



Talend ESB Administration

System Overview and Architecture



Objectives

At the end of this lesson, you will be able to:

- Describe Talend ESB-specific features and architecture
- Locate Talend ESB modules on a real server



Scenario

- You were appointed to administer the new *Talend server* your company recently purchased
- In addition to Data Integration Jobs, you need to understand and deploy *ESB routes* and *data services*
- You know nothing about ESB routes, data services, or Talend server ESB features



Talend Jobs: development cycle



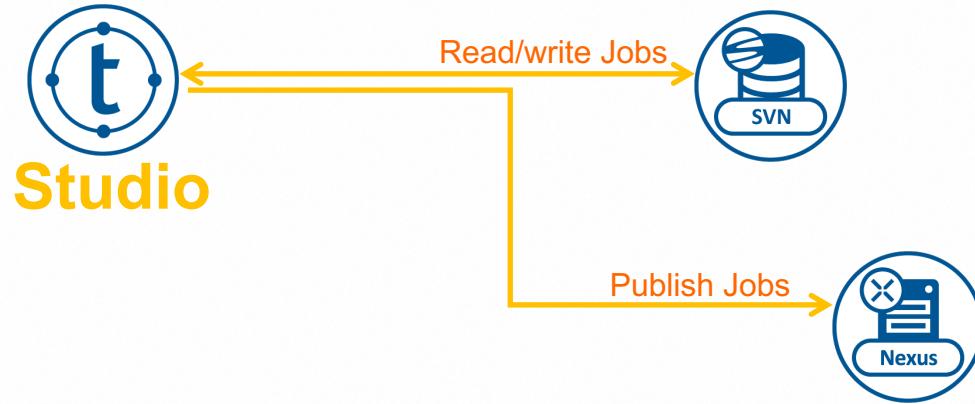
Developers create routes or data services using Talend Studio

Talend Jobs: development cycle, continued



Routes and data services sources are stored in *SVN server*

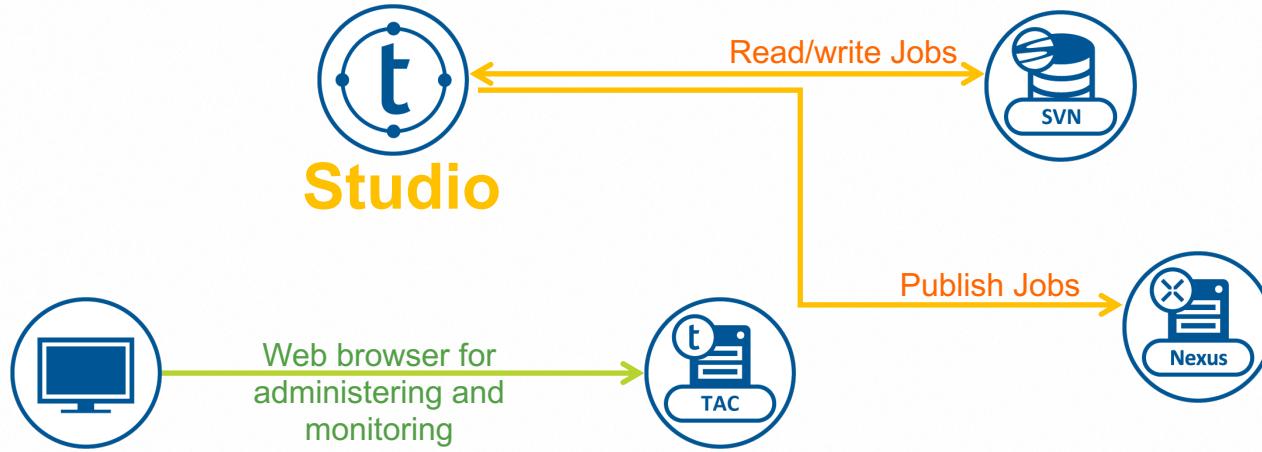
Talend Jobs: development cycle, continued



When ready, routes and data services are *published* on the *Nexus artifact repository*



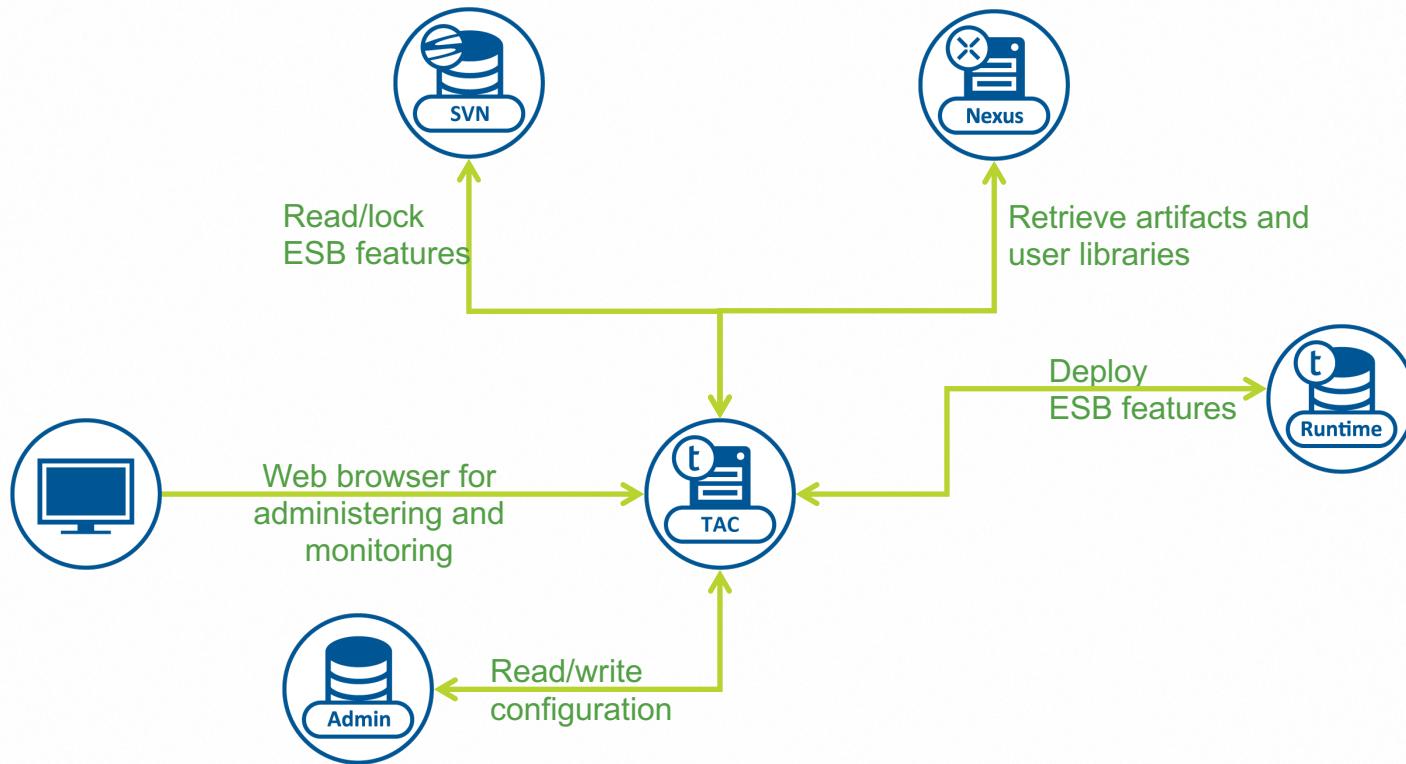
Talend Jobs: deployment



To deploy a route or data service, the *administrator* connects to the *Talend Administration Center (TAC)*



