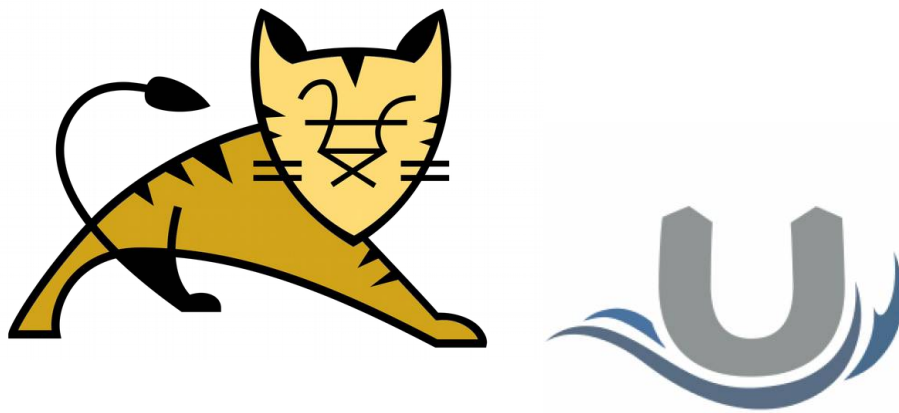


Tomcat native 2



20/03/16

Version 1.0

Supervised by Jean-Frederic Clere (Red Hat)

Project Coordinator: Dr. Hugues Mercier

Co-instructor: Pr. Jacques Savoy and Pr. Peter Kropf

TEAM AND CONTACT INFORMATION:

Jocelyn Thode (jocelyn.thode@unifr.ch)

Simon Brulhart (simon.brulhart@unifr.ch)

Table of contents

1. PROJECT DESCRIPTION	4
1.1. PROJECT CONTEXT	4
1.2. GOALS AND OBJECTIVES OF THE PROJECT	4
2. PROJECT ORGANIZATION	5
2.1. RESPONSIBILITY DISTRIBUTION	5
2.2. INTERACTIONS WITH THE CLIENT	5
3. METHODOLOGY	6
3.1. STATE OF THE ART	6
4. CONSTRAINTS AND ELEMENTS OF RISK	7
5. DELIVERABLE GOODS	8
6. Efforts and schedule	9

Version monitoring

Monitoring of the version in this project organization document enables following modification activities progress within the project. The version is changed only if a major change takes place. Otherwise, only a decimal is changed (1.0 to 1.1)

Version	Update date	Page modified
1.0	20.03.16	1-8

1. Project description

For this project we are working with Red Hat which is an multinational software company that provides open-source softwares and technical support to enterprises.

At the moment, the two biggest Java web servers are Tomcat and Undertow. Both rely on TLS/SSL to encrypt and secure communications between a client and a server. To achieve good performance Tomcat uses Tomcat-native to have access to OpenSSL directly. It uses Apache Portable Runtime (APR) to do so. Some work has been done in Tomcat to use Java Native Interface (JNI) instead of APR to interface with OpenSSL. Undertow isn't compatible with Tomcat Native at the moment.

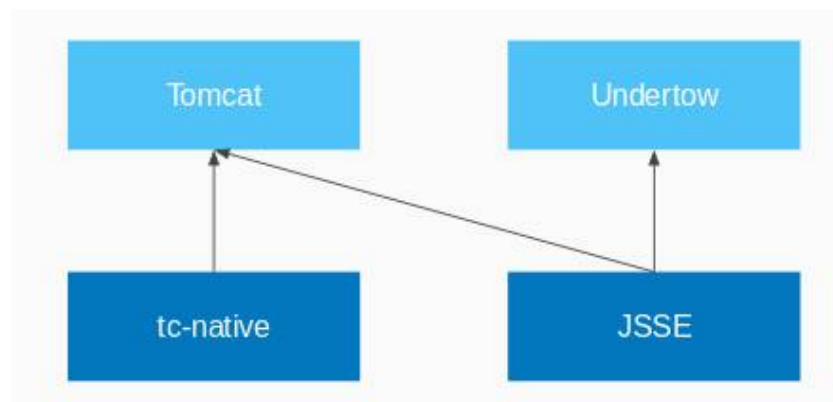


Figure 1: Which SSLEngine works with which web server

TLS/SSL is critical in terms of performance and security. Nowadays it is a must-have but it can slow down an entire application.

We want to use OpenSSL as it is fast. However, we don't want to use APR as this can make the code much more complicated. The best compromise seems to be using JNI calls to OpenSSL. This is still slower than APR but not by much while providing an easier to maintain code base.

1.1. Goals and objectives of the project

Our project aims to refactor Tomcat Native to drop APR and only rely on JNI calls to interface with OpenSSL. This should make Tomcat Native much more maintainable while still having great performances when using TLS/SSL. These changes should also make Tomcat Native usable in Undertow while keeping the compatibility with Tomcat.

