Complex Event Processing with Esper and WSO2 ESB

Paul Fremantle, CTO, WSO2

29th July 2008

ESB 1.7 Webinar Series

- June 17th – Introducing WSO2 ESB 1.7
- July 1st – Example Integration Scenarios
- July 3rd – Introducing WSO2 ESB 1.7 (repeat)
- July 15th – Using WSO2 ESB with FIX
- **July 29th - Complex Event Processing**

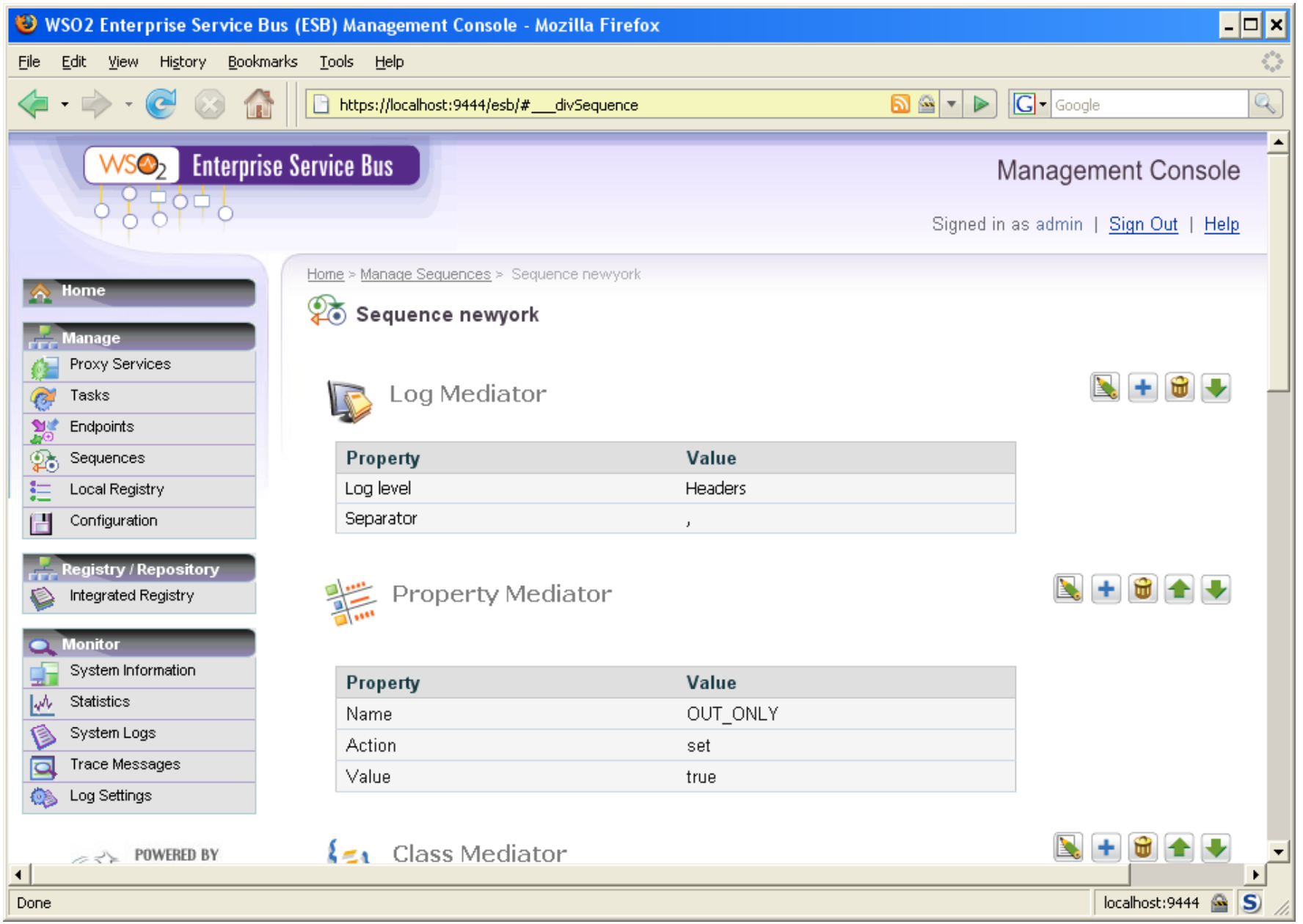
WSO2 ESB

- Based on the Apache Synapse ESB
  - Class leading performance
  - Non-blocking, streaming support
  - Routing, mediation, management, clustering
- WSO2 ESB adds:
  - Web-based management console
  - Integrated FIX support, Smooks library
  - Integrated Registry and Repository
  - Full commercial support subscription



WSO2 ESB

Configured through the simple UI



What is Complex Event Processing?

- Simple Event Processing
  - Acting on single events
  - e.g. a <filter> in the ESB
  - Is this a gold or platinum customer?
- Event Stream Processing
  - Looking across multiple events
  - Finding patterns – e.g. the CPU utilization has been more than 90% for the last 10 minutes
- Complex Event Processing
  - Looking across multiple event streams
  - e.g There has been a significant increase in overall trading activity AND the average price of commodities has fallen 2% in the last 4 hours

Complex Event Processing in Open Source

- Esper
  - http://esper.codehaus.org
  - A Java library that can be integrated into multiple systems
  - A GPLv2 project
    - (with a commercial edition available)
  - Supports multiple query models
    - Based on a SQL-like language
    - Grouping, aggregation, sorting, filtering and merging of event streams
    - Windows based on time, length, sorted, and others
  - Events can be XML, Map, Object

Esper in action

- In this presentation we mainly use the XML configuration and event models of Esper
- There are also Java API equivalents

Esper Configuration

- XML shape:

```
<tick xmlns="http://mycom.com/ns/tick">
    <s>IBM</s>