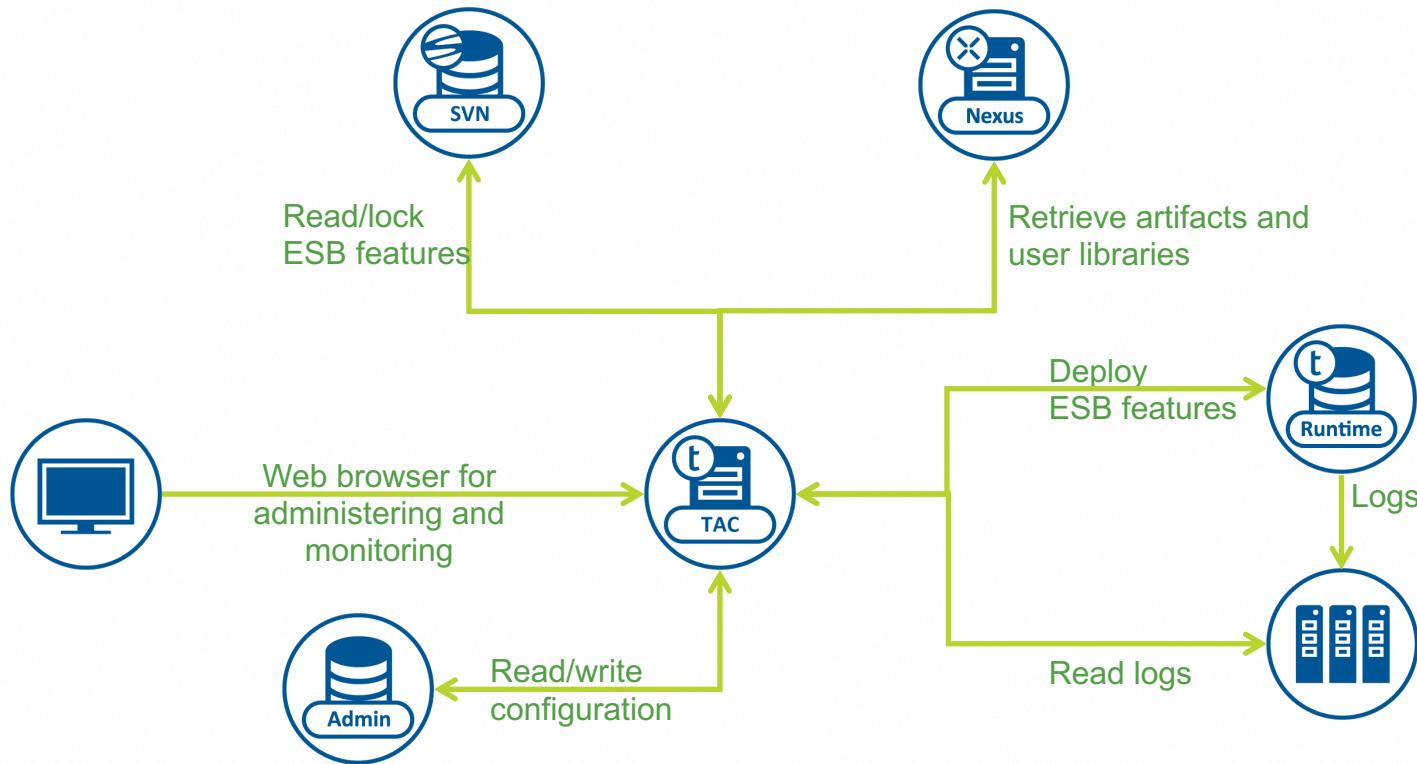
Talend Jobs: deployment, continued



The TAC *deploys routes and data services* on a dedicated execution node called *Talend Runtime*



