Tomcat Native 2

Initial plan

Jocelyn Thode and Simon Brulhart
jocelyn.thode@unifr.ch
simon.brulhart@unifr.ch

28th March 2016

Tomcat Native 2 is a project mandated by Red Hat. It aims at providing a JNI wrapper for OpenSSL for use in both Undertow and Tomcat. This project is the practical part of the R&D Workshop course given at the University of Neuchâtel as part of the Joint Master in Computer Science.

MASTER IN
COMPUTER
SCIENCE

redhat.

Contents

List of Figures

1 Project description

1.1 Project context

For this project we are working with Red Hat which is a multinational software company that provides open-source software and technical support to enterprises[1]. At the moment, the two biggest Java web servers are Tomcat[2] and Undertow[3]. Both rely on TLS/SSL to encrypt and secure communications between a client and a server. To achieve good performance Tomcat Native[4] provides access to OpenSSL in Tomcat. It uses Apache Portable Runtime (APR) to do so. Unfortunately, Undertow isn't compatible with Tomcat Native at the moment. Until last year, the APR would manage the encrypted sockets by itself. However this implied a lot of C code to maintain. To ease maintenance, some work has been done to use OpenSSL APIs directly instead. This works well enough despite some acceptable performance penalty. However the APR still provides the bindings to these APIs. We now want to strip the current solution of the APR and make the result compatible with Undertow.

Figure 1 shows an overview of the different way to use TLS/SSL in Tomcat.

1.2 Goals and objectives of the project

Our project aims to refactor Tomcat Native to drop APR and only rely on our own JNI wrapper to interface with OpenSSL. This should make Tomcat Native much more maintainable while keeping great performance when using TLS/SSL. The project also aims to make the resulting Tomcat Native 2 usable in Undertow while keeping the compatibility with Tomcat. This means we need to :

1. Study the source code of Tomcat Native, Undertow and some experiments already done to use OpenSSL in Undertow.
2. Remove all APR code present in Tomcat Native
3. Abstract Tomcat Native to be used in Tomcat and Undertow
4. Provide good documentation so that the open-source community can use the project
5. Run benchmarks to test the implementation on multiple platforms
6. Extend the OpenSSL support by adding JNIs for more features

Figure 2 shows what is expected at the end of the project.

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 code base 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 documenting the progress of our project. We put every report and presentation in the repo. This repo also contains a wiki which hosts the logbook, work plan 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, it should contain a ReadMe describing the building process and the main purpose of the project.

2.1 Responsibility distribution

Jocelyn Thode Student working on the project.
jocelyn.thode@unifr.ch

Simon Brulhart Student working on the project.
simon.brulhart@unifr.ch

Jean-Frederic Clere Project supervisor at Red Hat.
jfclere@redhat.com

Rémy Maucherat Co-Supervisor at Red Hat.
rmaucher@redhat.com

Stuart Douglas Works at Red Hat. Developer on Undertow.
sdouglas@redhat.com

Dr. Hugues Mercier Workshop coordinator.
hugues.mercier@unine.ch

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 VoIP software 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 activity 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 chat room so that we can easily and quickly share our findings and ask questions to each other. Otherwise we try to make a work plan 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 source code 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

At the moment, only Tomcat has access to OpenSSL through its library Tomcat Native. Unfortunately Undertow has to rely on the SSL implementation given by JSSE to provide SSL/TLS support. This leads to bad performance in Undertow.

On the other hand, Tomcat through Tomcat Native uses APR to access OpenSSL which is not ideal as this adds a lot of C code to maintain and isn't that useful anymore.

Our project should streamline Tomcat Native so that it can be used in both Undertow and Tomcat, giving access to a high-performance implementation of TLS/SSL to Undertow while making it easier to maintain the source code for Tomcat as well.

4 Constraints and elements of risk

4.1 Constraints

The main constraint of our project is that everything must be open-source. This forces us to have really good documentation as our source code is completely visible and will be used by the community afterwards.

An other constraint is the fact that we have to work with people all around the world. This can create some problems when waiting on answer because of timezone differences.

