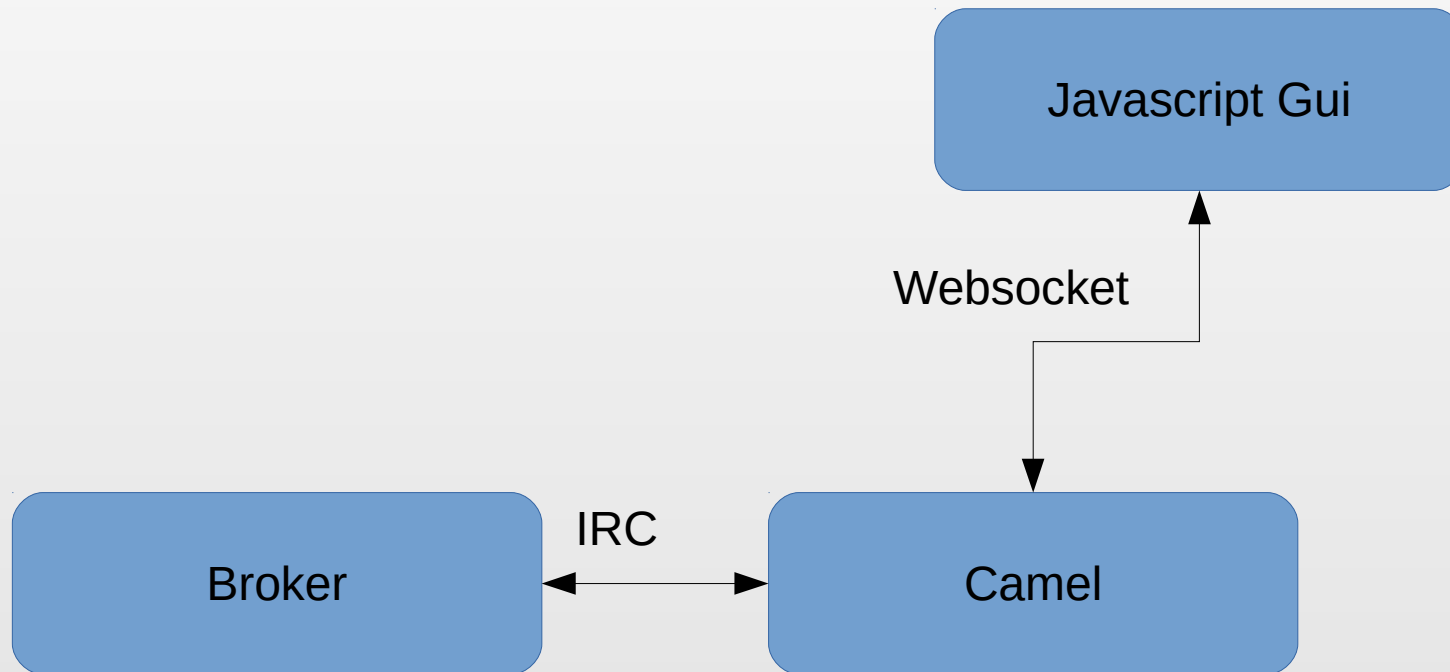


# Komponententechnologien

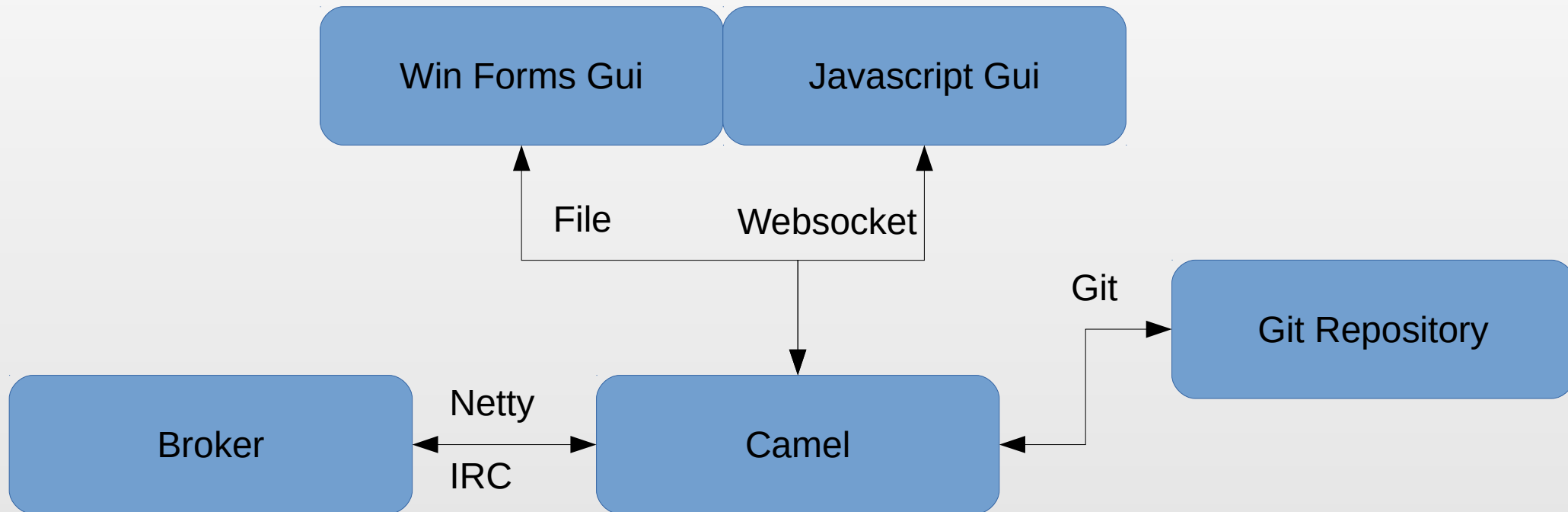
## GUI0 – Part 2

André Heraucourt  
Marek Graca  
Christopher Baumann

# Was bisher geschah...



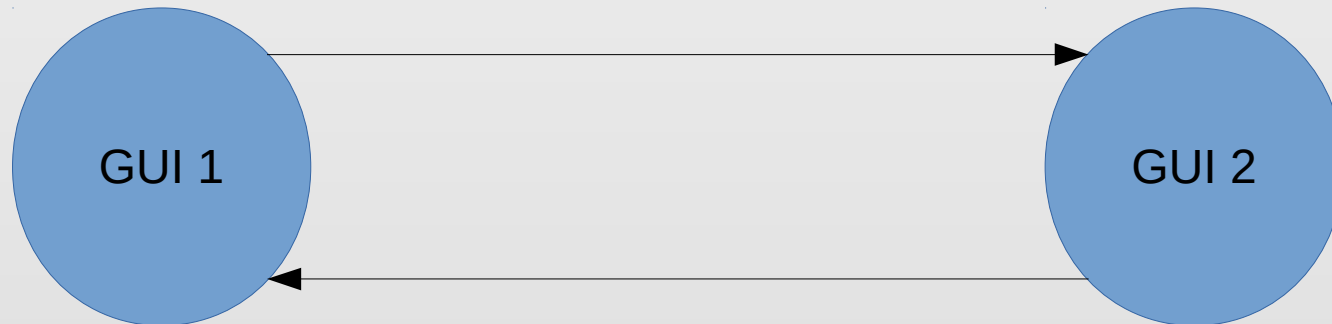
# ... wie es weiter geht





# Kommunikation mit anderer GUI




# Die Antwort:



# Social Sudoku

 This repository Search Pull requests Issues Gist + 


 sudokuGit / sudoku\_communicate Watch 0 Star 0 Fork 0









[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Pulse](#) [Graphs](#) [Settings](#)


No description, website, or topics provided. [Add topics](#) [Edit](#)

53 commits 1 branch 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download


 Goriar Thu Apr 06 01:47:56 CEST 2017 Latest commit 8267e2e 3 days ago

 .camel	Mon Apr 03 00:24:24 CEST 2017	6 days ago
 hiho	Thu Apr 06 01:36:40 CEST 2017	3 days ago
 master	Thu Apr 06 01:36:40 CEST 2017	3 days ago
 20170401.json	Sat Apr 01 19:42:16 CEST 2017	7 days ago
 README.md	Initial commit	8 days ago
 sudoku.json	Mon Apr 03 00:24:24 CEST 2017	6 days ago
 sudoku.json.camellLock	Thu Apr 06 01:36:40 CEST 2017	3 days ago
 test.md	test	8 days ago

 README.md

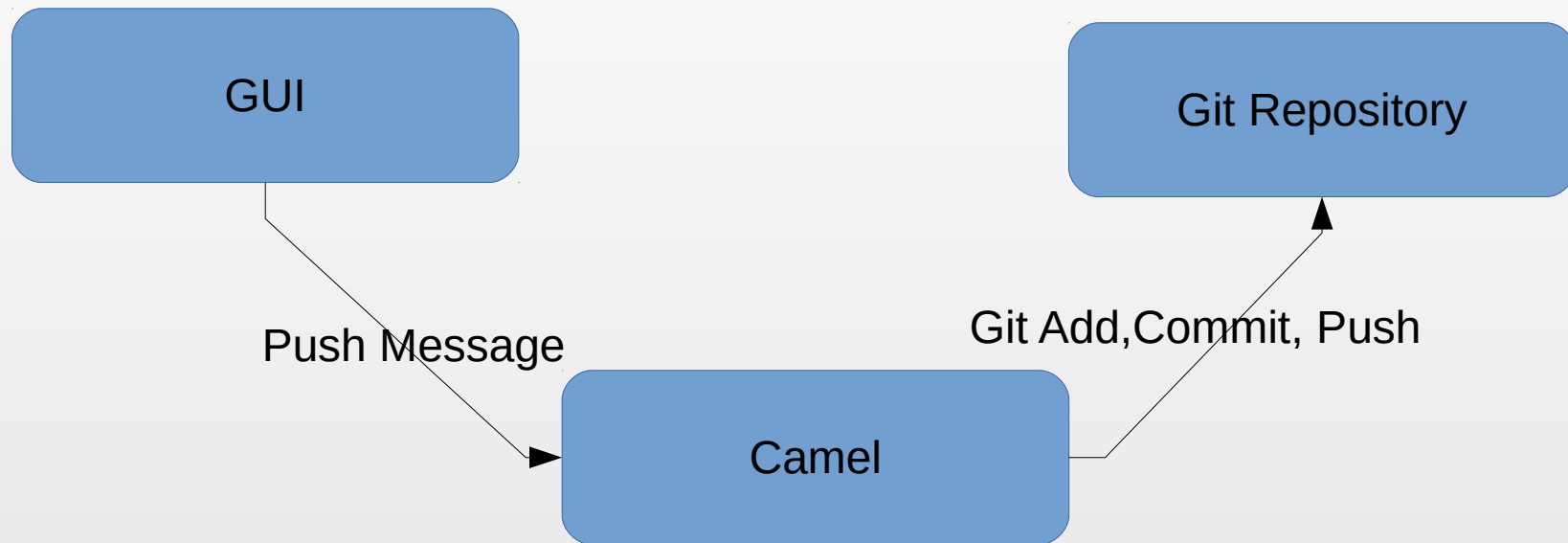
sudoku\_communicate

© 2017 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)