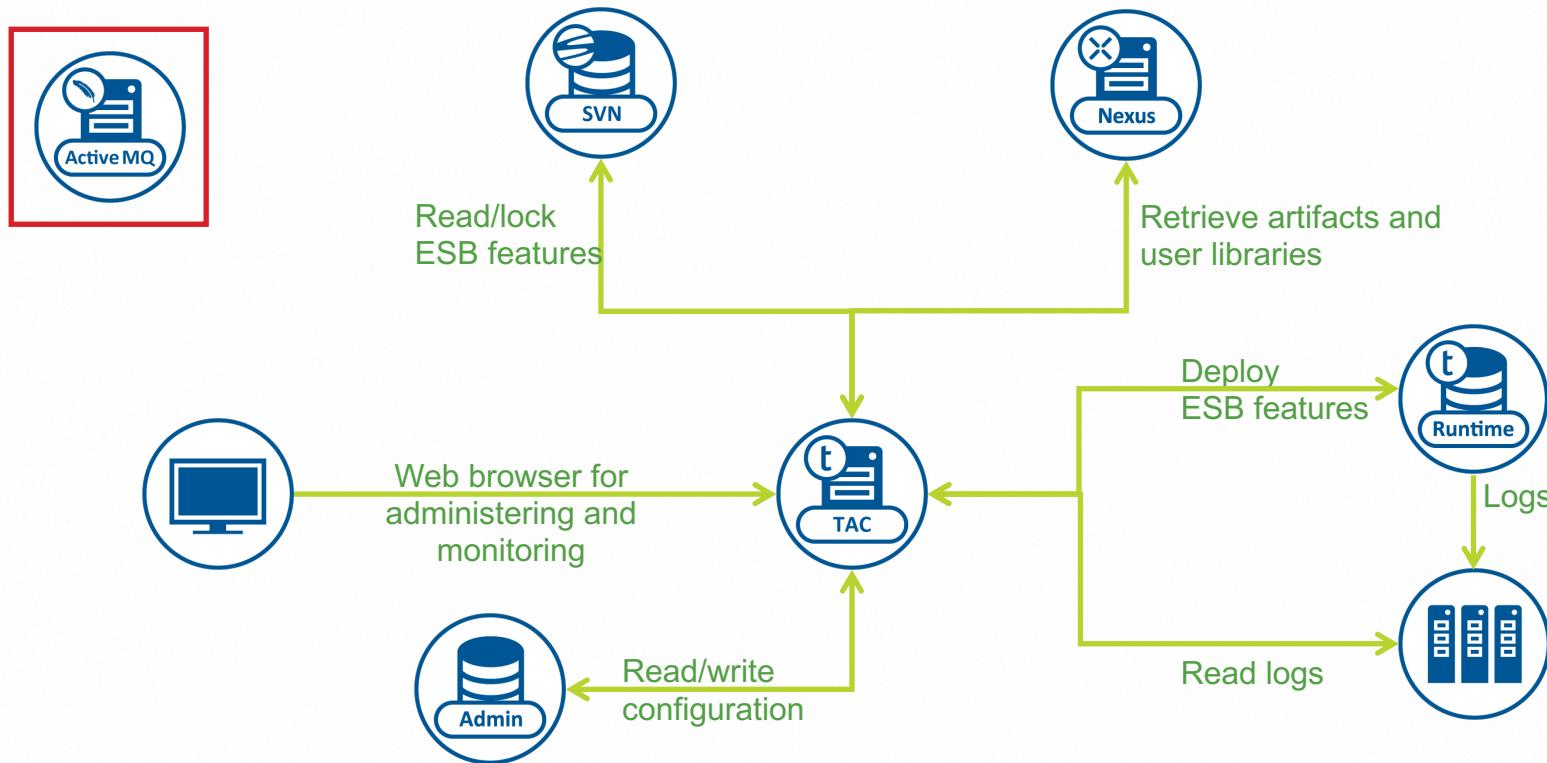
Talend Jobs: deployment, continued



- Talend Runtime can log in *Talend Log Server*
- TAC offers interfaces for analyzing the logs



Talend Jobs: deployment, continued



The *message broker* (ActiveMQ) runs independently in the system for both high-availability and security reasons



Lesson summary

- The Talend ESB server is based on the same unified platform and modules as Talend DI server
- Talend ESB server introduces a new execution environment called Talend Runtime
- Talend ESB server has an optional, highly recommended message broker called ActiveMQ
- The core of the architecture is still the TAC



Talend ESB Administration

Introduction to Talend ESB



Objectives

At the end of this lesson, you will be able to:

- Define ESB and application integration
- Explore ESB features vs. Data Integration features
- Contrast SOAP services with REST services
- Explain ActiveMQ message-delivery strategies



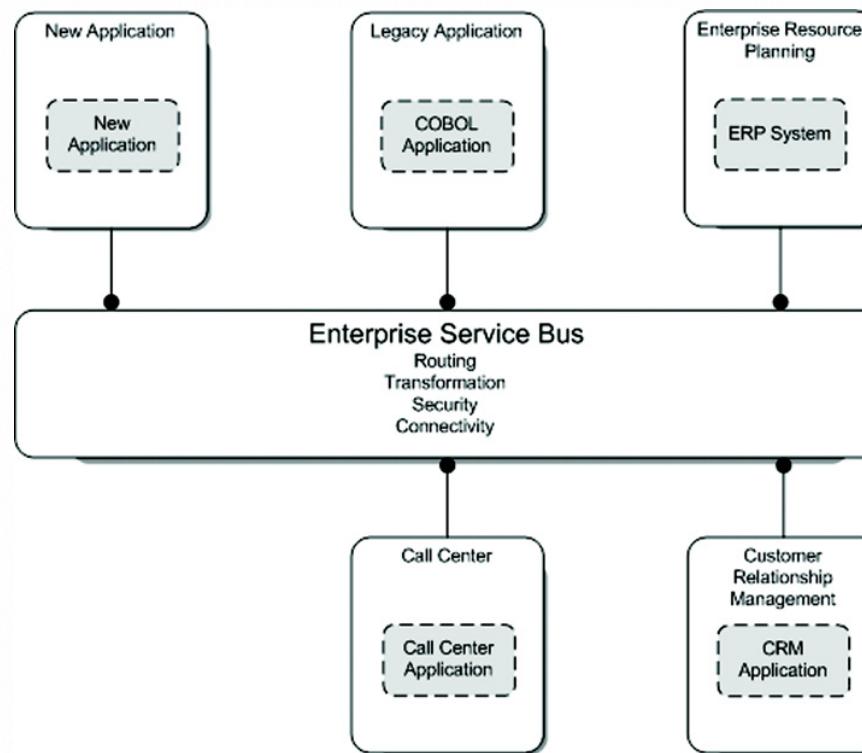
ESB concepts

- ESB stands for *enterprise service bus*
- Refers to the concepts and tools that allow *different heterogeneous applications* in the information system to *exchange messages* via standard formats in real time
- ESB is also referred to as *application integration*



ESB concepts, continued

An ESB is a standards-based integration backbone that combines messaging, web services, message transformation, and intelligent routing





ESB routes and data services

Talend ESB developers use Talend Studio to develop application integration programs called *routes* and *data services*



ESB routes

- Talend routes are programs designed to catch messages on the information system and route (redirect) them to a destination defined by a set of rules and transformations.
- Unlike Talend Jobs, routes are not supposed to end once deployed. Routes are *event-driven programs*, which means they keep scanning the information system for events, and react in real time when they detect one.



ESB routes: example



This is a consumer endpoint.
Consider it an antenna or sensor:
it listens in at a particular place in
your ESB system. It is configured
to catch a specific type of event.



ESB routes: example, continued



This consumer endpoint is configured to listen to an FTP server and catch any new text file from the ./catalogue folder

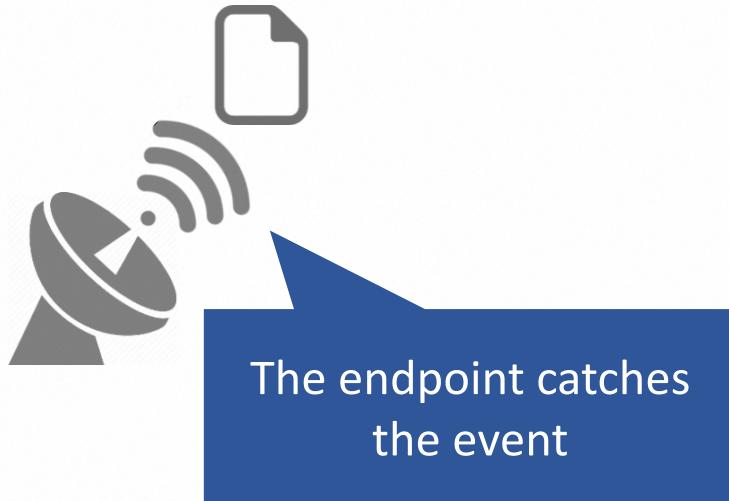


ESB routes: example, continued





ESB routes: example, continued





ESB routes: example, continued



The endpoint sends a message--the content of the file--to a route: a predetermined set of rules and transformations, and, finally, a destination



ESB routes: example, continued



The message is processed
so that the text data is
formatted in XML



ESB routes: example, continued



The message is then sent to its final destination (in this example, ActiveMQ)



ESB data services

- Talend *data services* are either *REST or SOAP web services*
- Web services are small programs running 24x7 that can be remotely queried via HTTP to either run a procedure or access a resource



SOAP web services

- SOAP web services expose operations on a server. These operations can be called remotely by sending an XML-formatted query.
- The SOAP protocol usually uses HTTP as the transport layer.
- All SOAP services declare a WSDL (Web Services Definition Language) file. This file defines all available operations, along with their request and response message formats.