    <p>89.2</p>
</tick>
```

- Esper config:

```
<event-type alias="Ticker">
 <xml-dom root-element-name="tick"
      default-namespace="http://mycom.com/ns/tick">
      <xpath-property property-name="symbol"
                      xpath="/tick/s" type="string"/>
      <xpath-property property-name="price"
                      xpath="/tick/p" type="number"/>
 </xml-dom>
</event-type>
```

Esper continued

- Event Processing Language (EPL)
  - Example:

```
SELECT symbol, AVG(price) FROM Ticker
  GROUP BY symbol
```

- Loosely based on SQL
  - But with the concept of Windows
    - Time-based windows
    - Length based windows
    - Batch – one event per x time or events
- Lots of options

Esper support for Axiom

- Since Esper 2.1.0
- Pluggable event types
  - Allows alternative representations of events
  - The first one is Axiom
- Based on a resolveURI concept
  - Each event type has a URI

Esper config for Axiom

```
<esper-configuration
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.espertech.com/schema/esper"
   xsi:noNamespaceSchemaLocation="esper-configuration-2-0.xsd">
    <plugin-event-representation
       uri="type://xml/apacheaxiom/OMNode"
      class-
name="com.espertech.esperio.representation.axiom.AxiomEventRepresentatio
    <plugin-event-type alias="Ticker">
        <resolution-uri value="type://xml/apacheaxiom/OMNode"/>
        <initializer>
            <xml-axiom root-element-name="getQuote"
                default-namespace="http://mycom.com/ns/tick">
                <namespace-prefix prefix="t"
                    namespace="http://mycom.com/ns/tick"/>
                <xpath-property property-name="symbol"
                    xpath="//t:tick/t:s" type="string"/>
                <xpath-property property-name="price"
                    xpath="//t:tick/t:p" type="number"/>
            </xml-axiom>
        </initializer>
    </plugin-event-type>
</esper-configuration>
```

Why use Esper with Synapse/WSO2 ESB?