This means we need to :

- Remove all APR code present in Tomcat Native

- Abstract Tomcat Native to be used in Tomcat and Undertow
 - Write JNI calls
- Provide a good documentation so that the open-source community can use the project
- Run benchmarks to test the implementation

Another goal is to add JNI calls to access more features from OpenSSL in Tomcat Native.

2. Project organization

This project requires us to read some documentation and source code. We also need to write reports and presentations. To keep the codebase clean, we decided to adopt the same strategy as Numa de Montmollin last year and split our work in two different repositories on Github.

[tomcat-native2-doc](#) is the repository dedicated to the documentation. We put every reports and presentation in the repo. This repo also contains a wiki which hosts the logbook, workplan and our different articles. The articles are useful to share our findings and more importantly not to forget about them.

[tomcat-native2](#) is the repository dedicated to the code. This repo includes our implementation, and the documentation directly concerning it. For example, the Javadoc and a readme explaining how to build it and what is its main purpose can be found on it.

2.1. Responsibility distribution

As our project involves reading a lot of source code and removing code. We usually work together, sharing our findings. We meet most of the time using a voip program and assign the tasks of the day from there. If we have any question we can always contact each other on the spot.

2.2. Interactions with the client

As we have to work with Stuart Douglas for Undertow who lives in Australia, Rémy Maucherat for Tomcat who lives in France and Jean-Frederic Clere who works in Neuchâtel we usually interact by email or Hangout if we want a spoken conversation.

We also write a logbook of our daily activities as well as some articles on our understanding of the project on Github, so that Jean-Frederic Clere and Rémy Maucherat can follow our progress.

3. Methodology

For this project we do not have any choice in the technology used. We work with Java and specifically JNI. We have to modify Tomcat Native so that it works with both Tomcat and Undertow without APR.

As stated earlier, we almost always get together on a voip program so that we can easily and quickly ask questions to each other. Otherwise we try to make a workplan for the day and then split the tasks evenly from there.

We mostly use IntelliJ Idea for our development environment. We use git coupled to Github as the source version control. We may also use Github issues in the future to track problems we may encounter during the implementation.

Every tool we use is free (as in freedom). They are licensed under the Apache License 2.0. Our work is completely open-source as required by Red Hat and will probably be under the Apache Licence 2.0 as it is a fork of Tomcat Native.

3.1. State of the art

Our project is not related to any papers. However there are documentation available for each code base we use. Namely:

- Tomcat (<http://tomcat.apache.org/>)
- Tomcat Native(<http://tomcat.apache.org/native-doc/>)
- Undertow (<http://undertow.io/documentation.html>) .

4. Constraints and elements of risk

We have three main risks in our project :

1. Having to understand projects such as Tomcat, Tomcat Native and Undertow is a big task and could make us lose a lot of precious hours while not understanding a lot.

However, we can always ask Rémy, Jean-Frederic and Stuart for help if we are stuck on a particular problem as we already did multiple times. Since the projects are open source we can also rely on the community to provide us good documentation and help.

2. Evaluating the time to adapt Tomcat can be deceptive as we have to remove code instead of writing it. This could lead to some conflicts where we remove too much or not enough or we simply don't know what to remove to make it effective.

Stuart has already done some experiments with Undertow which can help us visualize what to remove. Having our workplan checked by Rémy and Jean-Frederic can also help us in that regard.

3. Adapting Tomcat Native to work with both Undertow and Tomcat might not be the best solution. An other approach could be to use the code written by Stuart Douglas for Undertow and modify it to work with Tomcat. This would require us to write a lot more code than the Tomcat Native route and could make us lose a lot of time if we don't choose the right alternative at the beginning.

Rémy and Jean-Frederic think that going the Tomcat Native route is the easiest and we agree. Removing code is almost always easier than writing it.

5. Deliverable goods

We will deliver an OpenSSL wrapper using JNI calls based on Tomcat Native and working with both Tomcat and Undertow as well as documentation for this wrapper.

These will be delivered on Github in the repository presented in chapter 2. This should help the community understand and continue working on our project.

6. Efforts and schedule

This work plan is extracted from here (<https://github.com/jocelynthode/tomcat-native2-doc/wiki/Workplan>)

This isn't a definitive work plan and may change over time.

7. Credits

The Tomcat logo is from :

<https://upload.wikimedia.org/wikipedia/commons/thumb/7/7b/Tomcat-logo.svg/2000px-Tomcat-logo.svg.png> (Consulted on 20.03.2016)

The Undertow logo is from :

<https://avatars3.githubusercontent.com/u/2001898?v=3&s=400> (Consulted on 20.03.2016)