SOAP web services: example, continued



A Talend runtime server exposes a SOAP web service on the Internet. This service can place an order in an e-commerce system.

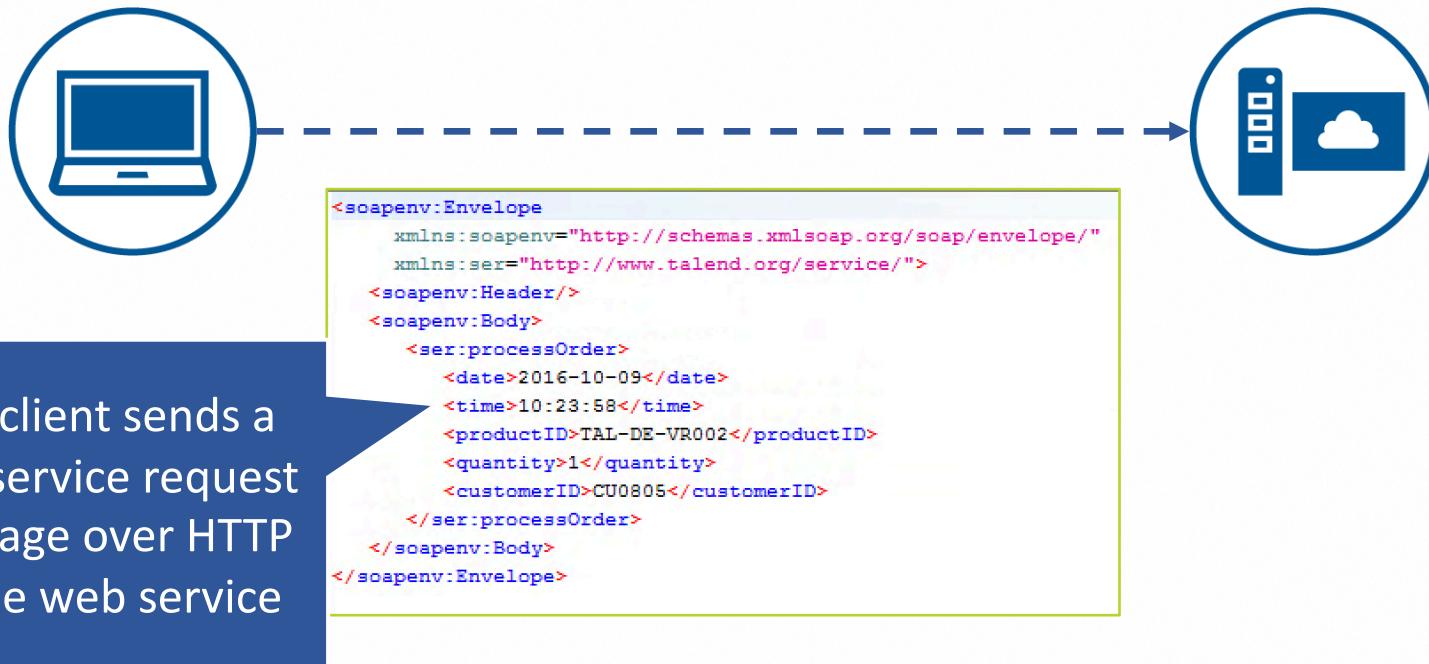
SOAP web services: example, continued



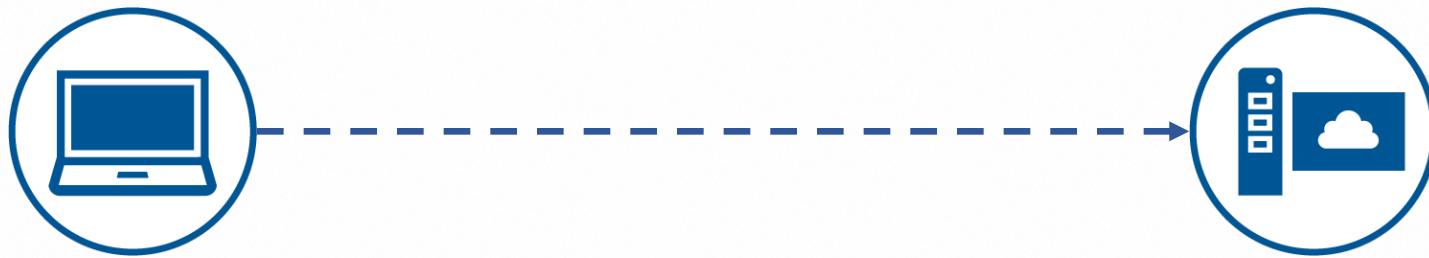
This web service-client
machine needs to place an
order by sending a request to
the SOAP service



SOAP web services: example, continued



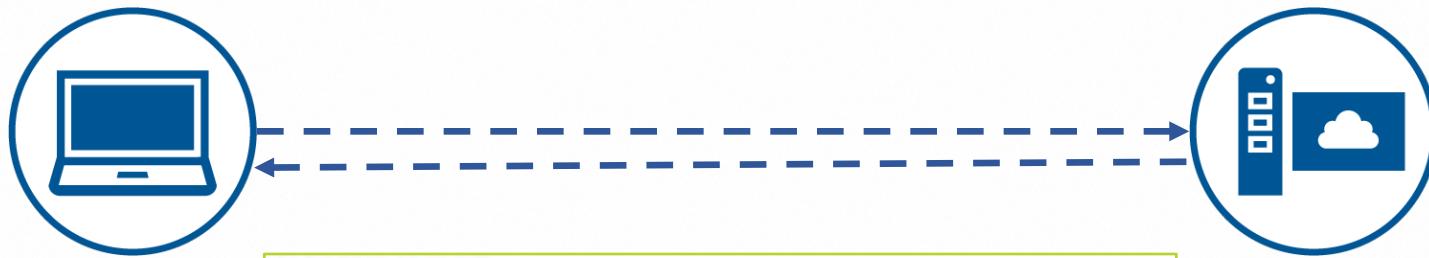
SOAP web services: example, continued



The server processes
the request and
places the order
according to the
information
contained in the
request message



SOAP web services: example, continued



```
<soap:Envelope  
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
    <soap:Body>  
        <tns:processOrder  
            xmlns:tns="http://www.talend.org/service/">  
            <response>Created order #1009</response>  
            <returnCode>1</returnCode>  
        </tns:processOrder>  
    </soap:Body>  
</soap:Envelope>
```

When the operation is complete, the server sends a response message to the web service client



REST web services

- REST web services expose a resource. They make the resource available with four basic operations: GET, POST, PUT, and DELETE.
 - GET allows you to SELECT data from the resource
 - POST allows you to INSERT data in the resource
 - PUT allows you to UPDATE data in the resource
 - DELETE allows you to DELETE data in the resource
- REST services are queried simply by calling a URL



REST web services: example



A Talend runtime server exposes a REST web service on the Intranet. This service provides access to a company resource: the product catalog.