- Synapse/WSO2 ESB provide
  - A unified transport model
    - HTTP, JMS, XMPP, Atom, TCP, etc
  - Transformation
    - Can manipulate events as they are emitted from Esper and reformat to match a third-party system
  - Simple wiring
- Some examples:
  - Proxy SOAP/HTTP events and also send interesting events to a third system
  - Take JMS ticker events and publish a 10 min batch summary as an Atom feed

Events in WSO2 ESB

- WSO2 ESB supports multiple event models
  - Any message in the ESB can be sent to Esper as an Event
  - JMS Publish/Subscribe
  - XMPP transport = Jabber
  - Atom add-in:
    http://esbsite.org/resources.jsp?path=/mediators/upul/Atom%20Mediator
  - Ongoing work towards WS-Eventing, Notification

Esper Axiom Mediator

- Originally written July 2007
- New project “Sci-Flex”
  - Student project to improve the work
  - code.google.com/sci-flex/
- Now uses Axiom support within Esper
- GPLv2 Licensed
- You can get it today from:
  - https://wso2.org/repos/wso2/people/paul/AxiomEsperMediator/
  - Will eventually be downloadable from Sci-Flex site

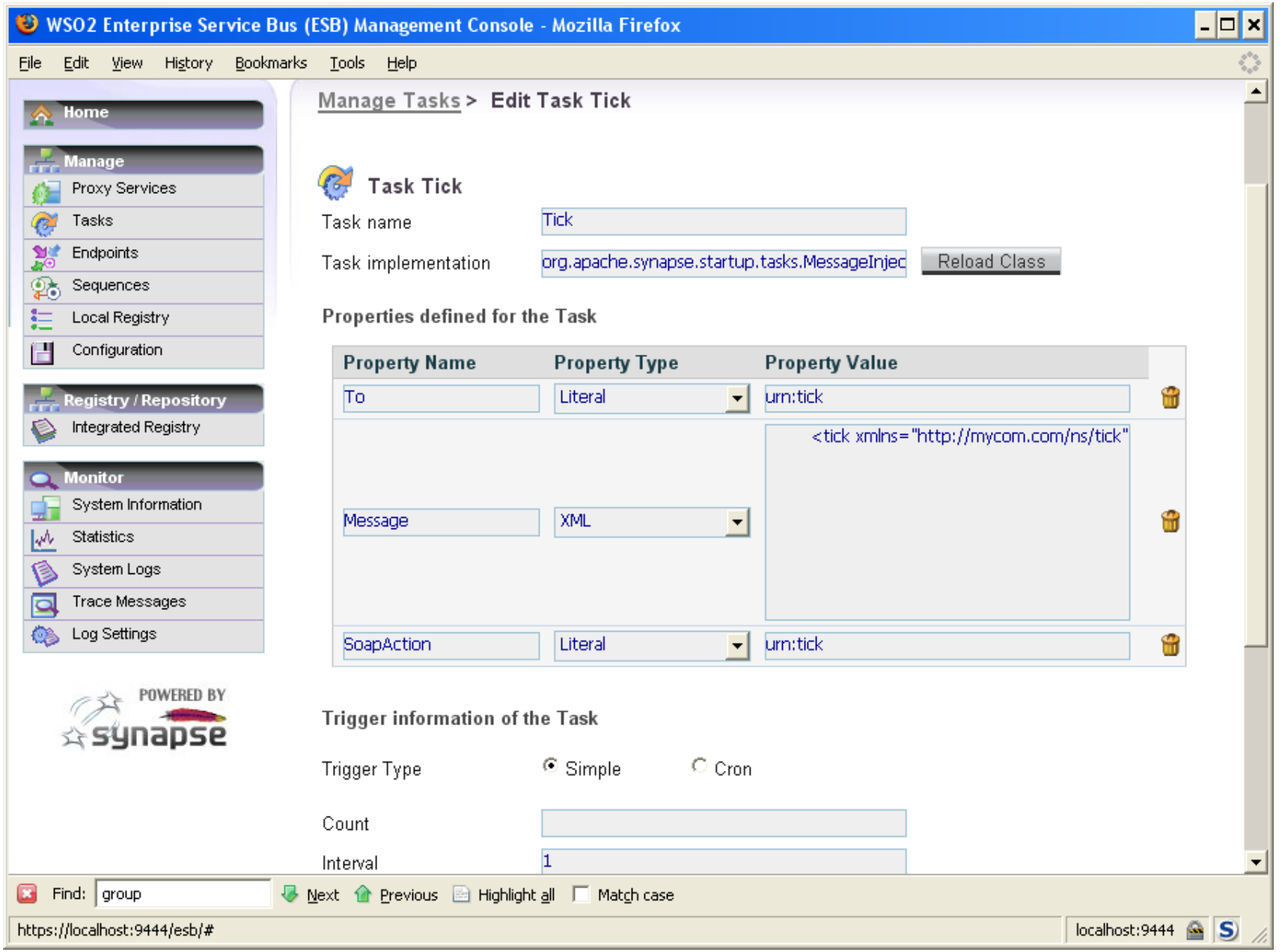
Getting the Axiom mediator working

- Add the following to the LIB path:
  - esper-2.1.0.jar [Main Esper]
