redhat.

JBoss Data Grid and JBoss Fuse Demo

Thomas Qvarnstrom
JBoss Technology Evangelist
Feb, 2015

WHAT IS THE CAMEL JBOSS DATA GRID COMPONENT?

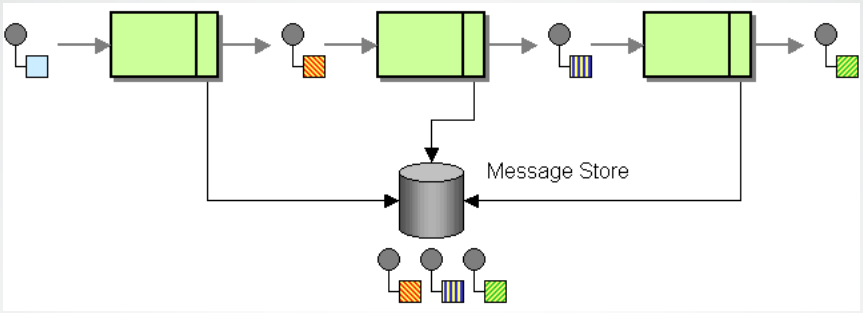
- A component that enables Camel integration flows to directly access JBoss Data Grid
  - Camel is a market leading integration framework that enables integration developers to use common integration patterns.
    - Camel is the core of JBoss Fuse ESB
  - JBoss Data Grid is an in-memory data grid:
    - A data grid is an in-memory distributed database designed for fast access to large volumes of data and scalability.
    - JDG is open source and based on the popular open source project called Infinispan

WHY COMBINE ESB AND DATA GRID?

- Benefits:
  - Shared storage of data
  - Distributed storage of data
  - Store data outside the bus
  - Optimize ESB for throughput
  - Optimize DATA GRID for storage

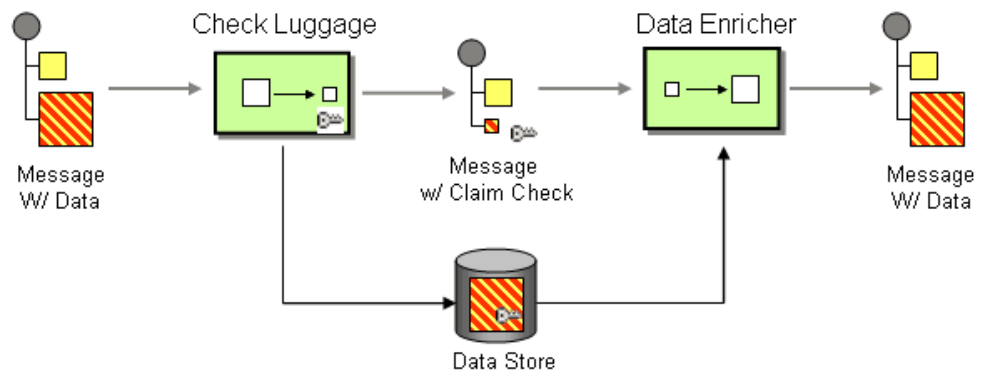
PATTERNS AND USE-CASES

EIP PATTER: MESSAGE STORE



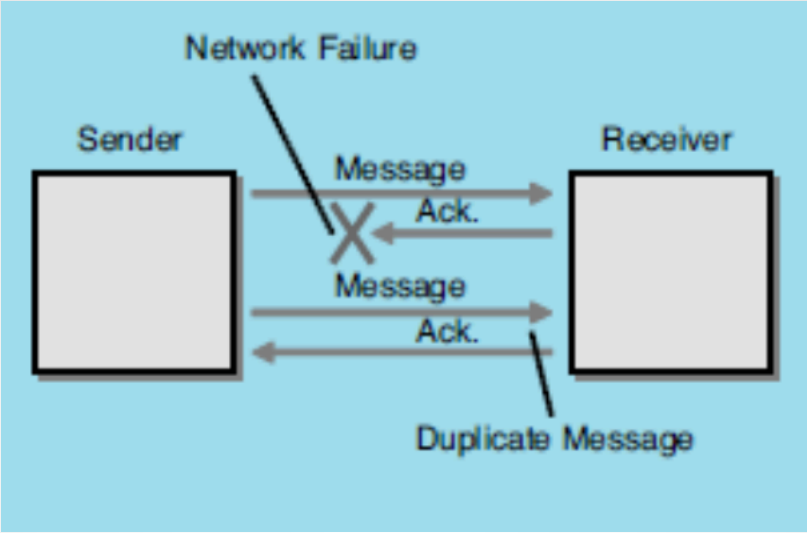
As the Message History describes, the architectural principle of loose coupling allows for flexibility in the solution, but can make it difficult to gain insight into the dynamic behavior of the integration solution.