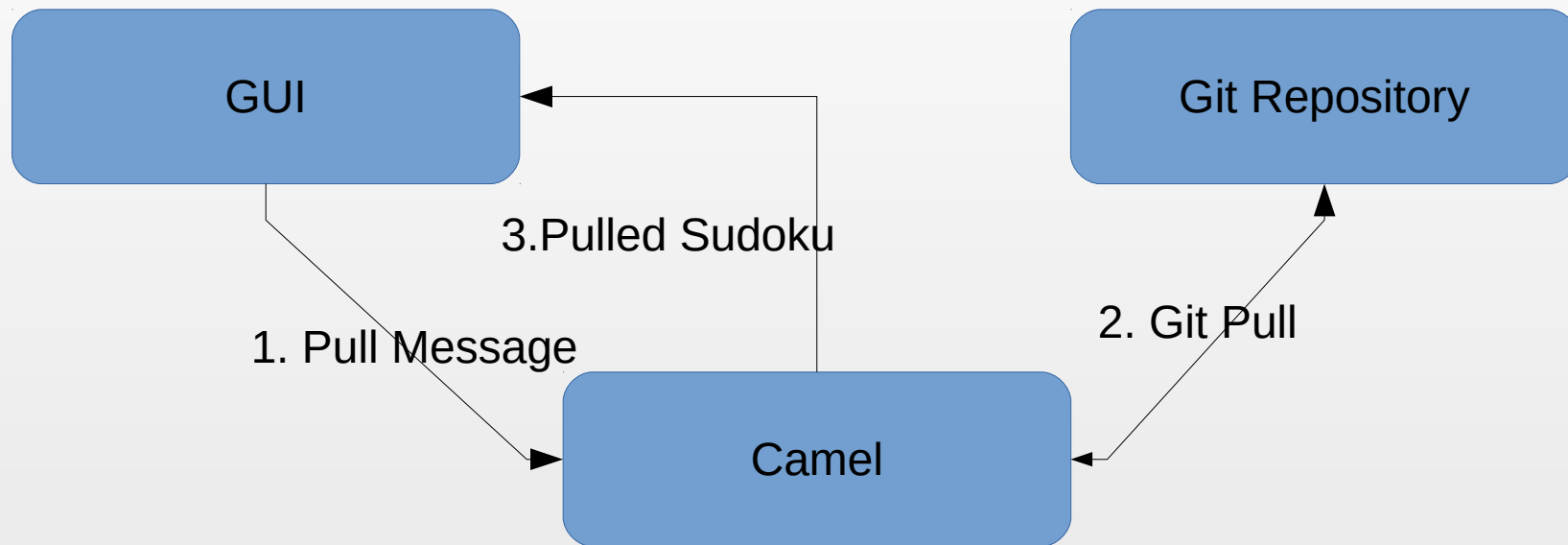


[Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

# Push durch Camel



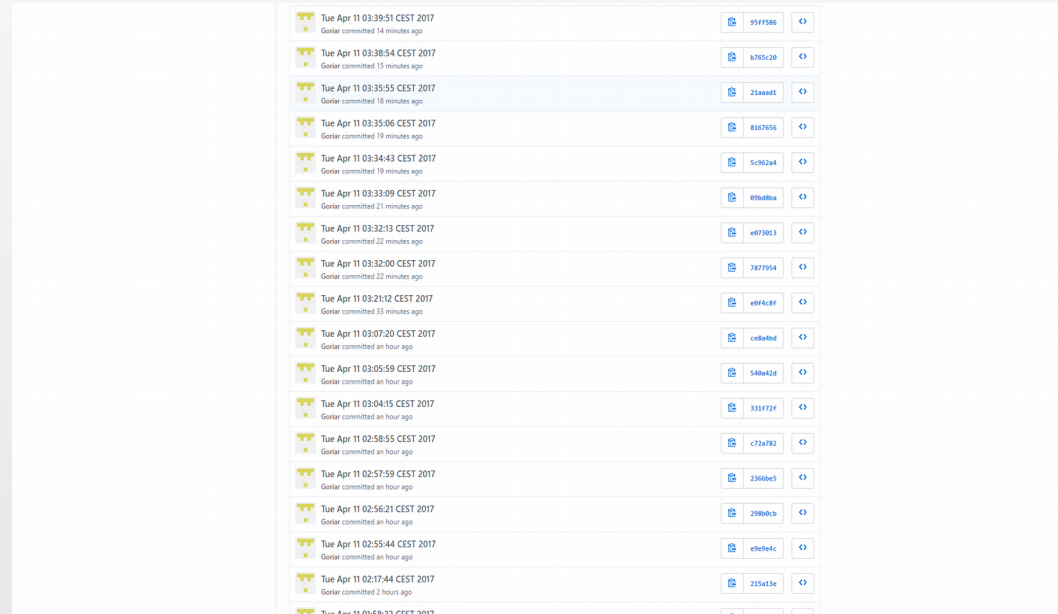
# Pull durch Camel



























































# Camel-Git

- Consumer:  
3 verschieden Typen: Commit, Branch, Tag
- Producer:  
Wichtigste Git-Operationen  
(Push, Pull, Clone, Branch, etc.)

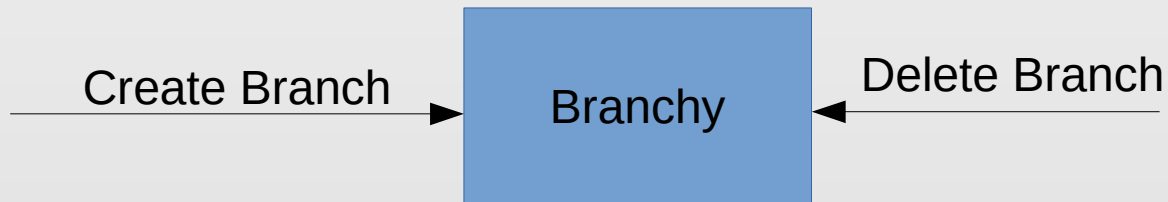


 Tue Apr 11 03:39:51 CEST 2017 Gerlar committed 14 minutes ago	 95f586	
 Tue Apr 11 03:38:54 CEST 2017 Gerlar committed 15 minutes ago	 b765c20	
 Tue Apr 11 03:35:55 CEST 2017 Gerlar committed 18 minutes ago	 21aaad1	
 Tue Apr 11 03:35:06 CEST 2017 Gerlar committed 19 minutes ago	 8167656	
 Tue Apr 11 03:34:43 CEST 2017 Gerlar committed 19 minutes ago	 5c962a4	
 Tue Apr 11 03:33:09 CEST 2017 Gerlar committed 21 minutes ago	 0f9d8ba	
 Tue Apr 11 03:32:13 CEST 2017 Gerlar committed 22 minutes ago	 e0730e3	
 Tue Apr 11 03:32:00 CEST 2017 Gerlar committed 22 minutes ago	 7877954	
 Tue Apr 11 03:21:12 CEST 2017 Gerlar committed 33 minutes ago	 e0f4c8f	
 Tue Apr 11 03:07:20 CEST 2017 Gerlar committed an hour ago	 c0b40d	
 Tue Apr 11 03:05:59 CEST 2017 Gerlar committed an hour ago	 540e42d	
 Tue Apr 11 02:04:15 CEST 2017 Gerlar committed an hour ago	 335f72f	
 Tue Apr 11 02:58:55 CEST 2017 Gerlar committed an hour ago	 c72a782	
 Tue Apr 11 02:57:59 CEST 2017 Gerlar committed an hour ago	 2360be5	
 Tue Apr 11 02:56:21 CEST 2017 Gerlar committed an hour ago	 2980bcb	
 Tue Apr 11 02:55:44 CEST 2017 Gerlar committed an hour ago	 e9e9e4c	
 Tue Apr 11 02:17:44 CEST 2017 Gerlar committed 2 hours ago	 235a13e	
 Tue Apr 11 01:58:22 CEST 2017		

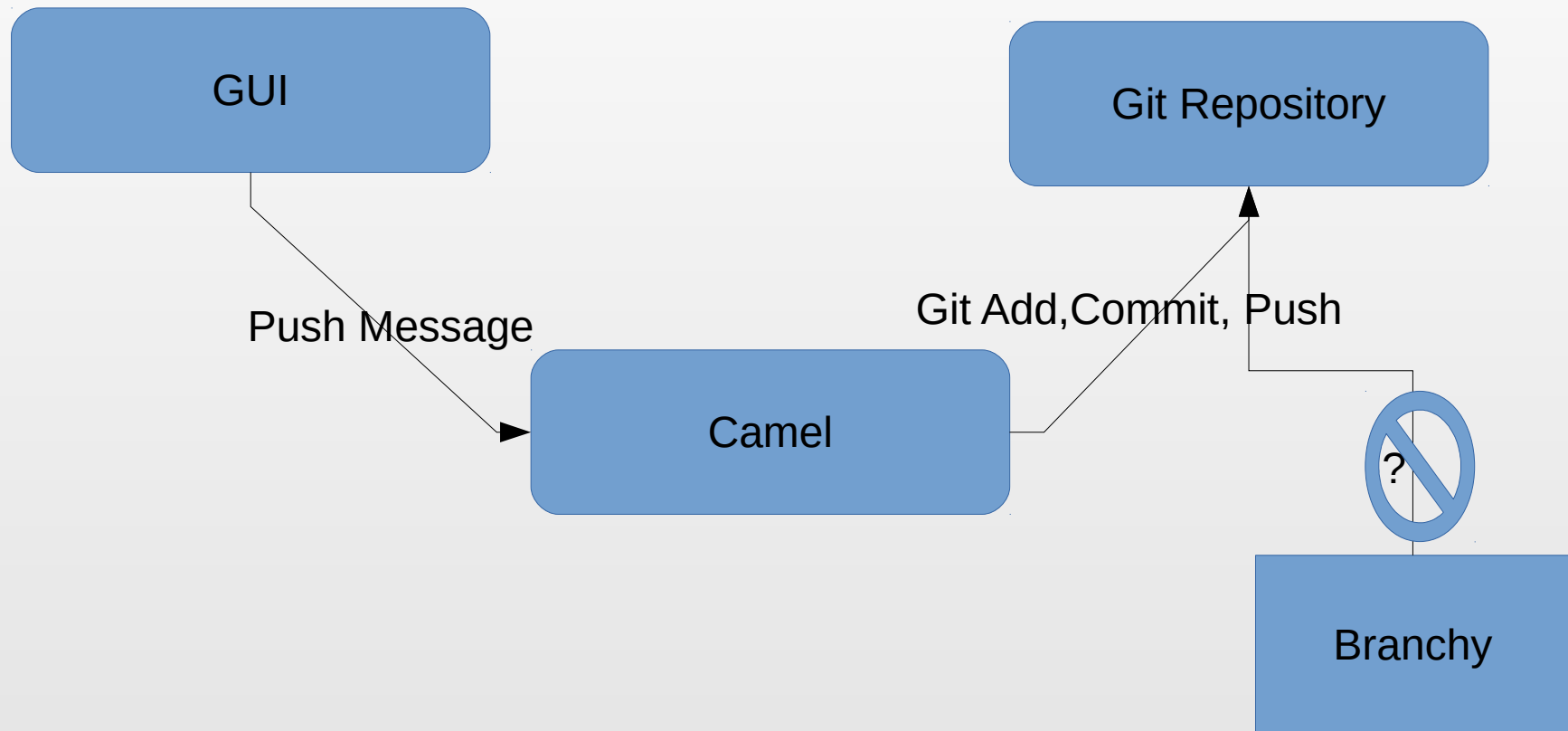
# Camel-Git

Header Optionen:

- Commit bezogene Informationen
- Dateien, für Add-Operation
- Operationen

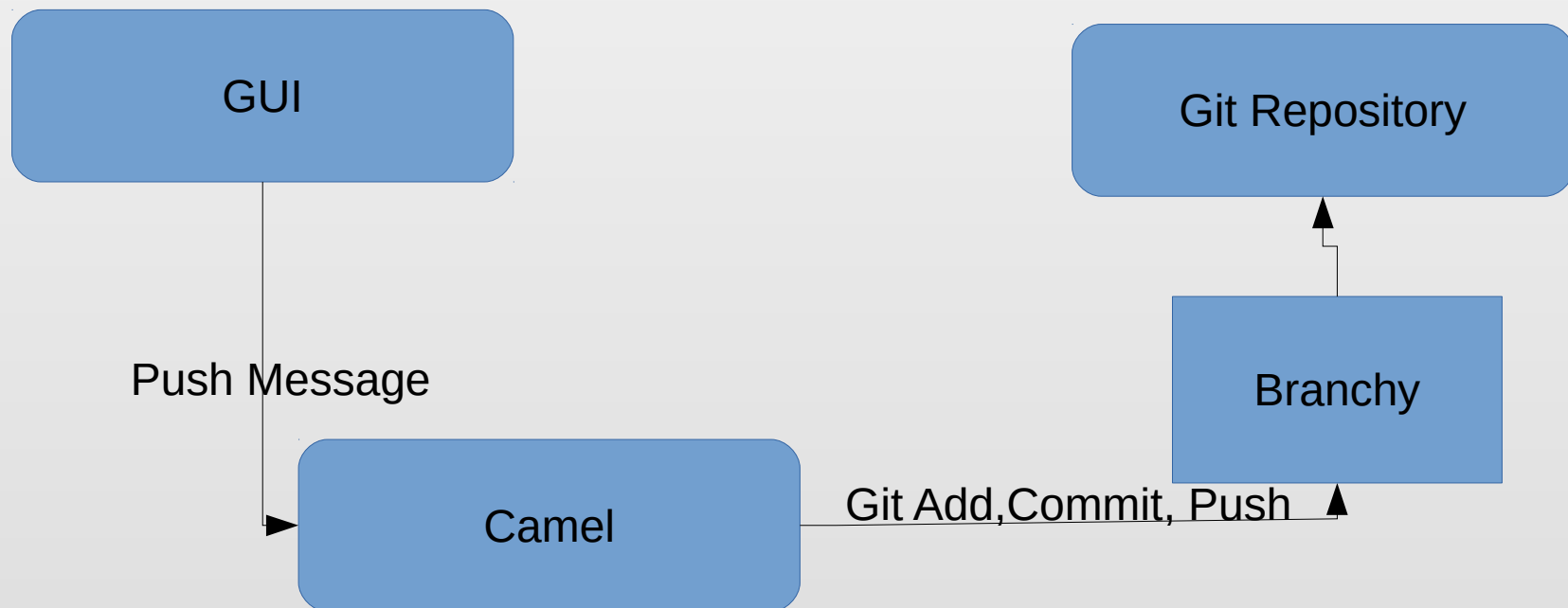


# Push durch Camel

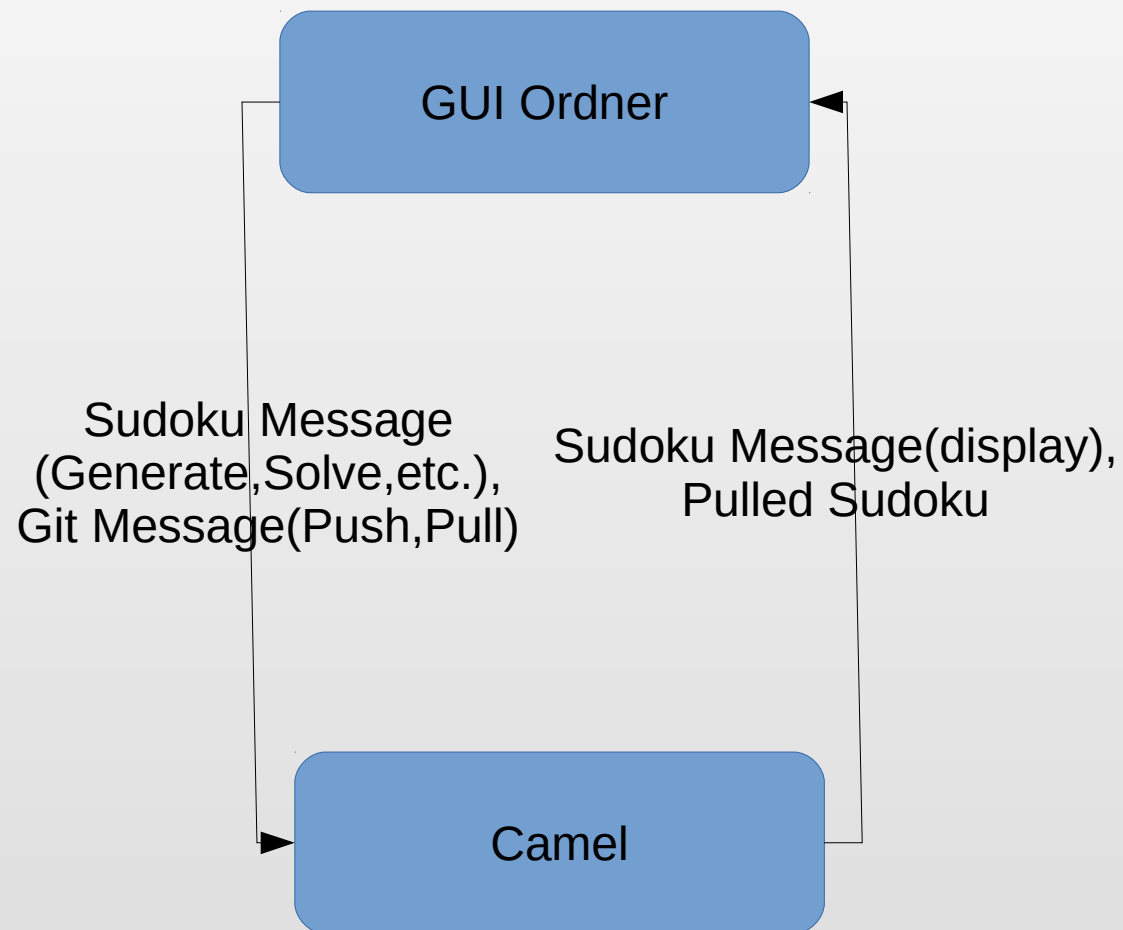


# Problem in der Route

- Kein Checkout, kein Merge
- Branch muss fest in Route eingetragen werden



# File Lösung



# Aufbau einer Git-Route

```
from("direct:start")
    .setHeader(GitConstants.GIT_FILE_NAME,
               constant("test.java"))
    .to("git:///tmp/testRepo?
        operation=add")
    .setHeader(GitConstants.GIT_COMMIT_MESSAGE,
               tant("first commit"))
    .to("git:///tmp/testRepo?
        operation=commit")
    .to("git:///tmp/testRepo?
        operation=push
        &remotePath=https://foo.com/test/test.git
        &username=xxx
        &password=xxx")
```

# Git-Route für WForms-GUI

```
public void configure() {
    from(windowsFormGitOut).convertBodyTo(byte[].class).process(new Processor() {

        @Override
        public void process(Exchange arg0) throws Exception {
            Message m = arg0.getIn();
            byte[] s = (byte[]) m.getBody();

            String in_string = new String(s, Charset.forName("UTF-8"));

            Gson gson = new Gson();
            try {
                GitMessage recv_msg_obj = gson.fromJson(in_string, GitMessage.class);
                branch = recv_msg_obj.getBranch();
                m.setHeader("operation", recv_msg_obj.getOperation());
                if(recv_msg_obj.getOperation().equals("push")){
                    m.setHeader(GitConstants.GIT_COMMIT_MESSAGE, new Date().toString());
                }

            } catch (JsonSyntaxException ex) {
                System.err.println("Message was RUBBISH and will be dropped");
            }
        }

    }).choice().when(header("operation").contains("pull")).to(git_pull).endChoice().otherwise()
        .setHeader(GitConstants.GIT_FILE_NAME, constant(".")).to(git_add)
        .to(git_commit)
        .to(git_push).endChoice().end();
}
```

# Camel

- Integrationshilfe
  - Enterprise Integration Patterns (EIPs)
- Komponenten
  - 13 Core Komponenten
  - 80 weitere Komponenten
  - Integrierte und nicht Integrierte Komponenten



# Twitter

- Integriert
- URI
  - `twitter://endpoint[?options]`
- Probleme:
  - Twitter hat nur 140 Zeichen
  - Für Developer Nutzer wird Handynummer benötigt

# Twitter

- Tweet erzeugen

```
from("direct:foo").to("twitter://timeline/user?  
consumerKey=[s]&consumerSecret=[s]&accessToken=[s]&accessTokenSecret=[s]")  
;
```

- Account pollen

```
from("twitter://timeline/home?  
type=polling&delay=60&consumerKey=[s]&consumerSecret=[s]&accessToken=[s]&a  
ccessTokenSecret=[s]").to("bean:blah");
```

- Keyword Suche

```
from("direct:foo").to("twitter://search?  
keywords=camel&consumerKey=[s]&consumerSecret=[s]&accessToken=[s]&accessTo  
kenSecret=[s]");
```

# Netty4

- Integriert
- Ist eine einfache Socketverbindung
- URI
  - `netty4:tcp:///localhost:99999[?options]`

# Java TCP Server

```
// Socket erstellen
ServerSocket welcomeSocket = new ServerSocket(8888);

while (true) {
    // Auf Verbindung warten
    Socket connectionSocket = welcomeSocket.accept();

    // In- und Out- Channel erzeugen
    BufferedReader inFromClient = new BufferedReader(new
    InputStreamReader(connectionSocket.getInputStream()));
    DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());

    // Empfangen und Versenden
    clientSentence = inFromClient.readLine();
    System.out.println("Received: " + clientSentence);
    capitalizedSentence = clientSentence.toUpperCase() + '\n';
    outToClient.writeBytes(capitalizedSentence);
}
```

# Java TCP Client

// Socket erstellen

```
Socket clientSocket = new Socket("192.168.56.101", 80);
```

// In- und Out- Channel erzeugen

```
BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));
```

```
DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());
```

```
BufferedReader inFromServer = new BufferedReader(new  
InputStreamReader(clientSocket.getInputStream()));
```

// Versenden und Empfangen

```
sentence = inFromUser.readLine();
```

```
outToServer.writeBytes(sentence + '\n');
```

```
modifiedSentence = inFromServer.readLine();
```

```
System.out.println("FROM SERVER: " + modifiedSentence);
```

```
clientSocket.close();
```

# Probleme mit Broker

- Message zum Broker ging
- Rückverbindung fehlgeschlagen
- Test über eigenen Testbroker mit TCP Verbindungen

# Broker Code

// Netty Broker Empfangs route

```
from("netty4:tcp://localhost:5555").process(new  
RouteExtractProcessor()).to("file://camel/echo/");
```

// Netty Broker Versands route

```
from("file://camel/echo/").process(new  
EchoMessageProcessor(msg)).to(toRoute);
```

# Client Code

```
// Netty Client Versands route
```

```
from("file://camel/input").process(new  
MyTestProcessor()).to("netty4:tcp://192.168.56.101  
:5555");
```

```
// Netty Client Empfangs route
```

```
from("netty4:tcp://localhost:8888/").process(new  
NettyTestRespProcessor()).to("stream:out");
```



# netstat

- Protokollstatistik und aktuelle TCP/IP Verbindungen

```
C:\Users\marek>netstat -an | find /I "abhören"
```

TCP	0.0.0.0:135	0.0.0.0:0	ABHÖREN
TCP	0.0.0.0:445	0.0.0.0:0	ABHÖREN
TCP	0.0.0.0:5357	0.0.0.0:0	ABHÖREN
TCP	0.0.0.0:49674	0.0.0.0:0	ABHÖREN
TCP	0.0.0.0:52563	0.0.0.0:0	ABHÖREN
TCP	127.0.0.1:5555	0.0.0.0:0	ABHÖREN
TCP	127.0.0.1:44430	0.0.0.0:0	ABHÖREN
TCP	127.0.0.1:61100	0.0.0.0:0	ABHÖREN
TCP	127.0.0.1:62514	0.0.0.0:0	ABHÖREN
TCP	192.168.0.7:139	0.0.0.0:0	ABHÖREN
TCP	192.168.56.1:139	0.0.0.0:0	ABHÖREN
TCP	:::135	:::0	ABHÖREN
TCP	:::445	:::0	ABHÖREN
TCP	:::5357	:::0	ABHÖREN
TCP	:::49674	:::0	ABHÖREN
TCP	:::52563	:::0	ABHÖREN

# netstat

- netty4:tcp://0.0.0.0:5555/

```
C:\Users\marek>netstat -an | find /I "abhören"
```

TCP	0.0.0.0:135	0.0.0.0:0	ABHÖREN
TCP	0.0.0.0:445	0.0.0.0:0	ABHÖREN
TCP	0.0.0.0:5357	0.0.0.0:0	ABHÖREN
TCP	0.0.0.0:5555	0.0.0.0:0	ABHÖREN
TCP	0.0.0.0:49674	0.0.0.0:0	ABHÖREN
TCP	127.0.0.1:44430	0.0.0.0:0	ABHÖREN
TCP	127.0.0.1:61100	0.0.0.0:0	ABHÖREN
TCP	127.0.0.1:62514	0.0.0.0:0	ABHÖREN
TCP	192.168.0.7:139	0.0.0.0:0	ABHÖREN
TCP	192.168.56.1:139	0.0.0.0:0	ABHÖREN
TCP	:::135	:::0	ABHÖREN
TCP	:::445	:::0	ABHÖREN
TCP	:::5357	:::0	ABHÖREN
TCP	:::5555	:::0	ABHÖREN
TCP	:::49674	:::0	ABHÖREN

# JBoss / WildFly

- Application Server
- JBoss Developer Studio
- JBoss Tools
- Camel Unterstützung

# JBoss Tools Integration Stack

- Camel Developer Tools
  - Visual Route Editor
  - Visual Data Mapper
  - Test and monitor
  - Debugging
  - Deploying

# Visual Route Editor

The screenshot displays the Visual Route Editor interface. The main workspace shows a route diagram for 'Route\_route1' on a grid background. The route consists of three components connected by downward arrows: a green box labeled 'netty4:tcp:0.0.0.0:8888', an orange box labeled 'Process\_process1', and a purple box labeled 'netty4:tcp:localhost:123455'. On the right side, there is a 'Components' palette with a search bar and a 'Select' button. The palette lists various components: Netty HTTP, Netty4, Netty4 HTTP, POP3, POP3S, Process, and Quartz. Below these are folders for 'Routing', 'Control Flow', 'Transformation', and 'Miscellaneous'. At the bottom, there is a 'Properties' panel with tabs for 'Details', 'Advanced', and 'Documentation'. The 'Details' tab is active, showing the 'General' section for the selected component 'netty4:tcp:0.0.0.0:8888'. The 'General' section includes fields for 'Uri \*' (containing 'netty4:tcp:0.0.0.0:8888'), 'Id' (containing '\_from1'), and 'Ref (deprecated)' (a dropdown menu).