REST web services: example, *continued*



This web service-client machine needs to get product information by querying the REST service.



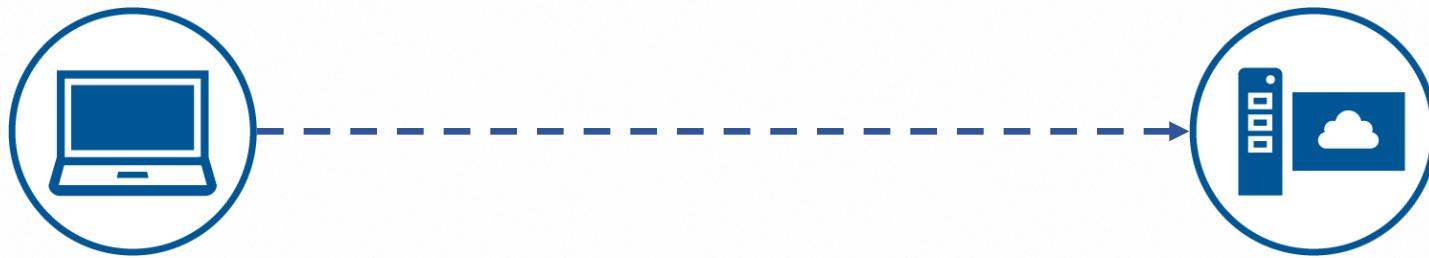
REST web services: example, continued



The client sends a
web service request
over HTTP to the
REST web service



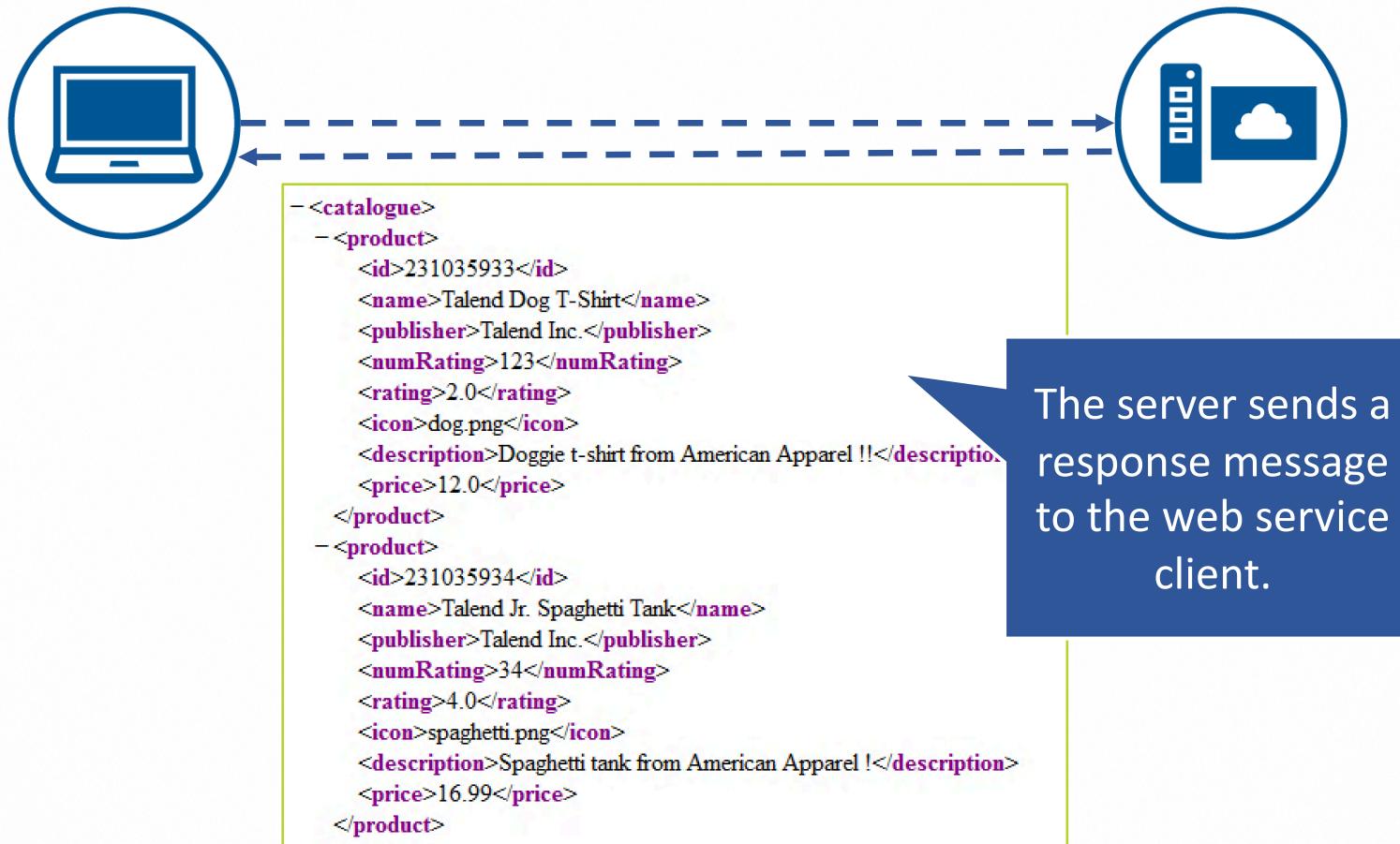
REST web services: example, *continued*



The server receives the request and analyzes the requested action. Then it accesses the resource to perform the request.



REST web services: example, continued





ESB messaging

- Talend ESB server embeds a message broker called ActiveMQ. This module is *message-oriented middleware* (MOM).
- Its only purpose is to handle messages and make sure they are delivered or broadcasted to their recipients.
- ActiveMQ uses standard formats (JMS natively) to make sure messages can be delivered to and understood by any application.