EIP PATTER: CLAIM CHECK



The Content Enricher tells us how we can deal with situations where our message is missing required data items. The Content Filter lets us remove uninteresting data items from a message. Sometimes, we want to remove fields only temporarily. For example, a message may contain a set of data items that may be needed later in the message flow, but that are not necessary for all intermediate processing steps. We may not want to carry all this information through each processing step because it may cause performance degradation and makes debugging harder because we carry so much extra data.

EIP PATTER: IDEMPOTENT RECEIVER



The term idempotent is used in mathematics to describe a function that produces the same result if it is applied to itself, i.e. f(x) = f(f(x)). In Messaging this concepts translates into the a message that has the same effect whether it is received once or multiple times. This means that a message can safely be resent without causing any problems even if the receiver receives duplicates of the same message.

CAMEL-JBOSSDATAGRID COMPONENT

CAMEL-JBOSSDATAGRID COMPONENT

- Provided in JBoss Data Grid (6.4 or later) for Fuse 6.1 or later
- Can run basic DG operations as camel endpoints
- Supports both in library and in remote/standalone
- Uses header parameters for keys and values
- Provided as OSGi Feature or standalone jar

ENDPOINT

- The endpoint for camel-jbossdatagrid component is infinispan://
  - First part of the URI is used as host and port for remote(hotrod) endpoints.
    - If a **cacheContainer** is specified, the cache will be local. If it is not, the cache will try to connect to remote cache using the supplied hostname/port.
  - The endpoint can be used as both producer or consumer
    - Producer are for PUT, GET, REMOVE or CLEAR commands

```
<to uri="infinispan://localhost?command=GET"/>
```

    - Consumers are listeners for different eventTypes (library mode only)

```
<from uri="infinispan://foo?cacheContainer=#cacheManager"/>
```

HOW TO RUN THE DEMO

PREPARATIONS

- Download the code from jbossdemocentral[1]
- Download JBoss Fuse and JBoss Data Grid from jboss.org[2][3]

[1]=http://public-jbossdemocentral.rhcloud.com/#/datagrid
[2]=http://www.jboss.org/download-manager/file/jboss-fuse-6.1.0.GA-full_zip.zip
[3]=http://www.jboss.org/download-manager/file/jboss-datagrid-6.4.0.GA.zip

INSTALL & SETUP

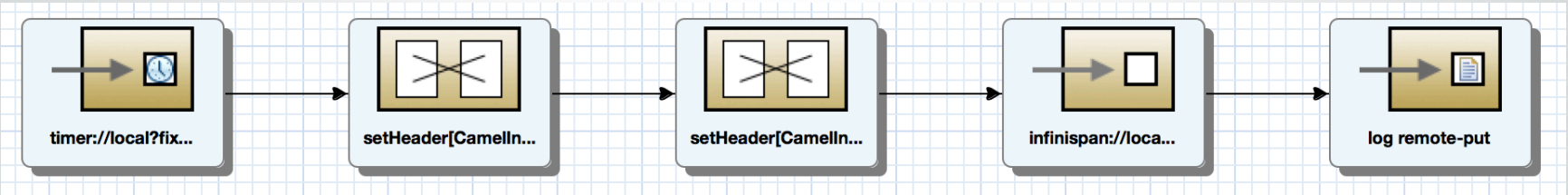
- Installation is fully automated with init.sh script

  ```
  $ sh init.sh
  ```

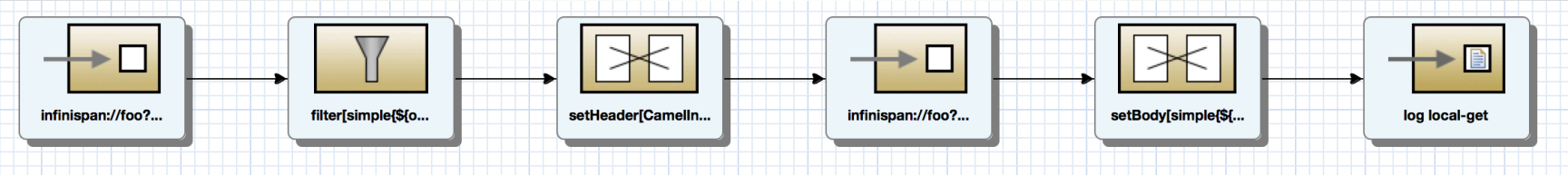
- The init script will
  - Stop any running fuse instances
  - Stop any running datagrid instances
  - Create a target dir
  - Install fuse
  - Install data grid
  - Build the projects
  - Start fuse
  - Start datagrid
  - Start a client configuration script
- After running the script you will have a
  - A running JBoss Data Grid instance
  - A running JBoss Fuse instance
  - All the demo projects are installed as features
  - Profiles with the features installed are created