  - esperio-2.1.0.jar [Contains AxiomPlugin]
  - antlr-runtime-3.0.1.jar [Used by Esper]
  - cglib-nodep-2.1_3.jar [Used by Esper]
  - EsperAxiomMediator.jar [Mediator code]

Simple example

```
<task
    class="org.apache.synapse.startup.tasks.MessageInjector"
name="Tick">
      <property name="to" value="urn:tick"/>
      <property name="soapAction" value="urn:tick"/>
      <property name="message">
          <tick xmlns="http://mycom.com/ns/tick">
              <s>IBM</s>
              <p>89.6</p>
          </tick>
      </property>
      <trigger interval="1"/>
  </task>
```

In the console



Simple example continued

```
<filter source="get-property('To')" regex="urn:tick">
    <class
name="org.sciflex.plugins.synapse.esper.mediators.AxiomMediator">
        <property name="Configuration">
          <esper-configuration>
             <!-- AS ON PREVIOUS SLIDE -->
          </esper-configuration>
        </property>
        <property name="statement"
             value="select * from Ticker.win:length_batch(10)"/>
        <property name="EventToAddress" value="urn:newevent"/>
    </class>
</filter>
<filter source="get-property('To')" regex="urn:newevent">
   <log level="full"/>
</filter>
```