Route\_route1

netty4:tcp:0.0.0.0:8888

Process\_process1

netty4:tcp:localhost:123455

Search

Palette

Select

Components

- Netty HTTP
- Netty4
- Netty4 HTTP
- POP3
- POP3S
- Process
- Quartz

Routing

Control Flow

Transformation

Miscellaneous

Design Source Configurations

Properties

Details

Advanced

Documentation

netty4:tcp:0.0.0.0:8888

General

Uri \* netty4:tcp:0.0.0.0:8888

Id \_from1

Ref (deprecated)

# Visual Data Mapper

The screenshot displays the Visual Data Mapper application with two tabs: `*camel-context.xml` and `transformation.xml`. A tooltip at the top reads: "Create a new mapping below by dragging a variable from the list of variables on the left to a property in target XYZOrder on the right."

**Variables Panel:**

(x)= Name	Value
(x)= ORIGIN	"Web"

**Transformations Panel:**

Source	Target
customerNu	custId
orderNum	orderId
status	priority
id	itemId
price	cost
quantity	amount
header	approvalCoc

**Target: XYZOrder Panel:**

- XYZOrder
  - origin

**Summary Diagram:**

```
graph LR; ABCOrder[ABCOrder] --> XYZOrder[XYZOrder]; ABCOrder -- header --> XYZOrder -- approvalCode
```

# Test and monitor

camel-context.xml Diagram View

Search:

```
graph TD
    cbr-route[cbr-route] --> file-work-cbr-input[file:work/cbr/input]
    file-work-cbr-input -- "Total: 5" --> Log_log1[Log _log1]
    Log_log1 -- "Total: 5" --> Choice[Choice]
    Choice -- "Total: 1" --> Otherwise[Otherwise]
    Choice -- "Total: 2" --> When_US[When /order:order/order:customer/order:country = 'US']
    Choice -- "Total: 2" --> Log_log5[Log _log5]
    Otherwise -- "Total: 1" --> Log_log4[Log _log4]
    Log_log4 -- "Total: 1" --> file-work-cbr-output-others[file:work/cbr/output/others]
    When_US -- "Total: 2" --> Log_log3[Log _log3]
    Log_log3 -- "Total: 2" --> file-work-cbr-output-us[file:work/cbr/output/us]
    Log_log5 -- "Total: 2" --> Log_log2[Log _log2]
    Log_log2 -- "Total: 2" --> file-work-cbr-output-uk[file:work/cbr/output/uk]
```

Properties

**Message Details**

Name	Value
CamelFileAbsolutePath	false
CamelFileAbsolutePath	/home/lhein/ws_fusetooling_dev/myCBR/work/cbr/input/order1.xml
CamelFileLastModified	1473865151000

<?xml version="1.0"?>  
<!--  
JBoss, Home of Professional Open Source  
Copyright 2014 Red Hat, Inc. and/or its affiliates and individual

JMX Navigator Terminal

Type in a filter

- Local Processes
  - Eclipse [19538][Disconnected]
  - Eclipse Runtime Workbench [19409][Disconnected]
- Local Camel Context [20218][Connected]
  - MBeans
  - Camel
    - cbr-example-context
      - Endpoints
      - Routes
        - cbr-route
          - file:work/cbr/input
- Server Connections
- User-Defined Connections

Messages View Servers Console

Search:

| Message Box           | Trace ID | Exchange ID   | CamelFileAb | CamelFileAb   | CamelFileLa | CamelFile |
|-----------------------|----------|---------------|-------------|---------------|-------------|-----------|
| <?xml version="1.0"?> | 1        | ID-westeros-4 | false       | /home/lhein/w | 14738651510 | 1462      |
| <?xml version="1.0"?> | 2        | ID-westeros-4 | false       | /home/lhein/w | 14738651510 | 1462      |
| <?xml version="1.0"?> | 3        | ID-westeros-4 | false       | /home/lhein/w | 14738651510 | 1462      |
| <?xml version="1.0"?> | 4        | ID-westeros-4 | false       | /home/lhein/w | 14738651510 | 1462      |
| <?xml version="1.0"?> | 5        | ID-westeros-4 | false       | /home/lhein/w | 14738651510 | 1462      |

# Debugging

Debug - myCBR/src/main/resources/META-INF/spring/camel-context.xml -

File Edit Navigate Search Project Run Window Help

Run Run camel-context.xml as Local on: Local

Quick Access Resource Fuse Integration Debug

Debug Servers

- Run camel-context.xml as Local CamelContext [Local Camel Context]
  - org.codehaus.plexus.classworlds.launcher.Launcher at localhost:36279
    - Thread [main] (Running)
    - Thread [org.apache.camel.spring.Main.main()] (Running)
    - Daemon Thread [Camel (cbr-example-context) thread #2 - file://work/cbr/input] (Running)
  - Camel Context at service:jmx:rmi:///jndi/rmi://localhost:1099/jmxrmi/camel
    - ID-westeros-42729-1473865191244-0-1
      - \_choice1 in cbr-route [camel-context.xml]**

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.102-1.b14.fc24.x86\_64/bin/java (14.09.2016, 16:59:40)

Variables Breakpoints

| Name                  | Value   |
|-----------------------|---|
| MessageBody           | <?xml version="1.0"?>\n<!--\n JBoss, Home of Professional Open So     |
| MessageHeaders        |   |
| CamelFileAbsolutePath | CamelFileAbsolutePath = false\nCamelFileAbsolutePath = /home/lein/ws. |
| CamelFileLastModified | false   |
| CamelFileLength       | /home/lein/ws_fusetooling_dev/myCBR/work/cbr/input/order1.xml         |
| CamelFileName         | 1473865151000   |
|                       | 1462  |
|                       | order1.xml  |

camel-context.xml

```
graph TD; Log1[Log_log1] --> Choice[Choice]; Choice --> Log2[Log_log2]; Choice --> Log3[Log_log3]; Choice --> Log4[Log_log4]; Log2 --> OutUK[file:work/cbr/output/uk]; Log3 --> OutUS[file:work/cbr/output/us]; Log4 --> OutOthers[file:work/cbr/output/others]; OutUK --> Log5[Log_log5]; OutUS --> Log5; OutOthers --> Log5;
```

Design Source Configurations

Console Tasks

Run camel-context.xml as Local CamelContext [Local Camel Context] /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.102-1.b14.fc24.x86\_64/bin/java (14.09.2016, 16:59:40)

org.apache.camel.spring.Main.main() SpringCamelContext

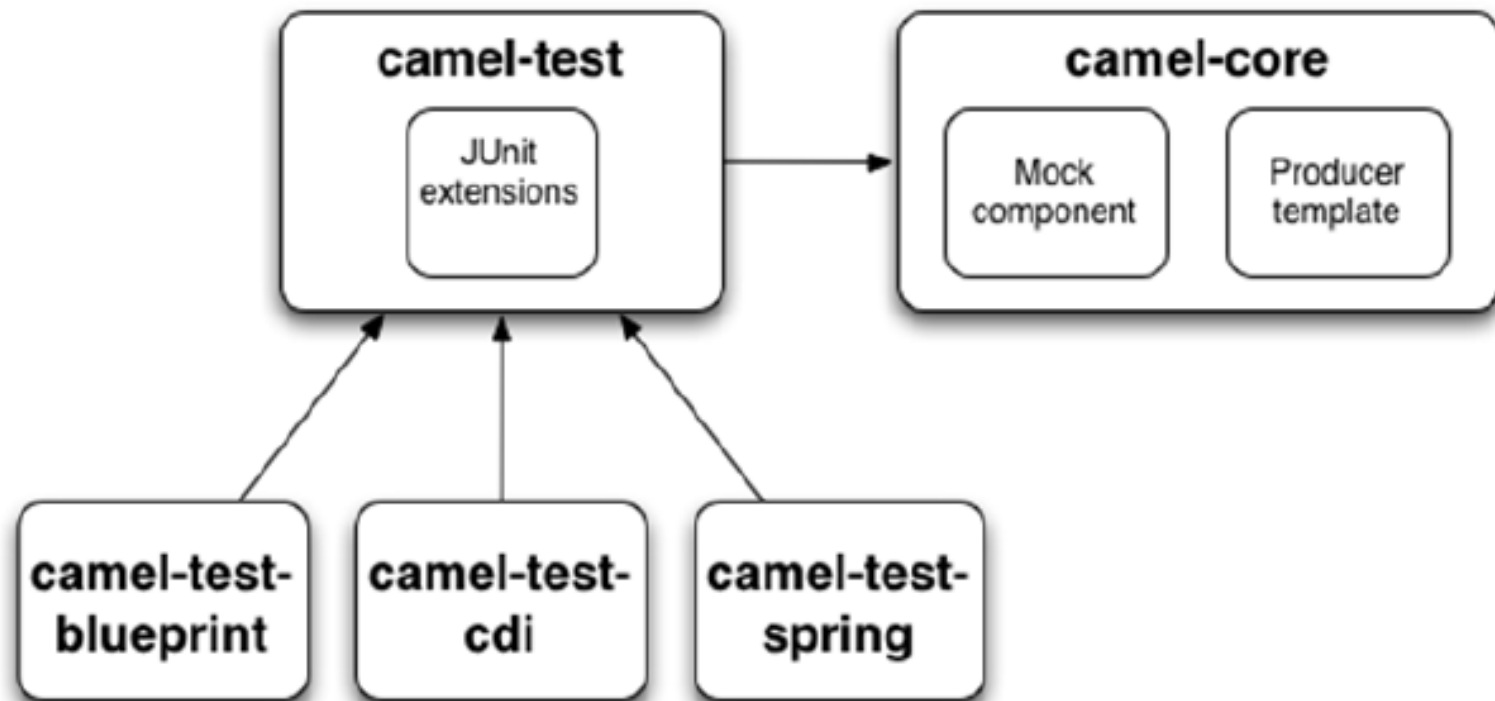
INFO - Route: cbr-route started and consuming from: Endpoint[file://work/cbr/input]



# Fragen

- Wie teste ich das Programmverhalten?
- Simulation von Komponenten?

