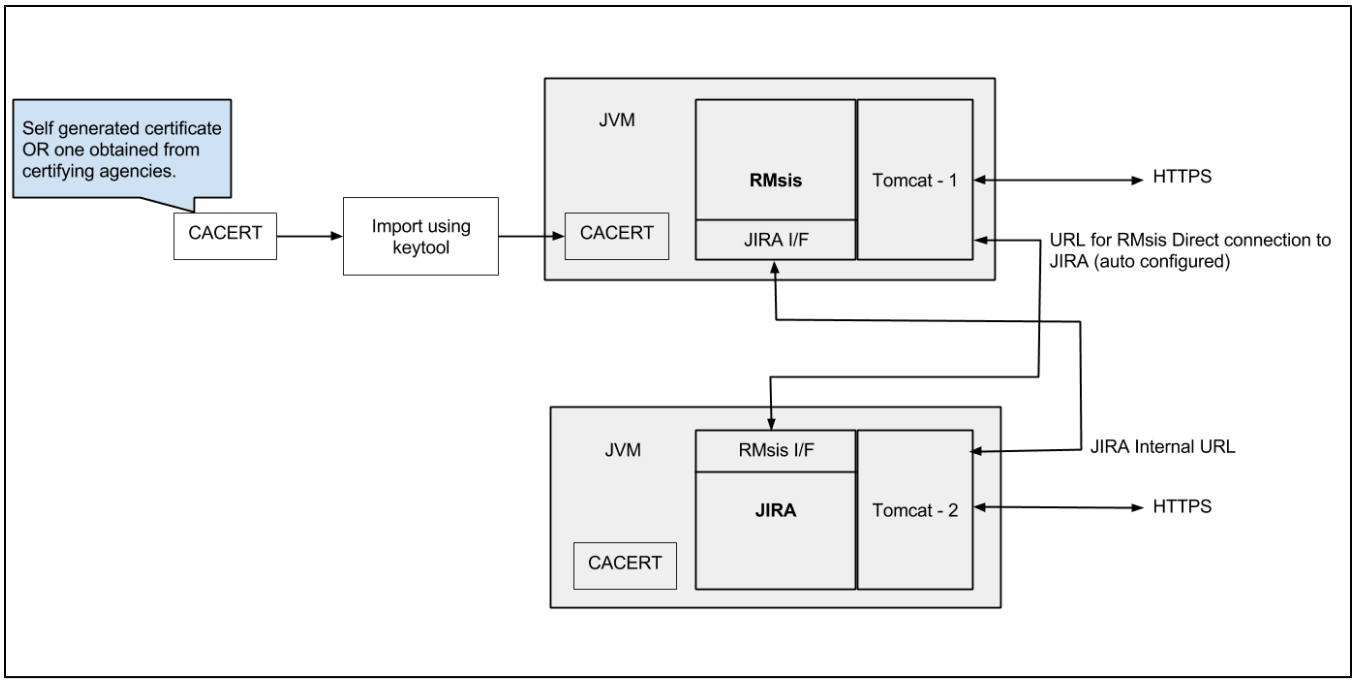
Running RMsis on SSL or HTTPS

Note

A key point related to mixed use of HTTP and HTTPS.

- Some of the browsers are now blocking HTTP calls from HTTPS pages.
- As per some problems reported recently, the users are not able to use RMsis from recent versions of Chrome OR Firefox.
- A solution is to
    - run both RMsis and JIRA on HTTPS
    - OR run both JIRA and RMsis on HTTP

HTTPS Setup Overview

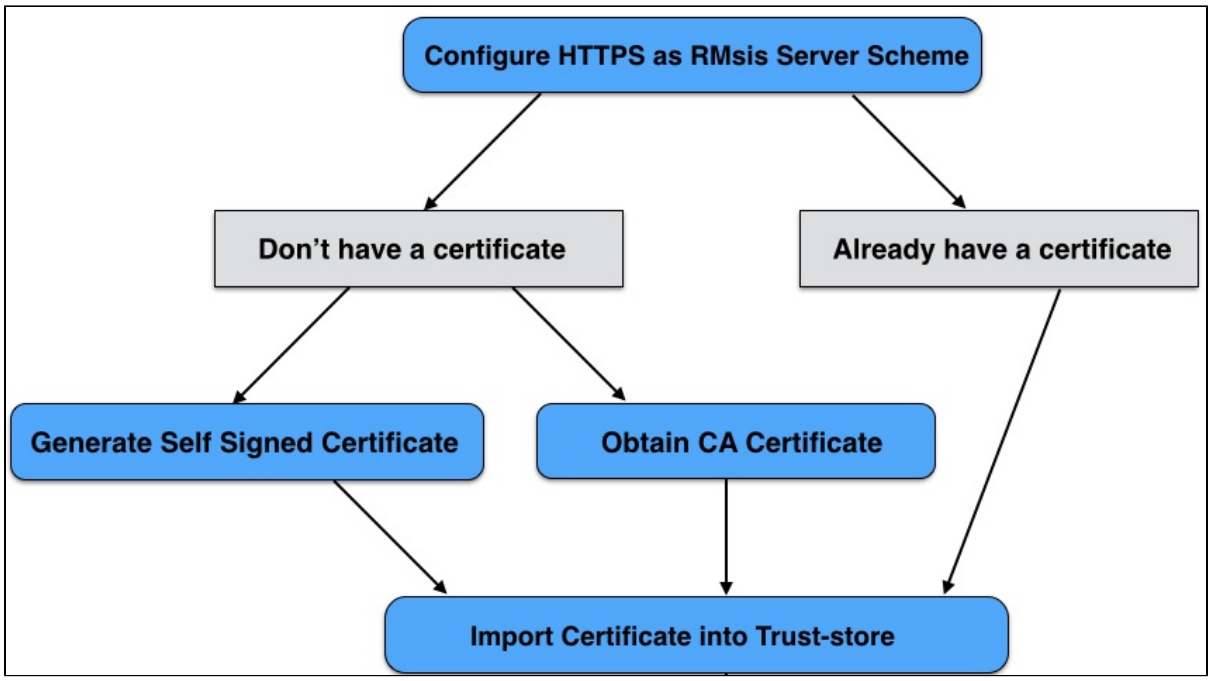


RMsis comprises of two components:

- RMsis Server, which runs independently on Tomcat
    - RMsis communicates with JIRA through the JIRA Internal URL specified in the configuration.
- A Plugin which integrates with JIRA.
    - This plugin communicates with a specific (auto configured) port of RMsis.

It may be noted that RMsis and JIRA could be using the same JRE OR different JRE. If they are running on different JRE, adequate care should be taken to ensure that the certificates are installed at the right locations.

Steps for configuration



SSL Certificate for RMsis

In order to run RMsis on SSL (over https), a certificate must be created and registered with RMsis. Please note that

- This document assumes Tomcat, while installing the security certificate.
- RMsis supports
    - JKS
    - PKCS#11
    - PKCS#12

Obtain a Certificate to import into truststore, either self-signed or ca-signed

A note on certificates

In the SSL world, certificates fall into two major categories: self-signed and CA-signed

- **Self-Signed**: These are certificates that have not been digitally signed by a CA, which is a method of confirming the identity of the certificate that is being served by the web server. They are signed by themselves, hence the name self-signed.
- **CA-Signed**: A certificate that has had its identity digitally signed by a Certificate Authority (CA) like Verisign.

Generating a Self Signed Certificate

Self signed certificates are useful in cases where you require encryption but do not need to verify the website identity. They are commonly used for testing and on internal corporate networks (intranets).If you are using RMsis within a closed group or Intranet, you can use a self signed certificate.

Run **keytool** on command line, which is available with JAVA 1.6, and enter the responses against the prompt (a sample is shown here). The keytool utility will prompt you for two passwords: the keystore password and the key password for Tomcat. **You must use the same value for both passwords**.

```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA

Enter keystore password: // Password for your java keystore, it is 'changeit' by default
What is your first and last name?
        RMSIS_SERVER // Enter fully qualified server name; for example jira-rmsis.optimizory.com
What is the name of your organizational unit?
  [Unknown]: ORG_UNIT_NAME
What is the name of your organization?
  [Unknown]: ORG_NAME
What is the name of your City or Locality?
  [Unknown]: CITY
What is the name of your State or Province?
  [Unknown]: STATE
What is the two-letter country code for this unit?
  [Unknown]: US
Is <CN=ORG_UNIT_NAME, OU=ORG_UNIT, O=ORG_NAME, L=CITY, ST=STATE, C=US> correct?
  [no]: yes

Enter the key password for <key-alias>
        <RETURN if same as keystore password>: <> // Press Return here and do not specify a password.
```

Now export the certificate to use it with Tomcat

```
$JAVA_HOME/bin/keytool -export -alias tomcat -file file.cer

// file.cer contains the certificate details to be imported in the next step.
```

NOTE: Due to the certificate not being signed by a Certification Authority (CA), users may be prompted that the site is untrusted and may have to perform several steps to "accept" the certificate before they can access the site. This usually will only occur the first time they access the site.

Importing the Certificate into Trust Store

If you already have an existing certificate available (for example from a CA like Verisign), please perform the following operation as root (or sudo)

```
$JAVA_HOME/bin/keytool -import -alias tomcat -file file.cer
```

Assuming your certificate is called "file.cer" whether obtained by a CA or self generated, the above command will add this certificate to the Truststore.

A note regarding location of keystore

The keytool utility creates the keystore as a file called .keystore in the current user's home directory.

- For Unix/Linux the home directory is likely to be /home/<username> as determined by the user.home system property.
- For Windows it is likely to be C:\Documents And Settings\<UserName>.

If this file does not exist, it will be created.

Use keystore switch to specify existing keystore (if any)

```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore <PATH_TO_KEYSTORE>/keystoreFile.jks
```

Installing Public Certificate for RMsis, when JIRA is running on HTTPS

If JIRA is running on HTTPS, a Public Security Certificate is expected to be installed and accessed by RMsis. For RMsis versions 1.5.2 and later, the system will automatically accept the default certificate configured for JIRA. In case of an exception, you will need to add security certificate of JIRA Server in java trust store which resides at <JRE_PATH>/lib/security/cacerts. Below is a small how-to for certificate installation.

- Ensure that JIRA Server is running.
- Unzip and extract **InstallCert.class** and **InstallCert$SavingTrustManager.class** to some location (from where java path is accessible). [Download ZIP]
- Run **InstallCert** binary using command line.
    - $ java InstallCert <JIRA_SERVER>:<JIRA_SERVER_PORT>
    - In case you are using the default port, JIRA_SERVER_PORT parameter is optional
    - Follow the subsequent instructions in the program
- This will create new file with name jssecacert in current directory. Just copy this file to <JRE_PATH>/lib/security/cacerts.
- Restart JIRA and try again with RMsis.

References

- Apache Tomcat 7 - SSL Configuration : http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html#Prepare_the_Certificate_Keystore
- Running JIRA over SSL or HTTPS : http://confluence.atlassian.com/display/JIRA044/Running+JIRA+over+SSL+or+HTTPS