

Genesys call center integration with Digipass authentication system at SEB Latvia

by Agris Klimkans
Adventus Solutions
May 2016

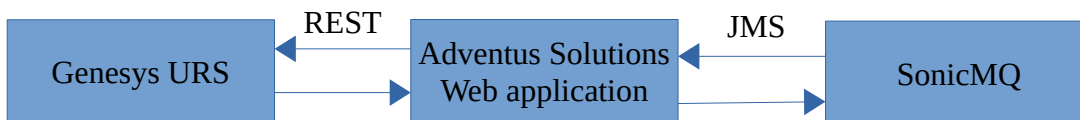
1. Introduction

A software component has been written for integrating the Digipass authentication services in the Genesys contact center. It enables authentication of customers during phone calls. Status of the customer is shown in the agent's Genesys Workspace Desktop Environment application.

2. Architecture

The software component is a web application, written in Java and deployed on Apache Tomcat server.

2.1. Authentication Workflow:



2.2. Tools/components:

Eclipse vMars.2 Release (4.5.2) with JAXB v1.0.200 plugin
Apache Maven v3.3.3
Apache Tomcat v9.0.0.M3
JDK v8u60

3. Installation and startup:

Copy the web application archive file (SEB_JMS_ORCHESTRA.war) to the Tomcat webapps folder. Restart Tomcat.

4. Operation

4.1. Client query function:

Request from Genesys URS:

GetChallengeCodeByPersonalCode?id=01020311111&connid=00a8027af2cd0fd0

where

id – personal code, connid – unique connection id

Responses:

Successful authentication:

customerId:I23317|username:I13317|idCode:010203-11111|firstName:JĀNIS|lastName:KALNIŅŠ|
userPhoneNumber:12345678|challengeCode:8568

Nonexisting customer:

error:NO_MATCH

Multiple customers:

error:MULTIPLE_MATCH

Generic error:

error:<error name>

4.2. Authentication code verification function:

Request from Genesys URS:

CheckAuthenticationCode?digipasscode=123456&username=I13317&connid=00a8027af2cd0cd7
where

digipasscode – 6 digit code from calculator

username – user name (received from previous function)

connid - unique connection id (same as in previous function)

Responses:

Success:

result:OK

Wrong code:

result:WRONGCODE

Generic error:

error:<error name>

4.3. Configuration

SEB_JMS_ORCHESTRA/WEB-INF/web.xml

```
<context-param>
    <description>JMS broker:port</description>
    <param-name>broker</param-name>
    <param-value>awebs.ad4.seb.net:6526</param-value>
</context-param>
<context-param>
    <description>JMS username</description>
    <param-name>usernameSonic</param-name>
    <param-value>tester1</param-value>
</context-param>
<context-param>
    <description>JMS password</description>
    <param-name>passwordSonic</param-name>
    <param-value>tester</param-value>
</context-param>
</context-param>
```

```

        <description>JMS queue</description>
        <param-name>queue</param-name>
        <param-value>SEB_SERVICES</param-value>
    </context-param>
    <context-param>
        <description>time in ms to wait for a reply on temporary queue. If 0,
blocks indefinitely until a message arrives</description>
        <param-name>connectionTimeout</param-name>
        <param-value>60000</param-value>
    </context-param>
    <context-param>
        <description>interval between PingPong messages, in ms</description>
        <param-name>pingPongInterval</param-name>
        <param-value>30000</param-value>
    </context-param>
    <context-param>
        <description>JMS Message attribute TTL, in ms</description>
        <param-name>TTL</param-name>
        <param-value>30000</param-value>
    </context-param>
    <context-param>
        <description>JMS Message attribute responseMsgTTL, in ms</description>
        <param-name>responseMsgTTL</param-name>
        <param-value>30000</param-value>
    </context-param>
    <context-param>
        <description>file with PingPong status (for NMS), contains 0 - no services
available, 1 - ok</description>
        <param-name>pingPongStatusFileName</param-name>
        <param-value>PingPongStatusFile.txt</param-value>
    </context-param>

```

4.4. Logging

Currently only in stdout and stderr. The files are in the Tomcat log folder.