# Camel Test Kit



- Korrektheit der eigenen Komponente

# JUnit

- Programmteile isoliert testen
- Wiederholbar, da selbstprüfend
- Testfälle direkt in Java programmiert

# Beispiel JUnit

```
import junit.framework.*;

public class EuroTest extends TestCase {

    public EuroTest(String name) {
        super(name);
    }

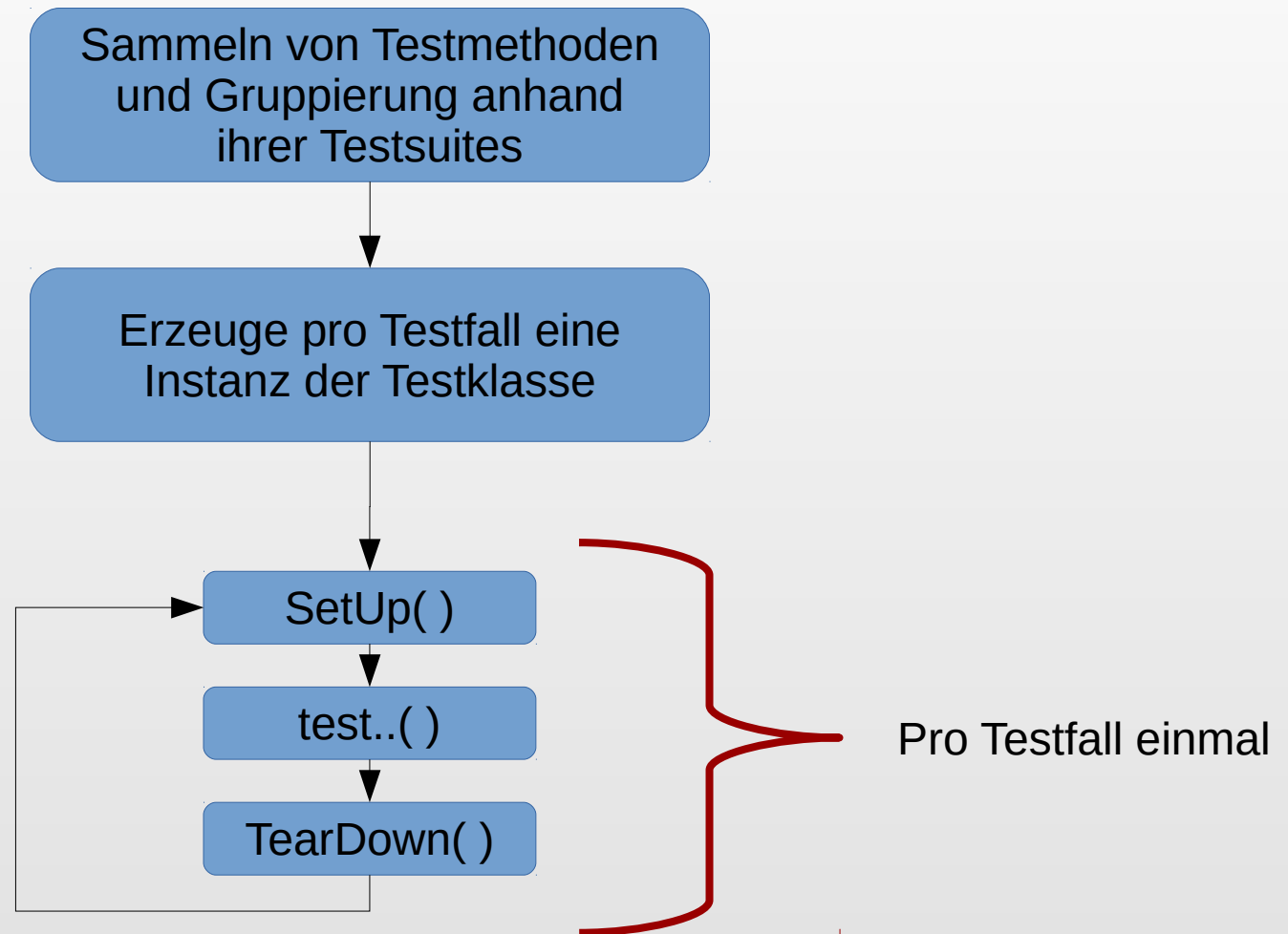
    public void testAmount() {
        Euro two = new Euro(2.00);
        assertTrue(2.00 == two.getAmount());
    }

    public static void main(String[] args) {
        junit.swingui.TestRunner.run(EuroTest.class);
    }
}
```

# Weiteres

- Testsuites
- Fixtures
  - SetUp( )
  - TearDown( )

# Testfall Lebenszyklus



# Testen von CamelRouten

- 1) Definiere Expectations
- 2) Sende In- Message
- 3) Verifiziere Ergebnis



# Weitere Abweichungen

**Listing 6.1 A first unit test using the Camel Test Kit**

```
package camelinaction;

import java.io.File;
import org.apache.camel.Exchange;
import org.apache.camel.builder.RouteBuilder;
import org.apache.camel.test.junit4.CamelTestSupport;
import org.junit.Test;

public class FirstTest extends CamelTestSupport {

    @Override
    protected RouteBuilder createRouteBuilder() throws Exception {
        return new RouteBuilder() {
            public void configure() throws Exception {
                from("file://target/inbox")
                .to("file://target/outbox");
            }
        };
    }

    @Test
    public void testMoveFile() throws Exception {
        template.sendBodyAndHeader("file://target/inbox", "Hello World",
            Exchange.FILE_NAME, "hello.txt");

        Thread.sleep(2000);

        File target = new File("target/outbox/hello.txt");
        assertTrue("File not moved", target.exists());
    }
}
```

- ① Defines route to test
- ② Creates hello.txt file
- ③ Verifies file is moved

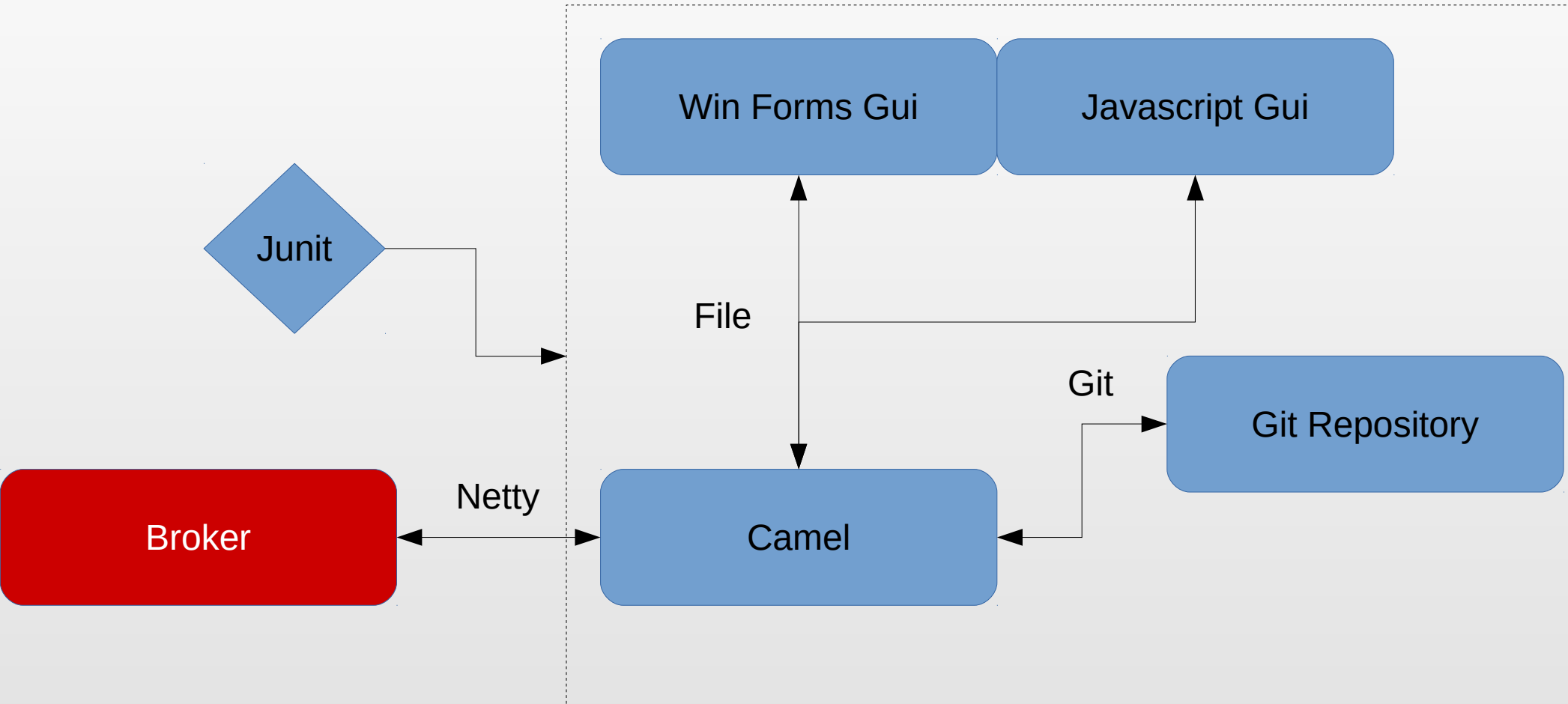
# Mock- Komponente

- Erlaubt Formulierung von Expectations für Endpoints

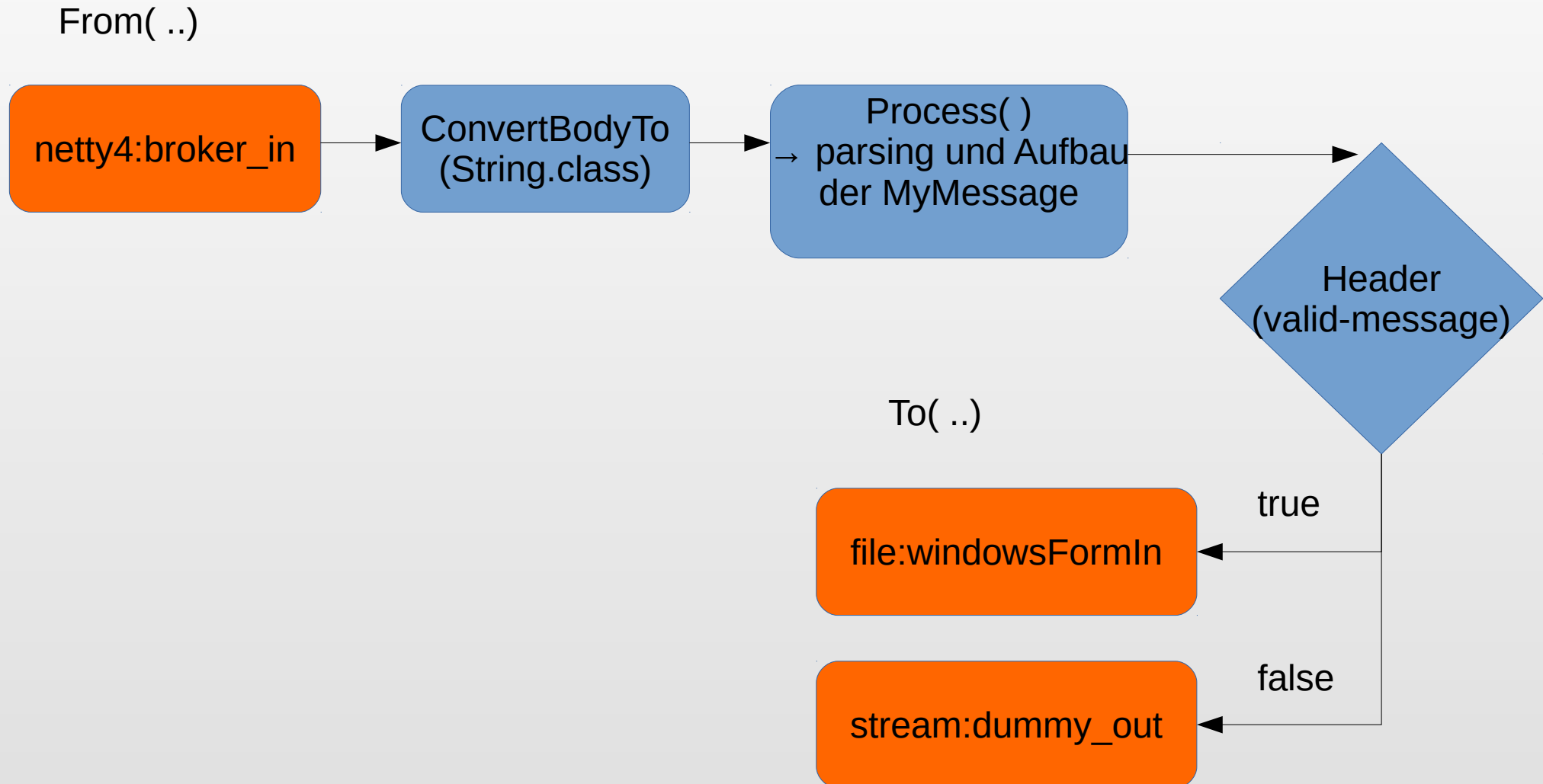
unter anderem korrekte...