ESB messaging, continued

There are two strategies for delivering a message using MOM:

- Point-to-point messaging, using *queues*
- Broadcasting, using *topics*



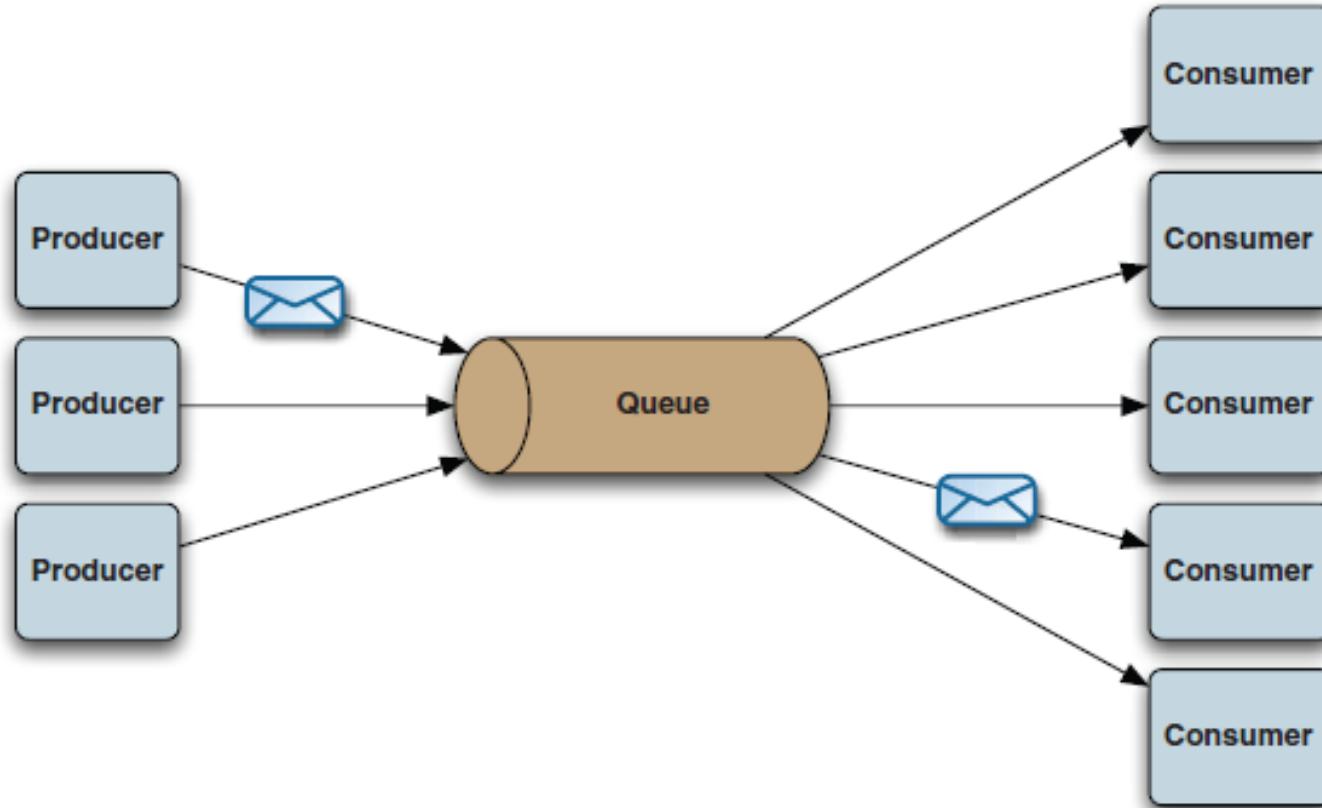
ESB messaging: point-to-point queue

- The sender of the message is called a *producer*.
- The producer sends a message to a queue on the MOM. A queue is a destination with a unique name.
- The recipient of the message is called a *consumer*.
- The consumer is connected to the MOM and listening to one queue.
- When a message is delivered to the queue, the consumer reads it.



ESB messaging: point-to-point queue, continued

- Once a consumer has read the message, *it is considered "consumed" and removed from the queue*
- A message sent on a queue *can have only one recipient*





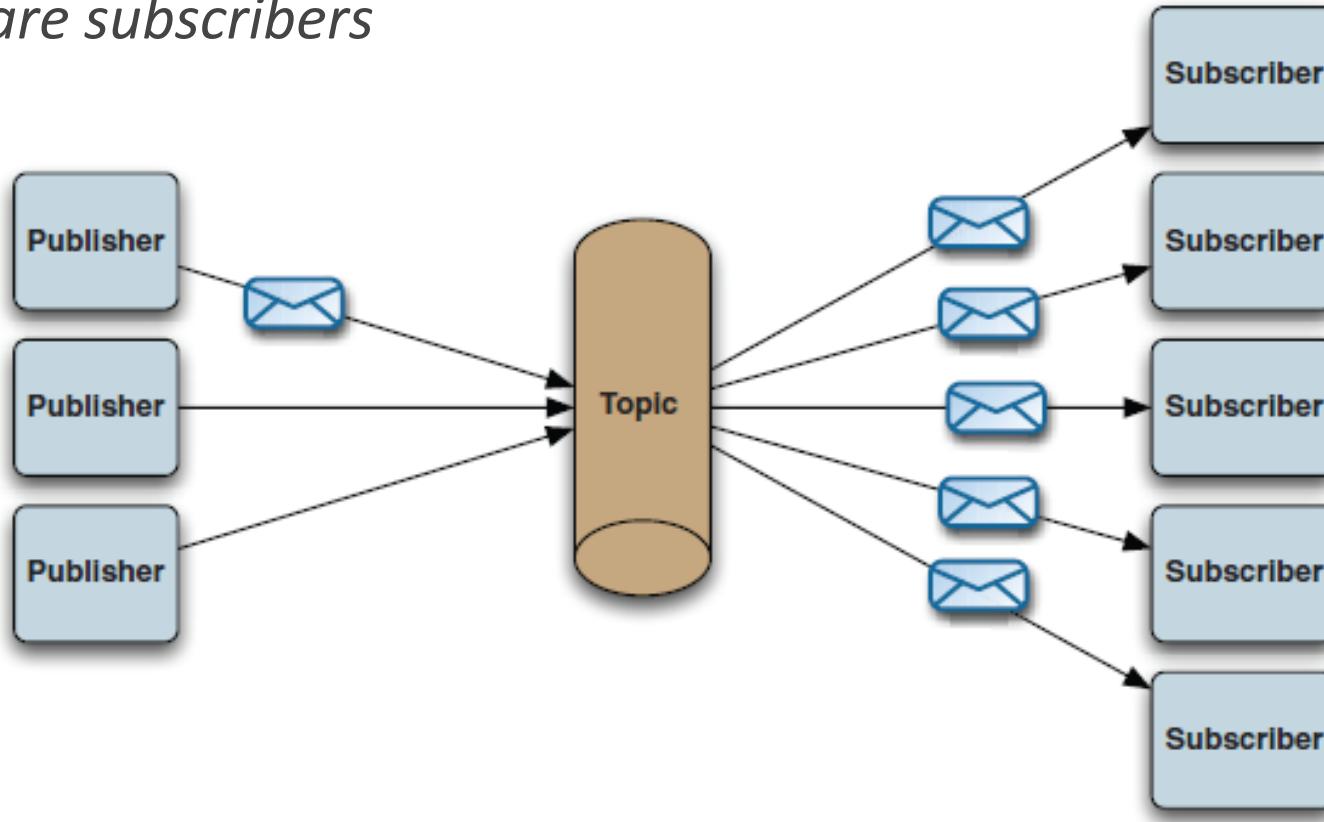
ESB messaging: broadcast topic

- The sender of the message is called a *publisher*.
- The publisher sends a message to a topic on the MOM.
A topic is a destination with a unique name.
- The recipient of the message is called a *subscriber*.
- The subscribers are connected to the MOM and listening to the topic.
- When a message is delivered to the topic, *all connected subscribers receive it*.