4.2 Risks

We have three main risks in our project:

1. Having to understand projects such as Tomcat, Tomcat Native and Undertow is a big task. We could get lost reading code and documentation for hours without really understanding anything.

   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 documentation and help.

2. Evaluating the time needed 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 work plan 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 the Tomcat Native route is the easiest and we agree. Removing code is almost always easier than writing it.

5 Deliverable goods

At the end of the project 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 our project and continue working on it.

6 Efforts and schedule

The full work plan and its associated logbook is available on the documentation repository referred in chapter 2. We also reproduced it in Figure 3

Our project is split up along the fixed deadlines/tasks specified by the R&D Workshop :

- **Deadline 1** is the deadline concerning this report. It defines what has to be done for the project as well as reading documentation.
- **Deadline 2** focuses on removing APR from Tomcat Native.
- **Deadline 3** focuses on making Tomcat Native 2 working on both Tomcat and Undertow.
- **Deadline 4** aims at extending OpenSSL support in Tomcat Native 2.

Each task is defined more precisely in the next sub-chapters.

6.1 Task 1

The first task is entirely dedicated to understand what is expected of us and how the different projects such as Tomcat Native work. Reading documentation will be done throughout the entire project, but here it is especially important as we have to understand the main concepts.

This deadline finishes with the hand-in of this plan. The project is on good tracks at the moment.

6.2 Task 2

In this second task, we want to understand more precisely what is needed to keep the OpenSSL engine for JSSE working.

We then will write JNIs for OpenSSL to replace the ones from the APR. At this point we won't need the APR anymore, so we will completely remove it from the Tomcat Native code base. This task should be done by the 10th April 2016.

6.3 Task 3

At the end of the third task, we aim at having a fully working Tomcat Native 2 for both Tomcat and Undertow. To achieve this goal we need to understand thoroughly how the JSSE SSL Engine and Java sockets are used in both web servers as well as understand how the JSSE OpenSSL engine works in Tomcat and how could we modify it to make it work in Undertow.

6.4 Task 4

Finally, we will have to benchmark Tomcat Native 2 with Red Hat to see how it behaves in different conditions. Then, if time permits, we will implement more features in Tomcat Native 2 regarding OpenSSL. For example, implementing support the OpenSSL handshake callback would be really useful as right now we have to rely on a hack to know when the handshake is completed.

Implementing support for proprietary large frames would also more performance on some systems.

Finally, implementing different security options for OpenSSL would also be a possibility.

7 References

[1] *Red Hat*. Red Hat. URL: `http://www.redhat.com/en/about` (visited on 25/03/2016).

[2] *Tomcat*. Apache Foundation. URL: `http://tomcat.apache.org/` (visited on 25/03/2016).

[3] *Undertow Documentation*. Red Hat. URL: `http://undertow.io/documentation.html` (visited on 25/03/2016).

[4] *Tomcat Native*. Apache Foundation. URL: `http://tomcat.apache.org/native-doc/` (visited on 25/03/2016).