- #erhaltenerNachrichten
- Reihenfolge
- Payloads
- Zeitspanne der Tests

# Anwendungsbeispiel: Brokerkommunikation



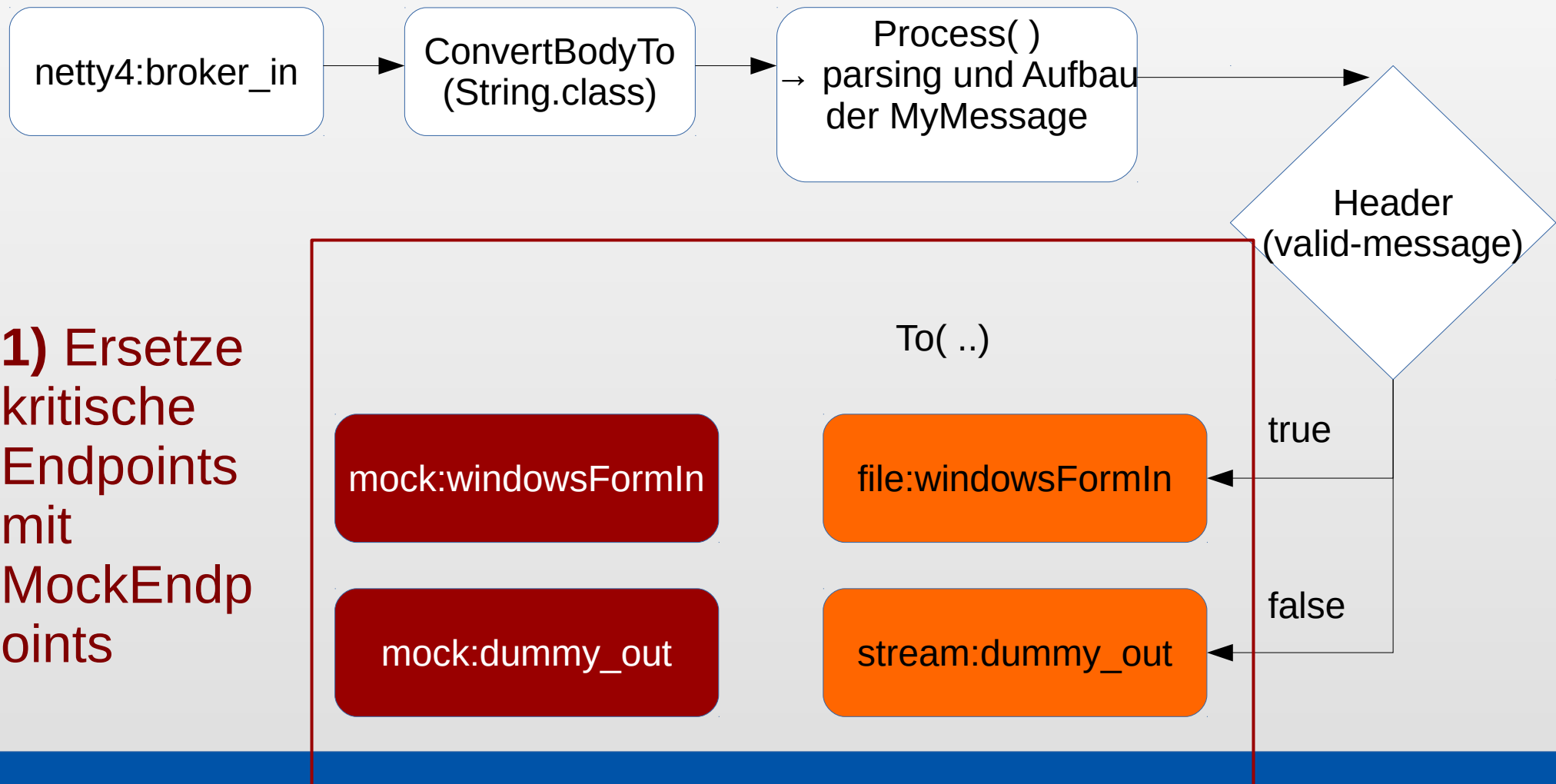
# Brokerkommunikation - Detailansicht



Werden die Nachrichten vom Broker korrekt gefiltert?

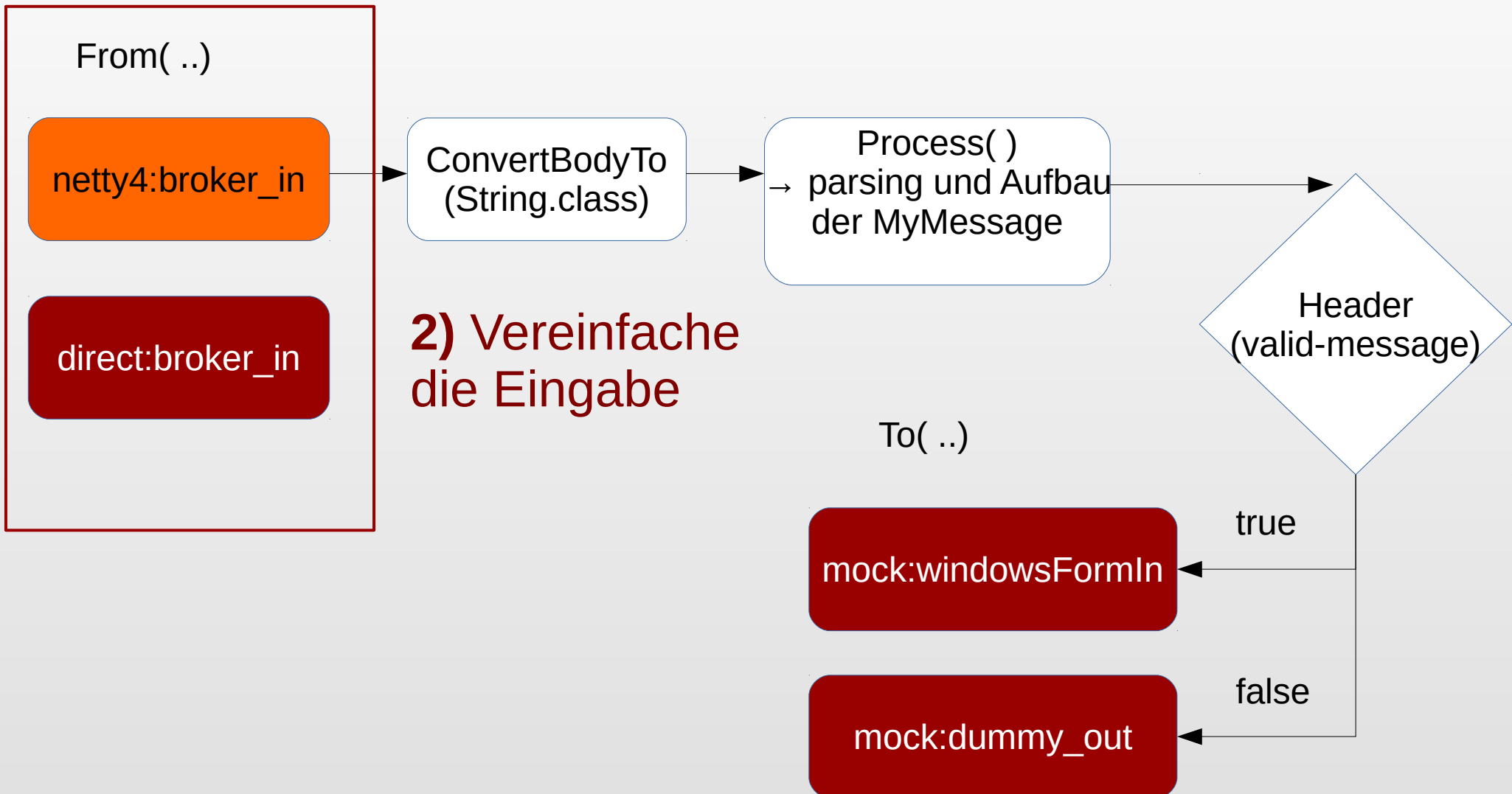
# Vereinfachung/ Anpassung der Route

From( ..)



Werden die Nachrichten vom Broker korrekt gefiltert?

# Vereinfachung/ Anpassung der Route



# RouteTest - Code

```
RouteTest.java
1 package gui0.tests;
2
3 import org.apache.camel.builder.RouteBuilder;
10
11 public class RouteTest extends CamelTestSupport{
12
13
14     //1. Erstelle vereinfachte Route
15     @Override
16     protected RouteBuilder createRouteBuilder() throws Exception {
17         return new RouteBuilder() {
18             public void configure() throws Exception {
19                 from("direct:broker_in")
20                     .convertBodyTo(String.class).process(new parsingProcessor())
21                     .choice().when(header("valid-message").isEqualTo(true))
22                     .to("mock:windowsFormIn").otherwise().to("mock:dummy_out");
23             }
24         };
25     }
26
27
28     //2. Schreibe Test
```

# RouteTest -Code f.

```
//2. Schreibe Test
@Test
public void testRoute() throws Exception {
    //nutze Methode der Basistestklasse, um auf die richtigen MockEndpoints zuzugreifen
    MockEndpoint mockWindowsFormsIn = getMockEndpoint("mock:windowsFormIn");
    MockEndpoint mockDummyOut = getMockEndpoint("mock:dummy_out");

    //2.1 Formuliere Expectations der Mocks
    //Sende im folgenden eine syntaktisch korrekte und zwei nicht korrekte Nachrichten
    mockWindowsFormsIn.expectedMessageCount(1);
    mockDummyOut.expectedMessageCount(2);

    //2.2 Initialisiere den Test durch senden einer In-Message
    template.sendBody("direct:broker_in", "Testnachricht -> dummyOut");
    template.sendBody("direct:broker_in", "Testnachricht -> dummyOut");

    //JsonObject
    MyMessage testMessage = new MyMessage();
    testMessage.setInstruction("register:gui");
    testMessage.setSender("netty4:test");

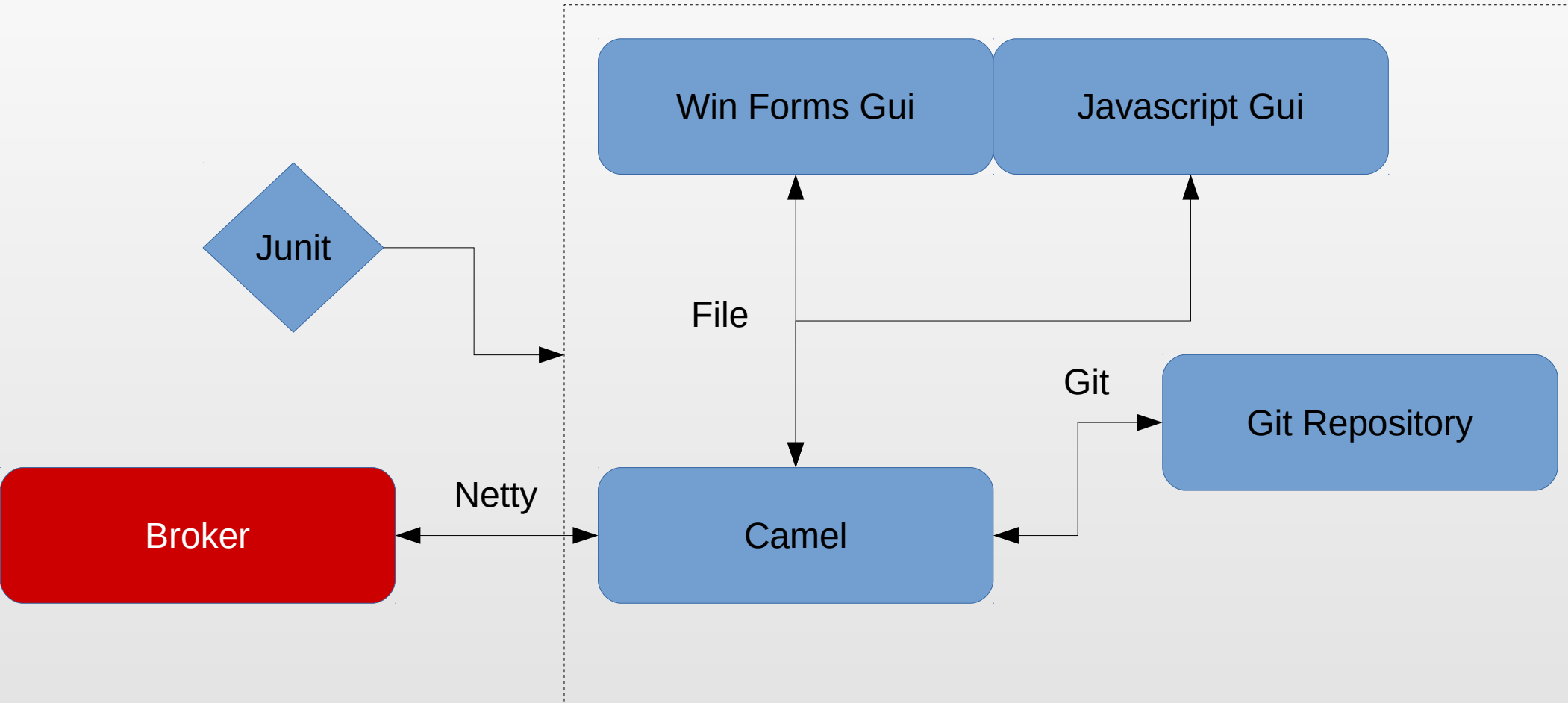
    template.sendBody("direct:broker_in", testMessage.toJSON());

    Thread.sleep(2000); //lasse Camel Zeit um die Nachricht zu verarbeiten

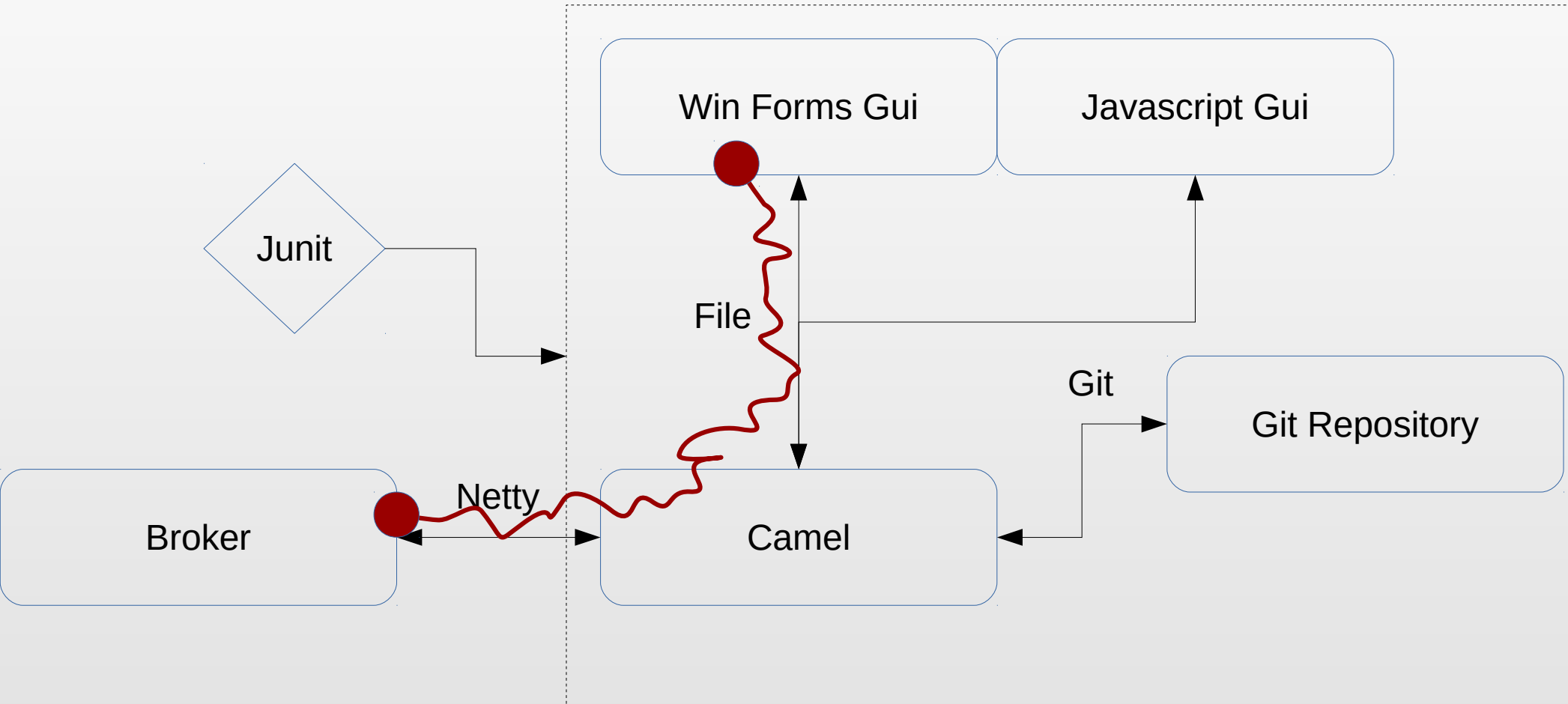
    //2.3 Verifiziere die MockEndpoints
    assertMockEndpointsSatisfied();
}
}
```



