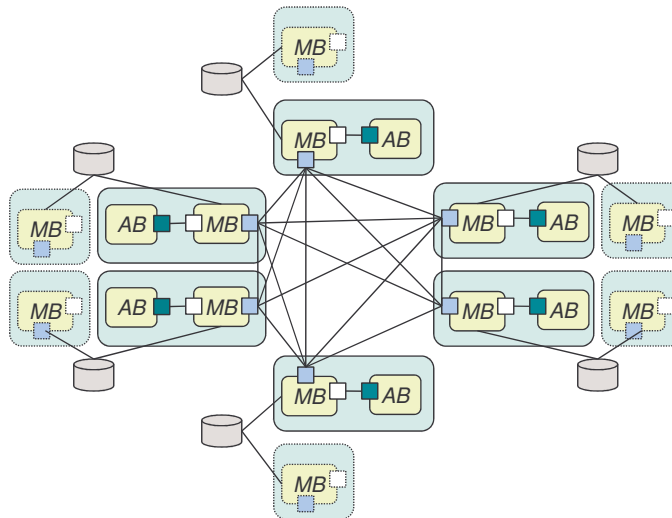
26

High Availability in the messaging layer

- The topologies discussed provide a certain level of fault tolerance already
- Sometimes it is required that one broker can take over seamlessly from another broker
 - This can be achieved by configuring a master slave configuration
 - Master/Slave pairs can then be combined into a network of Master/Slave brokers
 - Network of Master/Slaves should be using multicast discovery for ease of configuration
 - Easiest configuration was a shared directory between the Master and the Slave

27

High Availability in the messaging layer



28

Agenda

- Why look at topologies
- Service Mix 4 default topology
- Geographic distribution
- Scaling for large message volumes
- High Availability considerations
- Conclusion & References
- Q & A

29

Conclusion

- The FUSE ESB 4 default topology provides a quick start into the development and deployment of OSGi based Services
- We have shown patterns that address more throughput, message latency or fault tolerance
- Combining those patterns as shown in the geographic distribution sample allows for very flexible enterprise wide deployments
- It is important to sketch out the deployment and define naming conventions for all components deployed

30

References

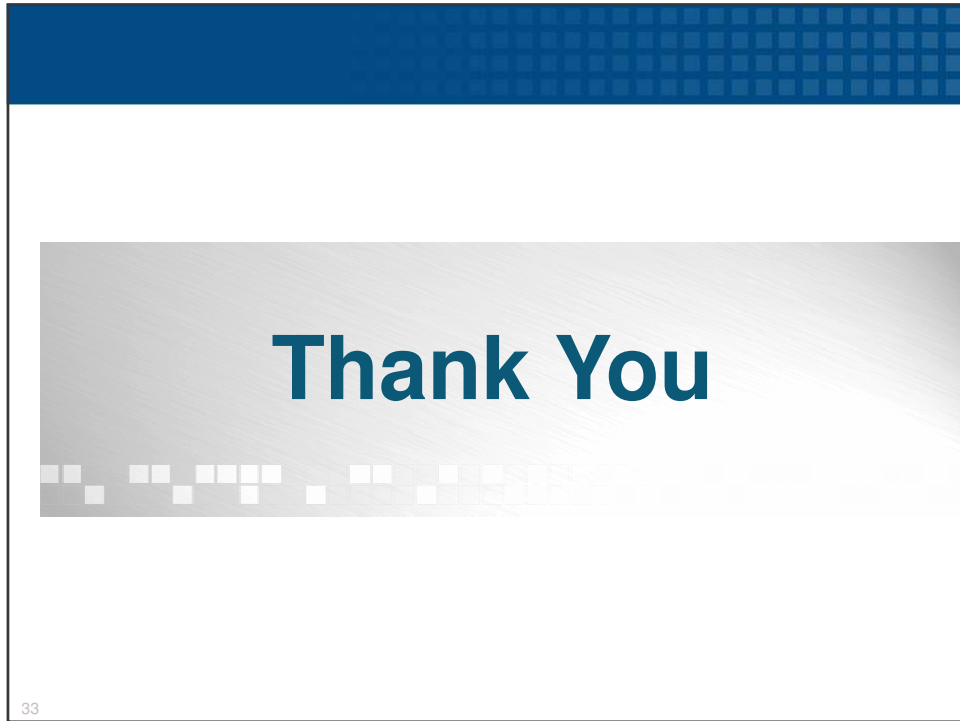
- The Apache ServiceMix web site
<http://servicemix.apache.org>
- The FUSE ESB web site
<http://fusesource.com/products/enterprise-servicemix4/>
- The Apache ActiveMQ web site
<http://activemq.apache.org>
- The FUSE Message Broker web site
<http://fusesource.com/products/enterprise-activemq/>
- The FUSE community
<http://www.fusesource.com>

31

Questions ?

Come and talk to us at <http://fusesource.com/forums/index.jspa>

32



33