

EUROPEAN MIDDLEWARE INITIATIVE

APEL SSM FUNCTIONAL DESCRIPTION

Document Version:	1.0
EMI Component Version:	2.0.0
Date:	13.02.2013

DOCUMENT CHANGE LOG

Version	Date	Comment	Author
1.0	13/02/2013	Initial Version	Will Rogers

Copyright notice:

Copyright (c) Members of the EMI Collaboration. 2013.

See <http://www.eu-emi.eu/about/Partners/> for details on the copyright holders.

EMI ("European Middleware Initiative") is a project partially funded by the European Commission. For more information on the project, its partners and contributors please see <http://www.eu-emi.eu>.

This document is released under the Open Access license. You are permitted to copy and distribute verbatim copies of this document containing this copyright notice, but modifying this document is not allowed. You are permitted to copy this document in whole or in part into other documents if you attach the following reference to the copied elements: "Copyright (C) 2013. Members of the EMI Collaboration. <http://www.eu-emi.eu>".

The information contained in this document represents the views of EMI as of the date they are published. EMI does not guarantee that any information contained herein is error-free, or up to date.

EMI MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, BY PUBLISHING THIS DOCUMENT.

1. INTRODUCTION

1.1. PURPOSE

The SSM (Secure STOMP Messenger) is used to send messages between computers using the STOMP protocol. It acts as both a sender and a receiver; only the sending component is supported in EMI.

1.2. DEFINITIONS AND ACRONYMS

STOMP [1]: Text based messaging protocol supported by various message brokers

SSM: Secure STOMP Messenger – a simple program used to send files using STOMP

dirq [2], [3]: A messaging library used to store messages on a local filesystem

2. OVERALL DESCRIPTION

A sending SSM reads files from a local filesystem, and then sends them, using the configured message brokers, to the configured queue. A receiving SSM listens to the queue and writes the received messages to a filesystem at the receiving end.

Messages are signed using an X509 certificate at the sender's side, and may optionally be encrypted using the server's certificate. The receiver verifies the signature and decrypts if necessary.

The python-dirq library is used to store messages on the filesystem. Messages may be added to SSM either by writing them directly as files, or using the python [2] or Perl [3] dirq libraries.

3. ARCHITECTURE

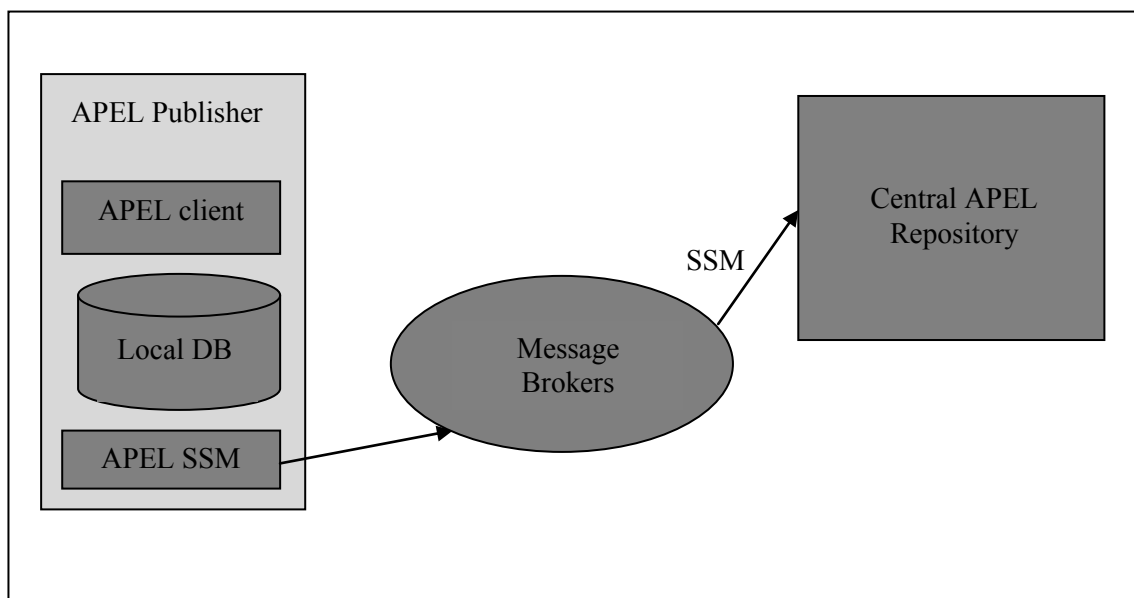


Figure 1: SSM and APEL Publisher

4. REFERENCES

[1] <http://stomp.codehaus.org/>

[2]: python-dirq: <http://pypi.python.org/pypi/dirq>

[3] Perl Directory::Queue: <http://search.cpan.org/~lcons/Directory-Queue/>