- Simulation von Komponenten



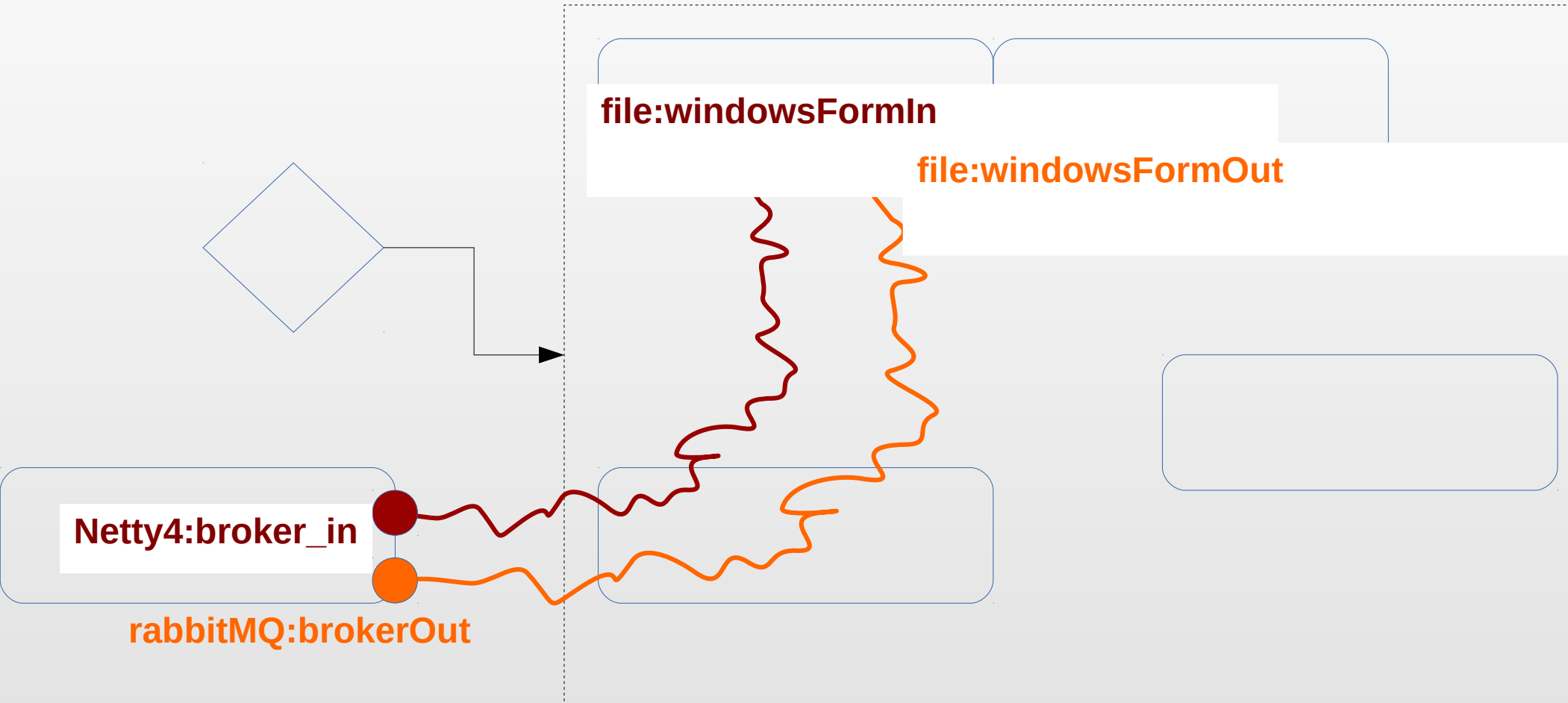
# Untersuchte Route: vom Broker zur Windows GUI