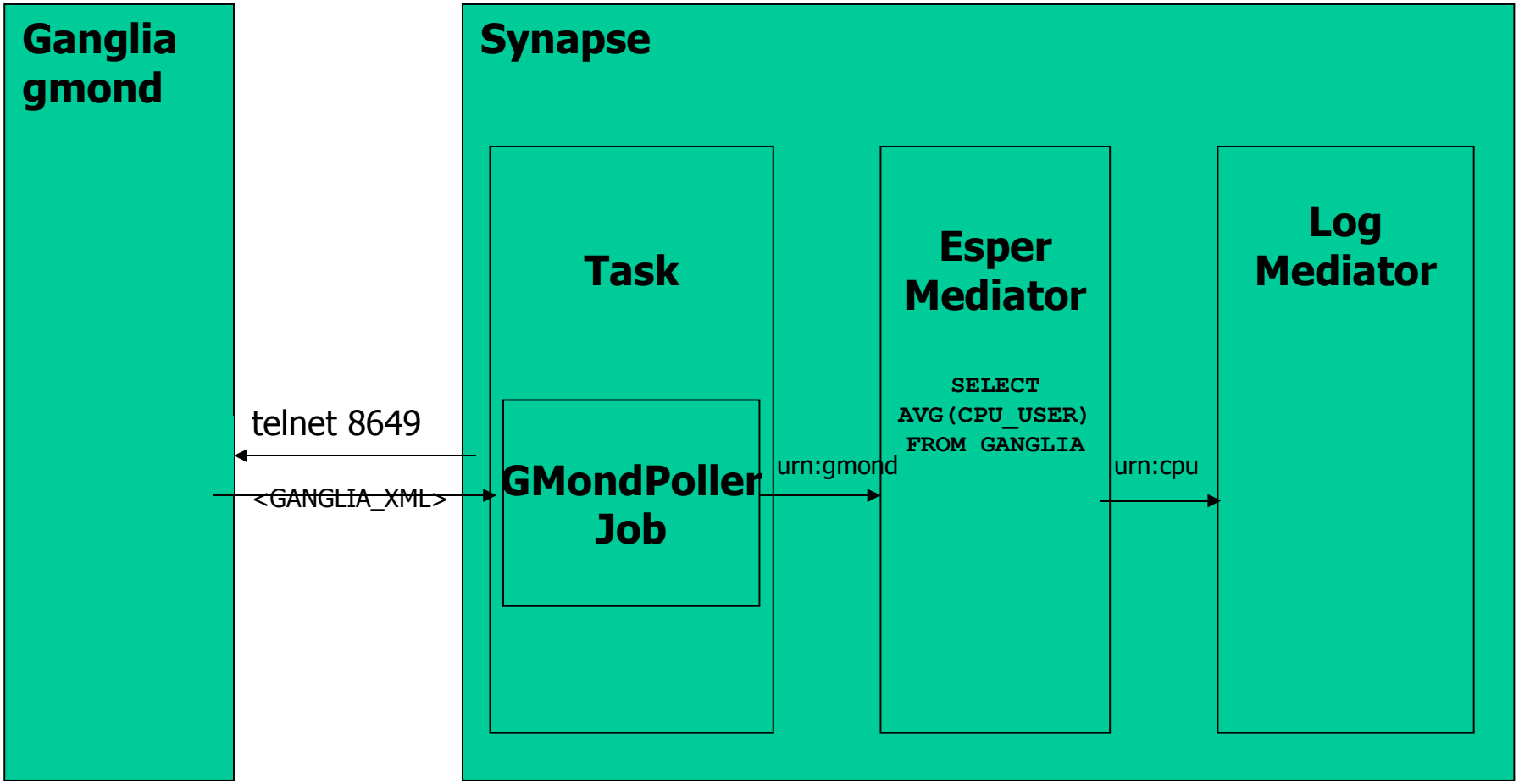
RESULT: Every ten ticks one event gets logged

What happens when Esper generates a new Event?

- Depends on your EPL
  - If your EPL has “SELECT * FROM …”
    - Esper gives the original message in XML
  - If your EPL has “SELECT symbol,… FROM”
    - Esper gives you a Map object
    - Synapse/WSO2 ESB has a standard representation of Map messages
    - PayloadHelper.get/setMap
- Synapse attaches the address specified in the config
- Injects the message into the bus
  - It is now handled by the “main” sequence