ESB messaging: broadcast topic

- When the subscribers receive the message, *it is removed from the topic*
- A message sent on a queue *can have as many recipients as there are subscribers*





Lab overview

ESB infrastructure

In this lab, you will:

- Use a client application to test SOAP and REST web services
- Explore ActiveMQ and its interface



Lesson summary

- The Talend ESB modules integrate seamlessly with the Talend data integration platform (TAC, Log Server, SVN, Nexus)
- Talend ESB artifacts can be SOAP or REST web services as well as routes
- ActiveMQ can deliver standardized messages to applications by using two strategies: queues and topics



Talend ESB Administration

Talend Runtime



Objectives

At the end of this lesson, you will be able to:

- Describe the roles and features of Talend Runtime
- Declare a runtime instance in the Talend Administration Center (TAC)
- Access the runtime console interface
- Run basic commands in the runtime



Scenario

- The new ESB infrastructure at your company requires Talend Runtime
- You are in charge of configuring and operating Talend Runtime
- You must know how to connect a runtime instance to an existing Talend infrastructure



Talend Runtime

- Talend Runtime is based on an open-source project named Karaf
- Its main feature is provision of an execution environment for ESB features: routes and services
- Unlike Talend JobServer, it can reveal route endpoints and data services endpoints on a network



Talend Runtime, continued

Talend Runtime is an OSGi container, which means it follows the OSGi Alliance specifications and must:

- Provide an execution environment
- Be portable and Java based
- Have a native service layer
- Be able to deploy Java jar bundles
- Handle security
- Provide life cycle management
- Handle modules dependency resolution



Talend Runtime, *continued*

In addition to OSGi standard features, Talend Runtime adds professional modules:

- Control via SSH or a web console
- Provisioning and deployment feature via filesystem or Maven
- Monitoring via JMX
- Configuration via filesystem, console, or JMX
- Dependency injection with Spring DM



Declaring a new Talend Runtime instance

- In the TAC, declaring a new Talend Runtime instance is as simple as declaring a JobServer
- Select the Talend Runtime check box and enter the correct ports, then save



Talend Runtime via SSH

- Talend Runtime can be reached via any SSH client
- By default, the runtime listens on port 8101. You need a username and password (these are “karaf” and “karaf” by default) to connect
- You can change the connection parameters in the configuration files, as you will see in the lab



Editing configuration parameters

- All configuration parameters are described in property files
- All property files are stored in the `./etc` folder in the runtime folder
- When editing a configuration file, your changes are applied seconds after you save them



Lab overview

Exploring Talend Runtime

- Add a new Talend Runtime instance to the Talend server infrastructure and configure it
- Connect to the runtime console using a popular simple SSH client (puTTY)
- Explore and edit some configuration files
- Run basic commands in Talend Runtime



Lesson summary

- New runtime can be easily added from the TAC
- Runtime can be accessed and controlled via SSH
- Configuration can be updated by editing the appropriate property files



Talend ESB Administration

Deploying Data Services and Routes



Objectives

At the end of this lesson, you will be able to:

- Describe a typical ESB artifact life cycle
- Manually deploy a data service
- Deploy data services from the Talend Administration Center (TAC)
- Publish and deploy data services and routes from the TAC
- Test data services and routes



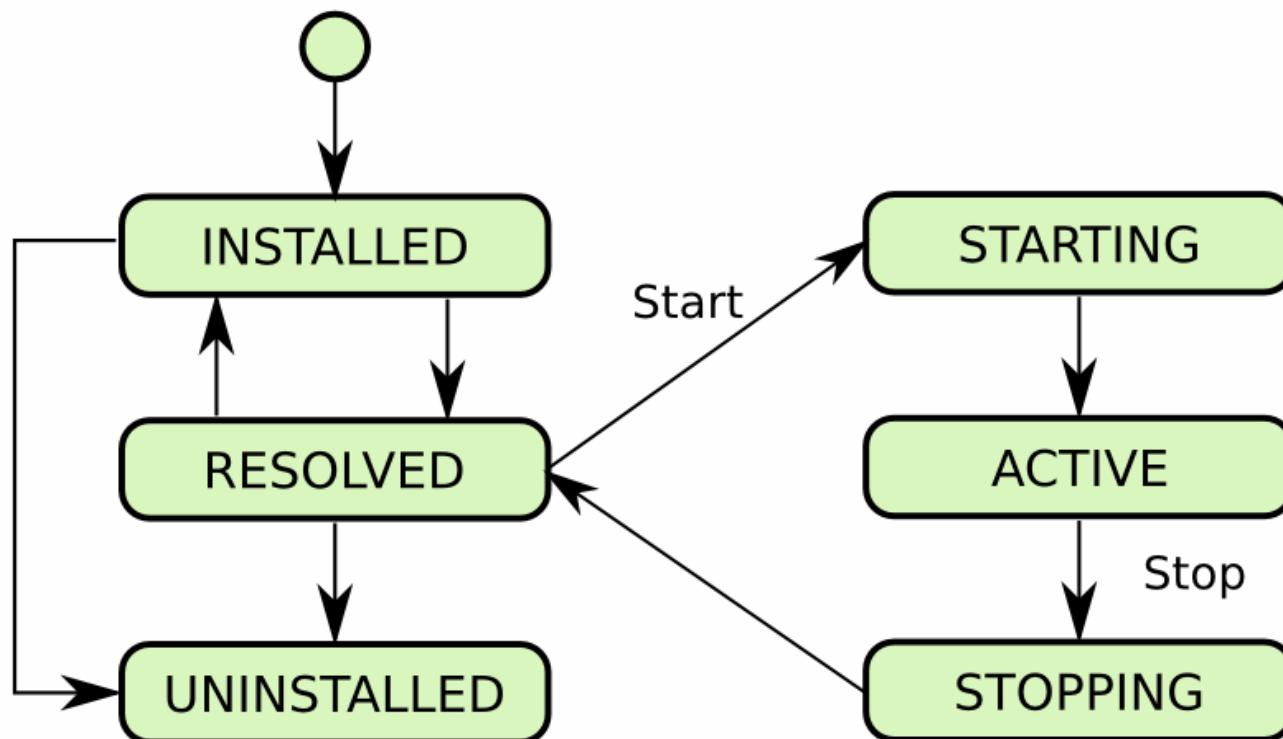
Scenario

- A new ESB project at your company requires data services and routes
- You are in charge of deploying SOAP services, REST services, and routes
- You must know how to check the status of these services and routes, as well as how to test them