## Legende:

- **rot:** Broker zu WindowsGUI
- **orange:** Windows GUI zu Broker

# ...Der Rückweg

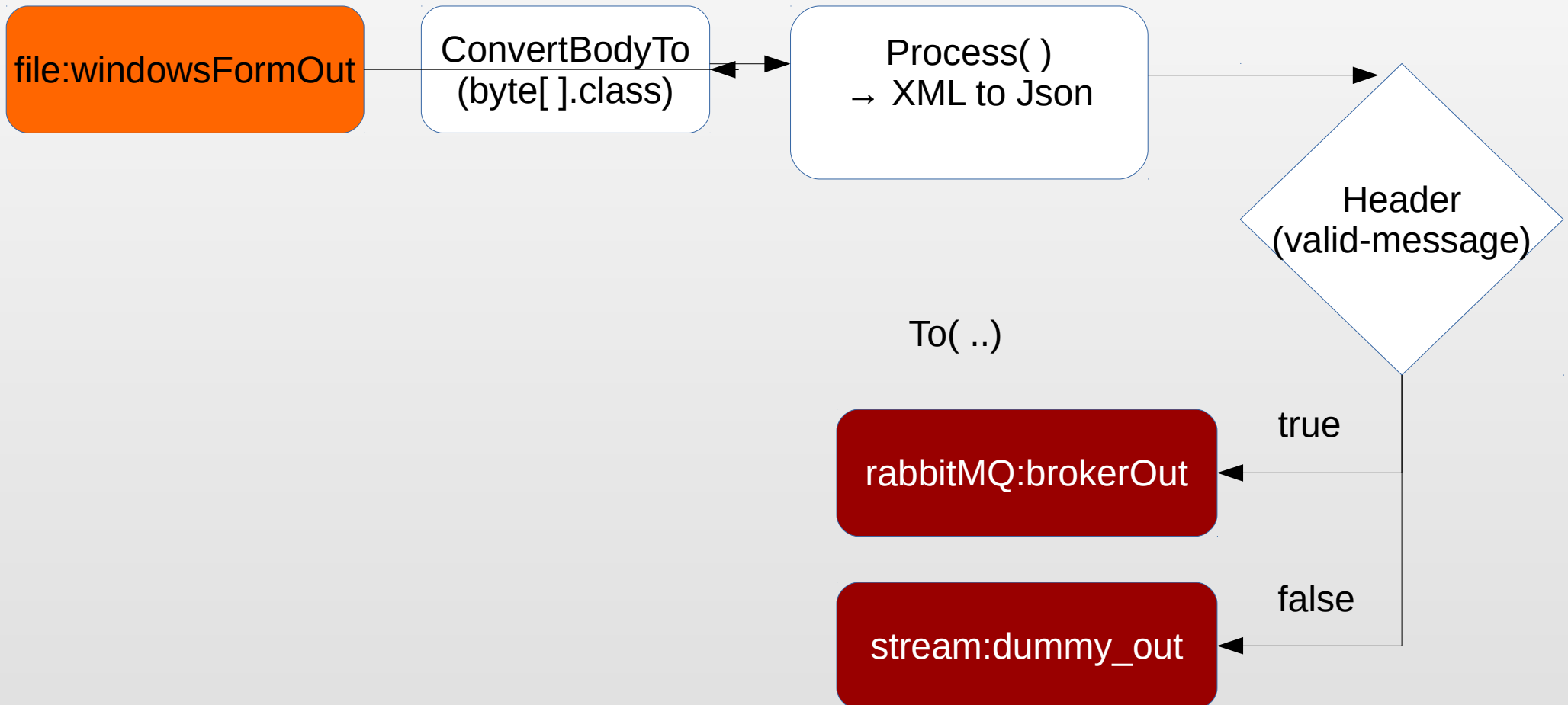


## Legende:

- **rot:** Broker zu WindowsGUI
- **orange:** Windows GUI zu Broker

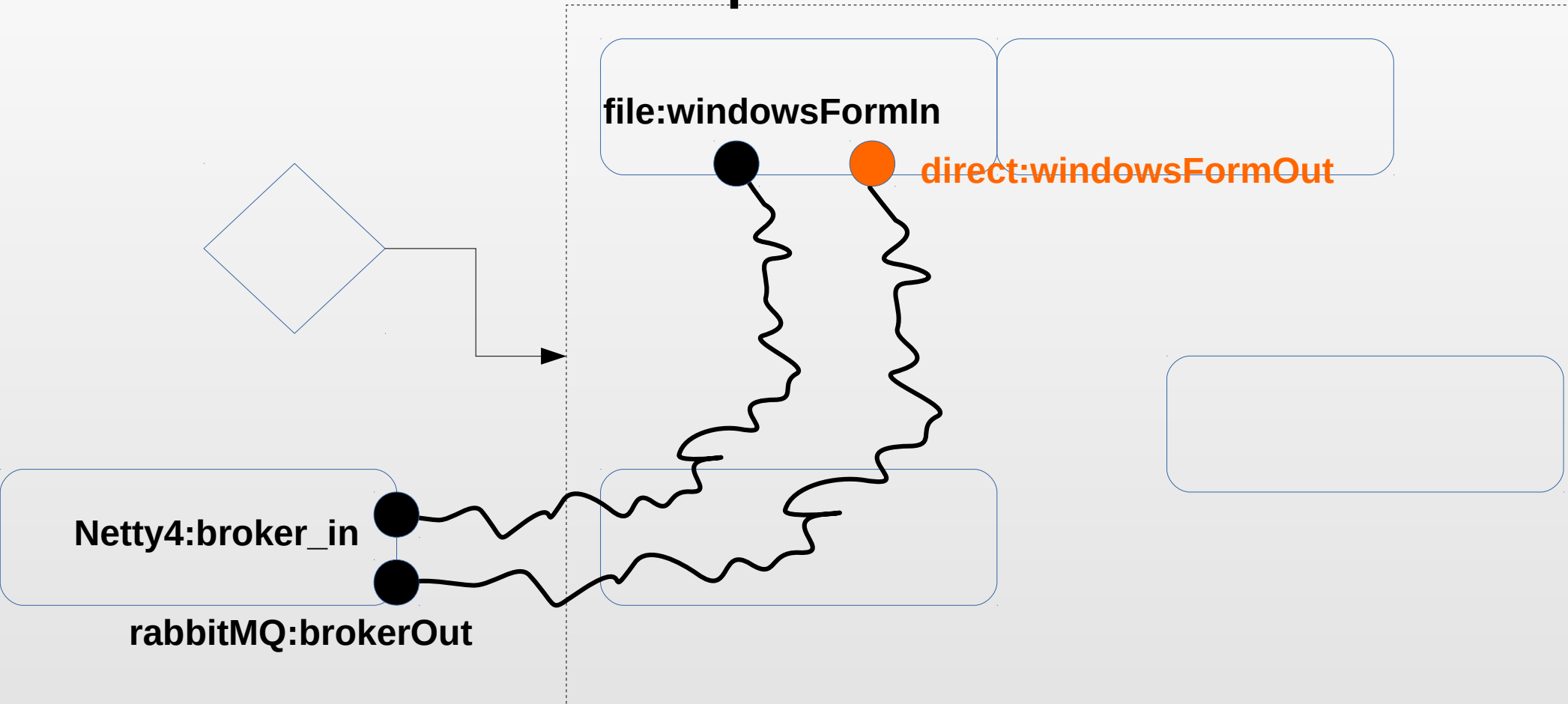
# Route: GUI zu Broker

From( ..)



## Vereinfachte Route

# - Nachrichteneingang über direct Komponente

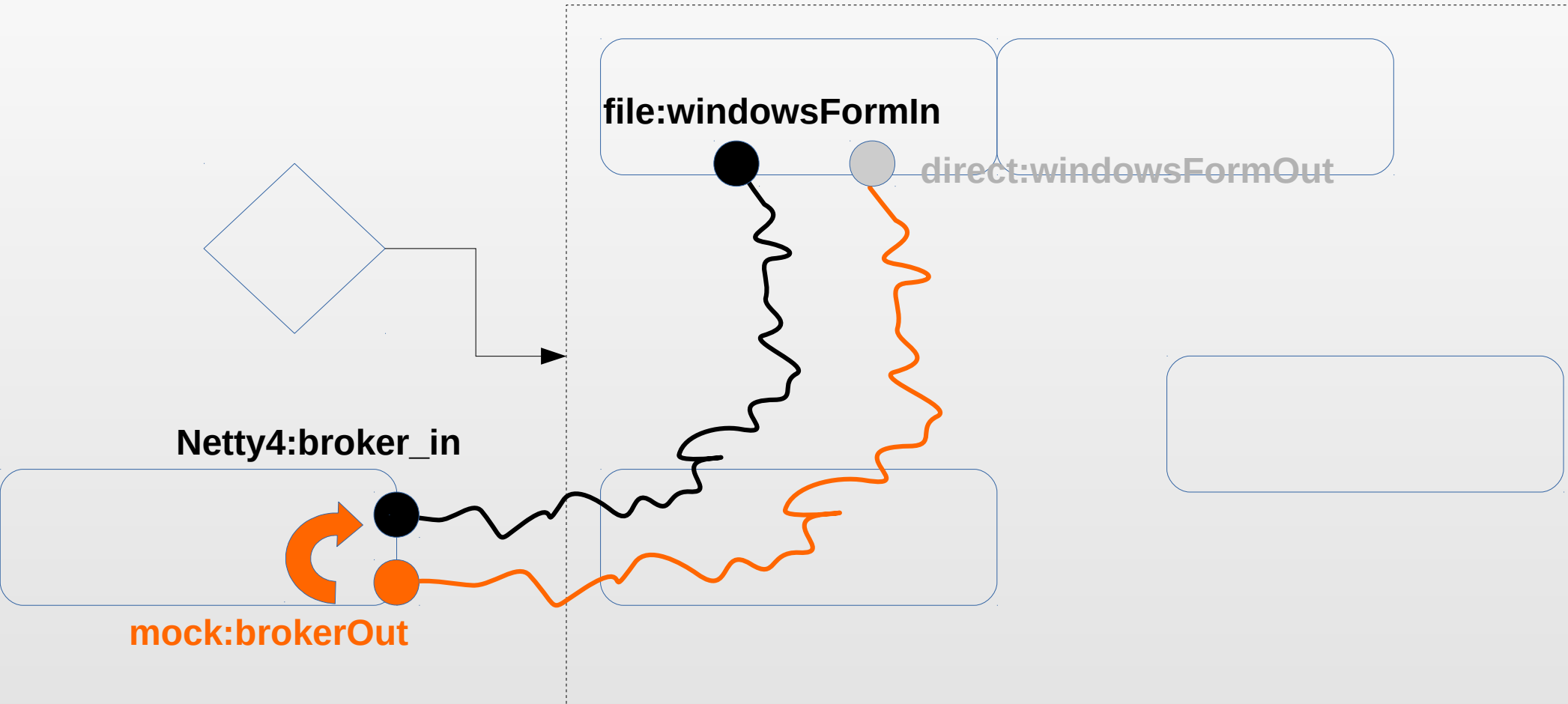


### Legende:

- **rot:** Broker zu WindowsGUI
- **orange:** Windows GUI zu Broker

# Vereinfachte Route

- Mock simuliert Komponente

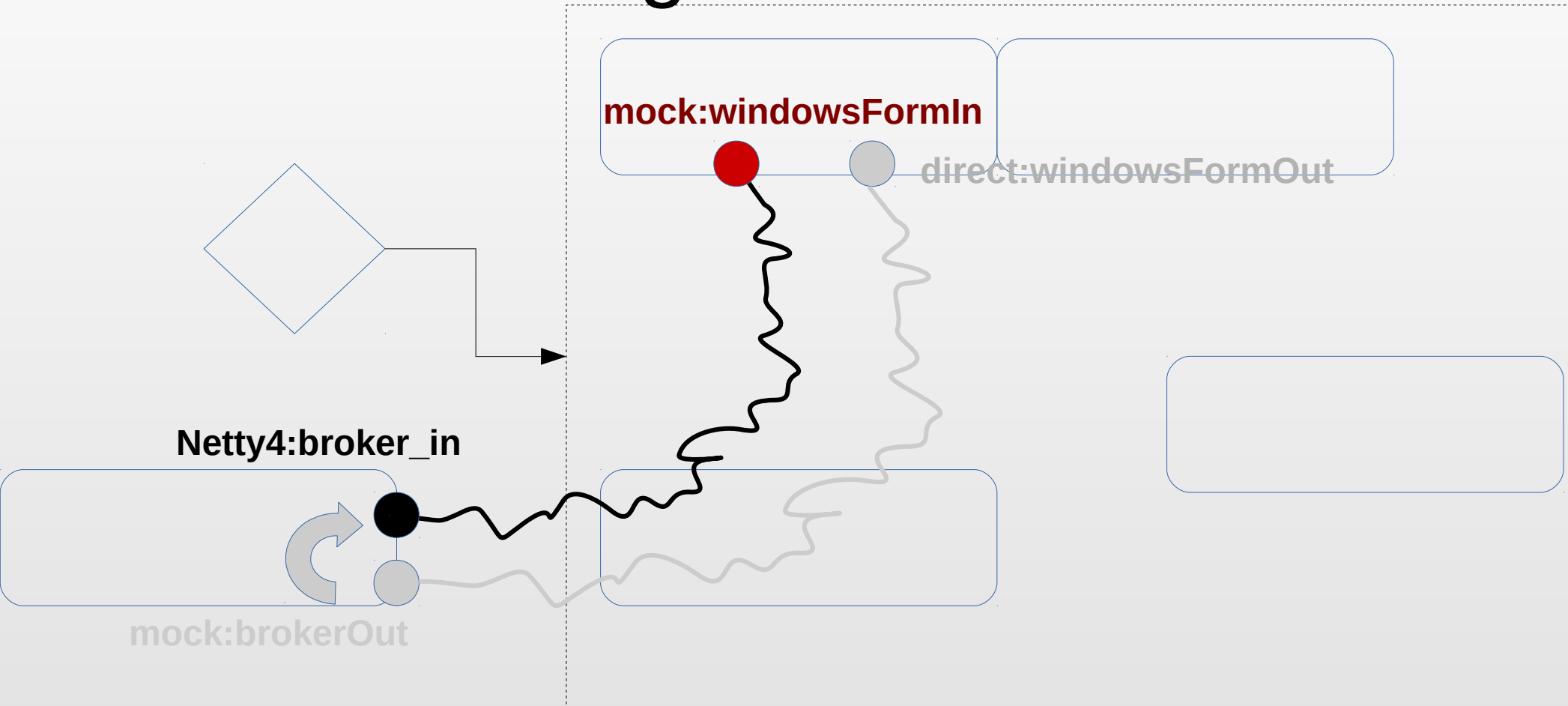


## Legende:

- **rot:** Broker zu WindowsGUI
- **orange:** Windows GUI zu Broker

# Vereinfachte Route

## - Mock zur Verifizierung des Ergebnisses



### Legende:

- **rot:** Broker zu WindowsGUI
- **orange:** Windows GUI zu Broker



```
//2. Schreibe Test

@Test

public void testRoute() throws Exception {

    //nutze Methode der Basistestklasse, um auf die richtigen MockEndpoints zuzugreifen

    MockEndpoint mockWindowsFormsIn = getMockEndpoint("mock:windowsFormIn");

    MockEndpoint mockDummyOut = getMockEndpoint("mock:dummy_out");


    //2.1 Formuliere Expectations der Mocks

    //Sende im folgenden eine syntaktisch korrekte und zwei nicht korrekte Nachrichten

    mockWindowsFormsIn.expectedMessageCount(1);

    mockDummyOut.expectedMessageCount(2);


    //Mock Komponente "broker_out" simuliert die Komponente Broker

    MockEndpoint mockBroker_out = getMockEndpoint("mock:broker_out");

    mockBroker_out.whenAnyExchangeReceived(new simulierterBrokerProcessor());


    //2.2 Initialisiere den Test durch senden einer In-Message

    template.sendBody("direct:windowsFormOut", "Testnachricht -> dummyOut");

    template.sendBody("direct:windowsFormOut", "Testnachricht -> dummyOut");


    //JsonTestObject

    MyMessage testMessage = new MyMessage();

    testMessage.setInstruction("register:gui");

    testMessage.setSender("netty4:test");


    template.sendBody("direct:windowsFormOut", testMessage.toJSON());


    Thread.sleep(2000);    //lasse Camel Zeit um die Nachricht zu verarbeiten


    //2.3 Verifiziere die MockEndpoints

    assertMockEndpointsSatisfied();
```

```
package gui0.tests;

import org.apache.camel.Exchange;

import org.apache.camel.Processor;

import org.apache.camel.ProducerTemplate;

import org.apache.camel.test.junit4.CamelTestSupport;

public class simulierterBrokerProcessor implements Processor {

    @Override

    public void process(Exchange exchange) throws Exception {

        ProducerTemplate template = exchange.getContext().createProducerTemplate();

        System.out.println("simulierender Broker leitet die Nachricht weiter");

        template.sendBody("netty4:tcp://localhost:8888?textline=true", exchange.getIn().getBody());

    }

}
```

Noch Fragen?