Example:
Looking for system monitoring events

http://ganglia.info



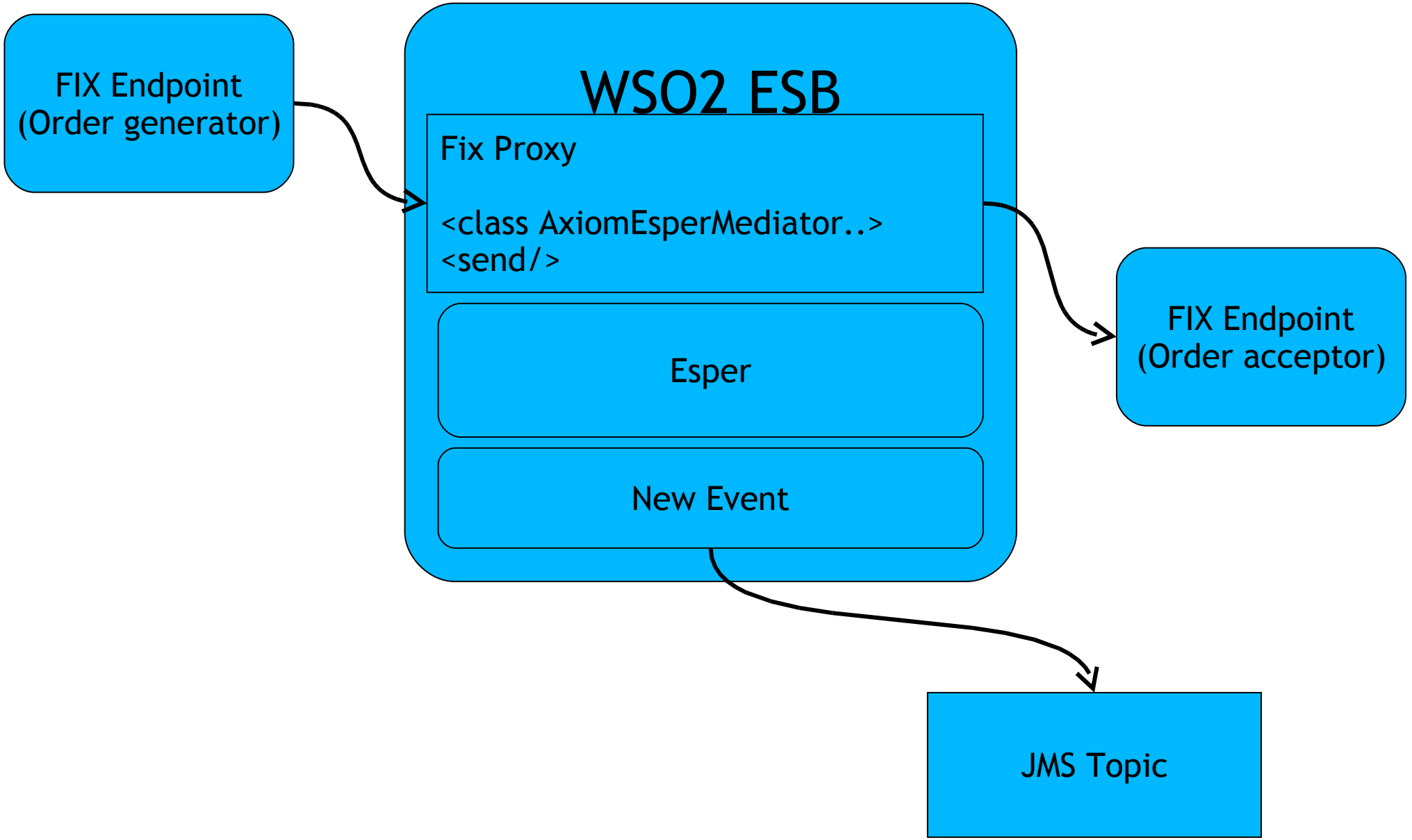
Sharing an Esper instance across two mediators

- Define the configuration in one, together with:
  - <property name="InstanceURI" value="EsperConfig1"/>
    - The name can be anything
- In the second mediator simply use:

```
<class
   name="org.sciflex.plugins.synapse.esper.mediators.AxiomMediator">
    <property name="InstanceURI"
                    value="EsperConfig1"/>
</class>
```

- You may additionally specify new statements in the second mediator

FIX example


FIX Endpoint
(Order generator)
WSO2 ESB
Fix Proxy
<class AxiomEsperMediator..>
<send/>
Esper
New Event
FIX Endpoint
(Order acceptor)
JMS Topic

Futures

- Improving the Mediator to support
  - Adding more than one statement per mediator
  - Having a “startup” configurator for Esper to allow a more logical configuration model

Questions



Resources

- Esper
  - http://esper.codehaus.org
- WSO2 Webinars
  - http://wso2.on.intercall.com
- WSO2 ESB documentation
  - http://wso2.org/project/esb/java/1.7/docs/docs_i

- User List
  - esb-user@lists.wso2.org
- Recent performance testing
  - http://wso2.org/library/3740