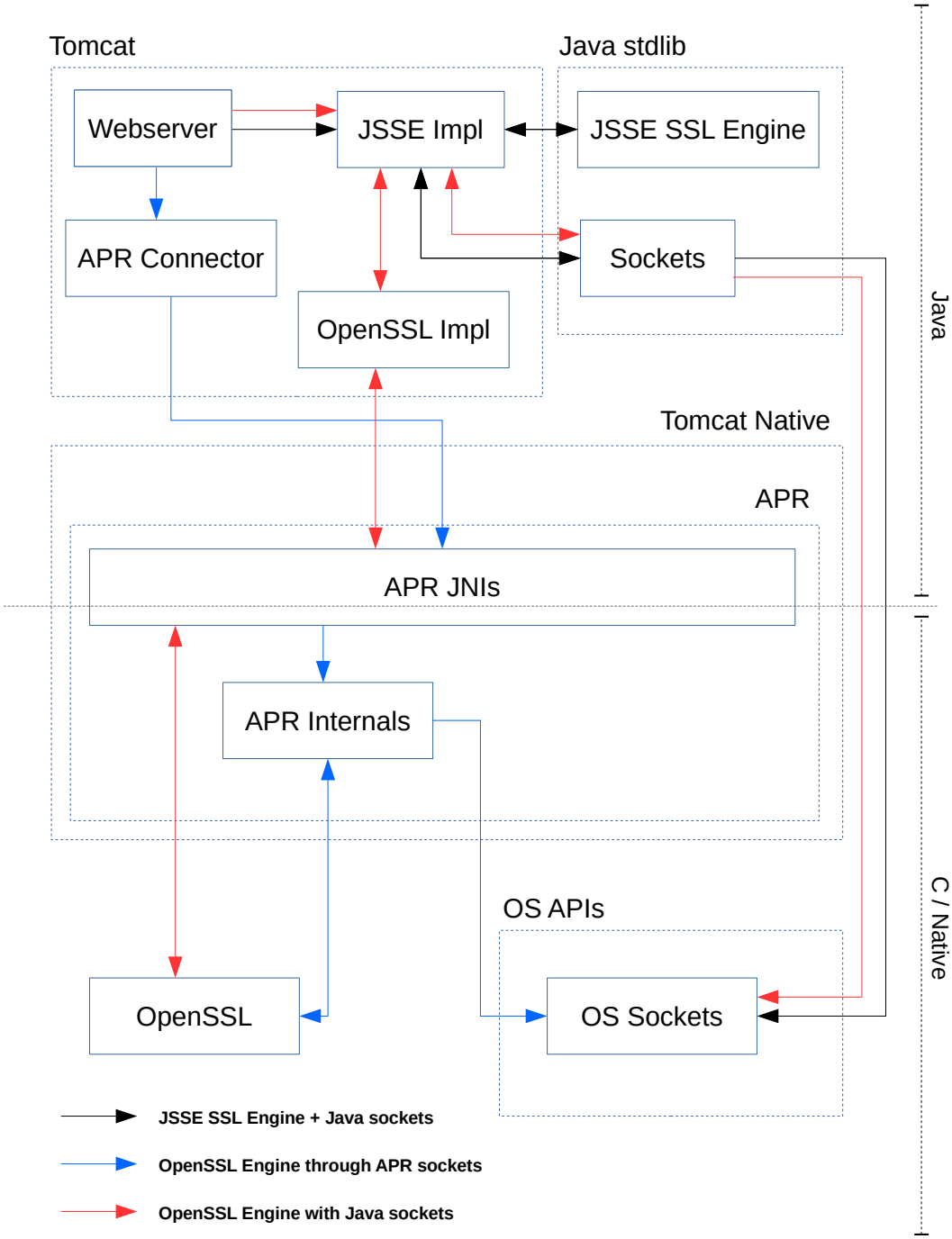

Figure 1: **Diagram showing the different possibilities to use TLS/SSL in Tomcat.**