ESB artifact life cycle

- The OSGi Alliance defines a standard life cycle for all deployed bundles:





Deploying ESB artifacts

There are three ways to deploy an artifact:

- Deploy a built bundle manually
- Deploy an artifact from Nexus
- Publish an artifact on Nexus and deploy it



Deploying a bundle manually

- It is possible for developers to export a built bundle from Talend Studio
- This bundle can be deployed by moving it to the `./deploy` folder in the runtime directory
- The runtime continually scans this folder and deploys any new bundle deposited in it
- *When to use it:* your runtimes are completely isolated from the Talend server and cannot access the TAC or Nexus



Deploying ESB artifacts

There are three ways to deploy an artifact:

- Deploy a built bundle manually
- Deploy an artifact from Nexus
- Publish an artifact on Nexus and deploy it



Deploying an artifact from Nexus

- When artifacts are already on one of the Nexus repositories (Snapshots or Releases), in the TAC, go to the ESB Conductor page
- Declaring a new service or route is very similar to declaring a new Job from the Job Conductor page
- Once you deploy your artifact, remember that it starts automatically
- *When to use it:* your runtime can be reached from the TAC and artifacts have been published on Nexus



Deploying ESB artifacts

There are three ways to deploy an artifact:

- Deploy a built bundle manually
- Deploy an artifact from Nexus
- Publish an artifact on Nexus and deploy it



Publish and deploy an artifact

- Most companies do not authorize developers to publish their artifacts on Nexus
- When they are not allowed to publish, administrators and operation managers can use the Publisher page from the TAC
- Publisher compiles and builds ESB artifacts from SVN sources, just as it does for Jobs
- *When to use it:* your runtime can be reached from the TAC but artifacts have not been published on Nexus



Lab overview

Deploying services and routes

In this lab, you must deploy developers' data services and routes

- One SOAP service will be deployed manually
- One SOAP service will be deployed from the ESB Conductor page in the TAC
- One REST service will be published from the Publisher, then deployed from the ESB Conductor
- One route will be published from the Publisher, then deployed from the ESB Conductor



Lab overview, continued

Deploying services and routes

- Remember to check the status of your bundles in the runtime console
- Test the route and services



Lesson summary

- All Talend artifacts deployed on the runtime follow the OSGi standard life cycle
- You can deploy artifacts manually by dropping them in the deploy folder of the runtime
- Deploying from the TAC and its ESB conductor interface is a good practice
- The Publisher can publish ESB artifacts on Nexus



Talend ESB Administration

Monitoring Talend ESB



Objectives

At the end of this lesson, you will be able to:

- Activate the Service locator feature
- Control the status of deployed services
- Monitor the activity of SOAP and REST services
- Explore the Kibana ESB dashboard



Scenario

- Routes and services are real-time features. They must be accessible 24x7, so it is important that you can monitor your system and track any change of failure.
- You must also be able to keep history of the requests that were sent to your services, as well as the messages the system returned



Service Locator

- The Service Locator feature tells administrators which services are up and which have failed
- Services must register with the Service Locator in order to be monitored
- When a service is registered, it has to send a very regular signal—called a heartbeat—so the Locator knows it is still alive
- If a service fails and ceases to send its heartbeat, the Locator considers it no longer available



Service Locator, continued

talend | TALEND DATA FABRIC

Refresh Refresh Period (sec) 30 Show live services only all services >

Status	Service Endpoint	Uptime	Transport/P...	Namespace
OrdersWSREST	http://localhost:8040/services/orders	6 min	HTTP JAXRS	http://www.talend.org/rest/
placeOrdersV2	http://localhost:8040/services/placeOr...	Last seen 13 days...	HTTP SOAP1	http://www.talend.org/service/
testSOAPWebService	http://localhost:8040/services/testSOA...	Last seen 13 days...	HTTP SOAP1	http://www.talend.org/service/

Page 1 of 1 Displaying 1 - 3 of 3 Rows per page 10 Last update

Info Metadata

Service Endpoint:	http://localhost:8040/services/placeOrdersV2
Namespace:	http://www.talend.org/service/
Transport:	HTTP
Protocol:	SOAP1
Last time started:	Thu Aug 04 23:57:00 GMT-700 2016
Last time stopped:	Thu Jul 28 01:35:47 GMT-700 2016



Service activity monitoring

- The Service Activity Monitoring feature logs all messages received and sent by a web service in a database.
- Services must register with the Service Activity Monitoring feature. This option must be selected in the Studio before publishing the service.
- TAC has an interface to display the logs.



Service activity monitoring, continued

SERVICE ACTIVITY

talend | TALEND DATA FABRIC

Date / Time ▾	WS portType / REST endpoint	Operation	Transport	Elapsed	Type
Thu Jul 28 02:03:17 GMT-700...	{http://www.talend.org/rest/}OrdersWSREST	GET[/8169]	HTTP	0.02 s	
Thu Jul 28 01:38:17 GMT-700...	{http://www.talend.org/service/}placeOrdersV2PortType	placeOrdersV2Operation	HTTP SOAP	0.67 s	

Page 1 of 1 Displaying 1 - 2 of 2

Details

Flow ID: urn:uuid:d8964d05-95b6-4cd7-b603-87a28c553fe8
WS portType / REST endpoint: {http://www.talend.org/service/}placeOrdersV2PortType
Operation: {http://www.talend.org/service/}placeOrdersV2Operation
Transport: http://schemas.xmlsoap.org/soap/http

Consumer

Provider

Request IN

Date / Time: Thu Jul 28 01:38:17 GMT-700 2016
Custom Info
address: /placeOrdersV2

Response OUT

Date / Time: Thu Jul 28 01:38:17 GMT-700 2016
Message ID: urn:uuid:1f781fab-d700-4f43-a1bc-169d3b0de1cd

76



TAC logging

- The Event Logger module collects all runtime logs by default and stores them in Talend Log Server
- The Kibana logging interface has a specific ESB dashboard for letting administrators explore and process ESB messages



Lab overview

Monitoring Talend ESB

- In the runtime console, activate the Service Locator feature
- Stop a service and observe what appears on the TAC Locator page
- Query your services, and, in the TAC, confirm that your queries and service answers have been recorded by the Service Activity Monitoring feature
- Explore the Logging page and its ESB dashboard



Lesson summary

- You can monitor service status by using the Service Locator feature
- The Service Activity Monitoring feature logs requests and responses for your services
- The Event Logger captures logs from your services and routes and displays them on an ESB- specific dashboard on the Logging page in the TAC