DEMO 1 - LOCAL CONSUMER AND PRODUCER

PRODUCER EXAMPLE

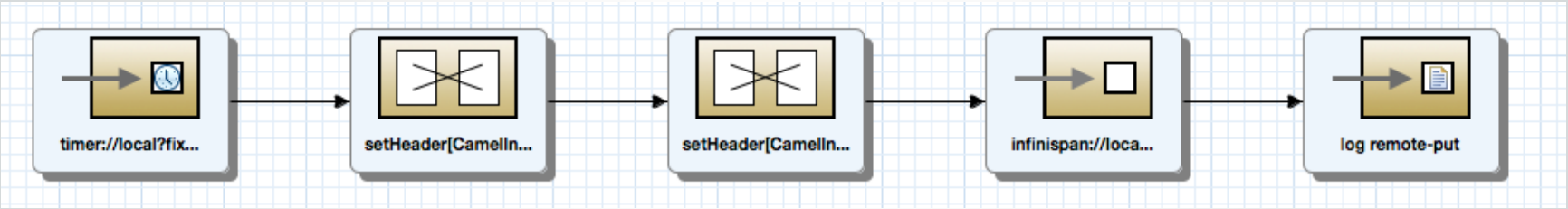


CONSUMER EXAMPLE

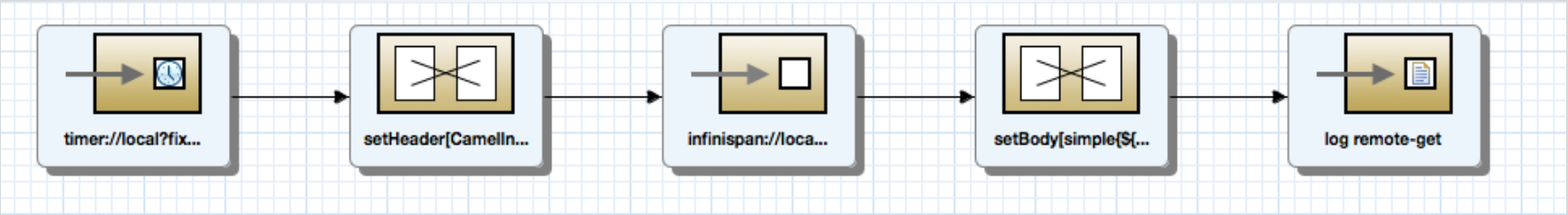


DEMO 2 - REMOTE PRODUCER AND CONSUMER