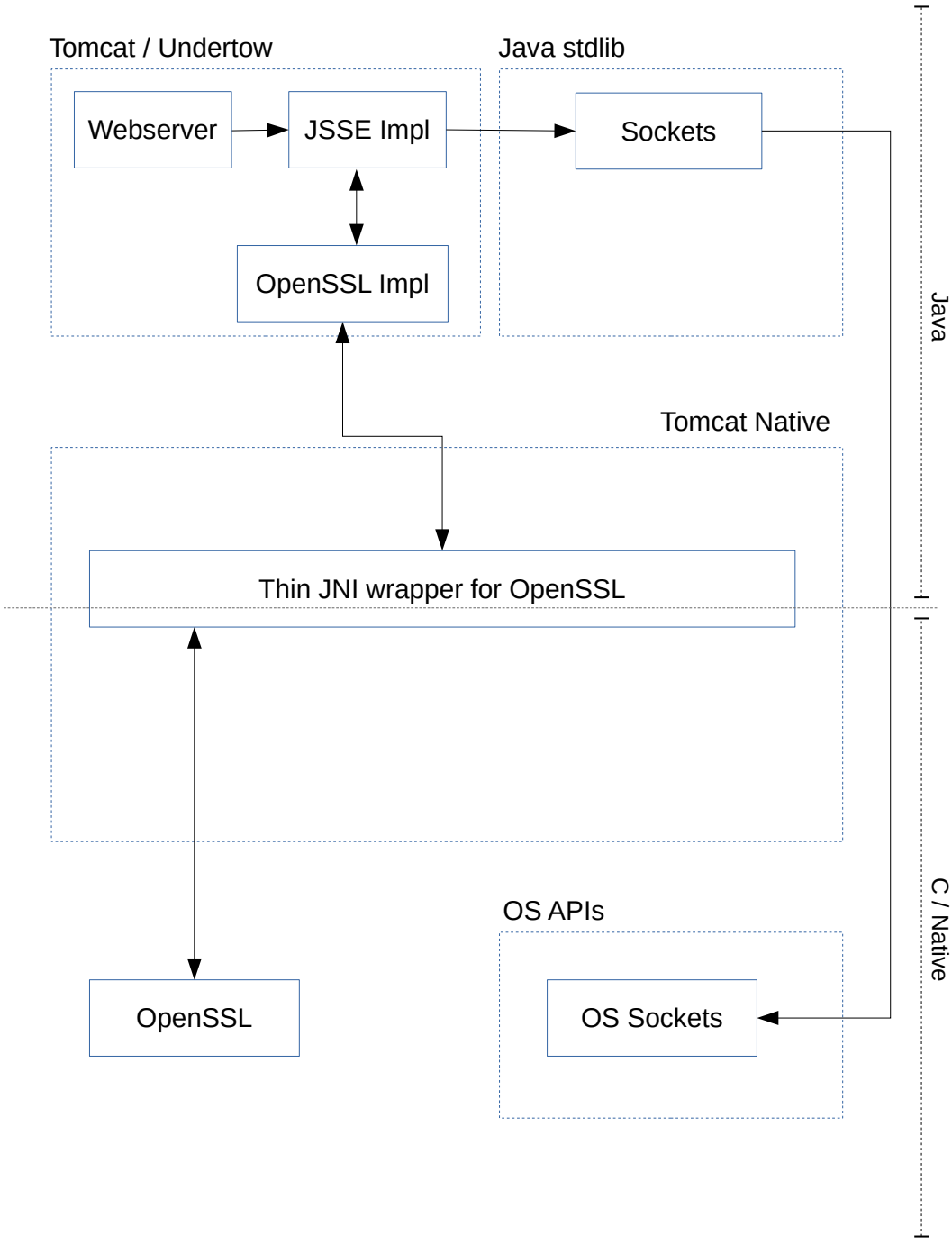

Figure 2: **Diagram showing the expected end result of our project.**

Task	estimated Hours
Set git repositories up	1
Search the documentation	2
Study doc and code	8
Prepare initial presentation	4
DEADLINE 0.5: 18.03.2016	**17**
Improve presentation	1
Presentation rehearsal	2
Write initial plan	4
DEADLINE 1: 27.03.2016	**7**
Understand which code is currently needed for Tomcat OpenSSL JSSE engine	10
Write the JNIs for OpenSSL and replace the ones from the APR	25
Remove all code related to the APR from tc-native	10
DEADLINE 2: 10.04.2016	**45**
Understand how Tomcat and Undertow use Java sockets and the JSSE SSL engine	10
Understand how the JSSE OpenSSL engine works in Tomcat and how it could work in Undertow	10
Adapt Tomcat-native to make it work both with Tomcat and Undertow	25
DEADLINE 3: 08.05.2016	**45**
Benchmark with Red Hat the resulting Tomcat-Native when used with Tomcat and Undertow	8
Implement support for the OpenSSL handshake callback	10
Implement support for large frames	10
Implement more security options for OpenSSL	6
DEADLINE 4: 29.05.2016	**34**
Write the report	10
Final presentation	10
DEADLINE 5: ~ 10.06.2016	**20**
TOTAL	**168**

Figure 3: **Workplan table from the wiki**