PRODUCER EXAMPLE

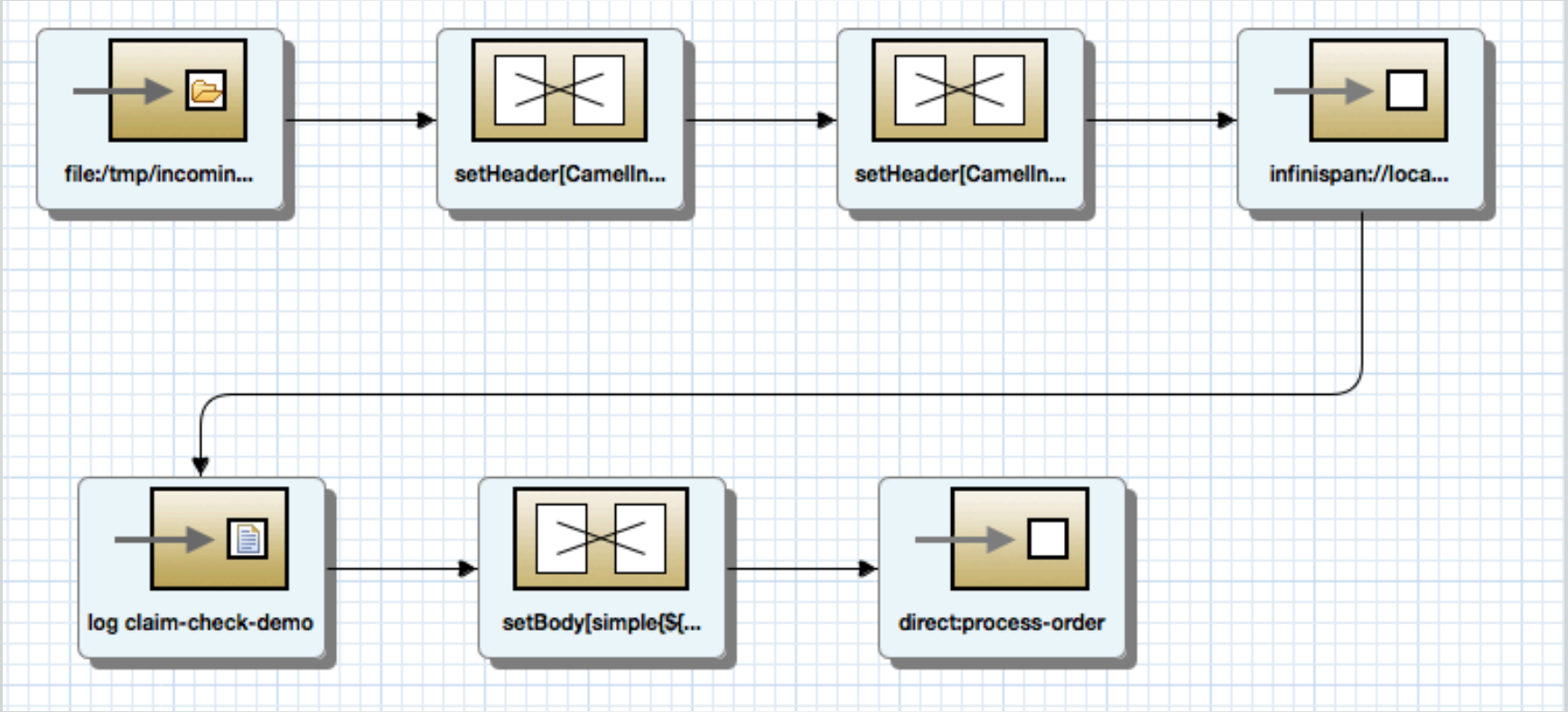


CONSUMER EXAMPLE

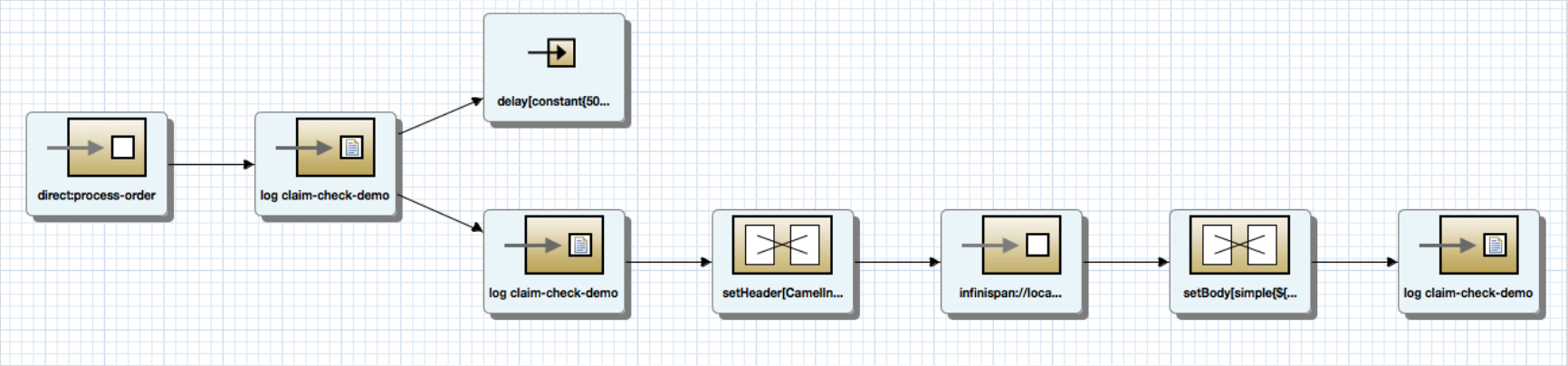


DEMO3 - CLAIM CHECK PATTERN

INCOMING ROUTE



CLAIME ROUTE



Thank you!