

JaffreProposal

<<Navigation(children)>>

- Jaffre Project Proposal
 - Abstract
 - Proposal
 - Rationale
 - Initial Goals
 - Current Status
 - Meritocracy
 - Community
 - Core Developers
 - Alignment
 - Known Risks
 - Orphaned Products
 - Inexperience with Open Source
 - Homogeneous Developers
 - Reliance on Salaried Developers
 - Relationships with Other Apache Products
 - A Excessive Fascination with the Apache Brand
 - Documentation
 - Initial Source
 - Source and Intellectual Property Submission Plan
 - External Dependencies
 - Cryptography
 - Required Resources
 - Mailing Lists
 - Subversion Directory
 - Issue Tracking
 - Other Resources
 - Initial Committers
 - Affiliations
 - Sponsors
 - Champion
 - Nominated Mentors
 - Sponsoring Entity

Jaffre Project Proposal

This is a draft version of the Jaffre proposal.

Author: Alexander Veit <<MailTo(alexander.veit AT jaffre DOT org)>>

Date: March 2009

Online version: <http://wiki.apache.org/incubator/JaffreProposal>

Status: .

#{renderedContent}

Abstract

Jaffre is a lightweight RPC library for the Java platform.

Proposal

Jaffre is a lightweight RPC library for the Java platform. It is designed to be extensible, robust, efficient, and easy to use.

Rationale

There are many RPC technologies for the Java platform.

Few of them limit themselves to connect only Java VMs with Java VMs, but try to be platform neutral. Platform neutrality often adds significant overhead to the development process, or comes with performance and/or memory consumption penalties (see e.g. SOAP based web services). On the other hand, RMI, as a Java-only technology, is often cumbersome to configure and to use.

Jaffre overcomes these limitations, and has additional benefits:

- It is not invasive when deployed in a JVM. The only required dependency is the JRE itself.
- It uses POJOs as service implementation classes. Clients use proxys that implement the POJOs interface to communicate with the server.

- It does not need fancy decoration stuff like annotations or XML configuration files.
- It promises to allow quite complex object graphs to be marshalled. In fact to be java.io.Serializable is the only requirement for marshalled objects.
- Jaffre does not need supplementary services such as registries or mediation services.
- It is easy to use. A service that exposes a given POJO, together with a simple socket connector can be set up in seven lines of code. A client that remotely calls the POJO's interface can be set up with five lines of code.

A simple benchmark may illustrate Jaffre's efficiency. 20000 remote calls of the interface

```
public interface ATest {  
    public String echo(String toBeEchoed);  
}
```

with string lengths from 1 to 20000 on the loopback interface of a Xeon E5420 @ 2.50GHz workstation and JDK 1.6.0_13 take 11800 ms (0.6 ms per call) with an unencrypted, and 25700 ms (1.3 ms per call) with a SSL encrypted connection.

Initial Goals

The initial goal is to revise the code and make it as robust and fault tolerant as possible.

Then a public beta version of Jaffre should be released. To decide if the project is worth to survive the incubation process, it is most important to see the strengths and weaknesses of the chosen approach in practice.

Current Status

The current code base has been developed by Alexander Veit in his spare time. Samples and test cases show that the basic concepts work. Performance tests show the efficiency of Jaffre.

Meritocracy

The initial committer is familiar with the meritocracy principles of Apache. He will follow the normal meritocracy rules also with other potential contributors.

Community

Jaffre strength comes primarily from it's simplicity. In analogy to a famous quotation from RFC 1425 a simple library should tend to ubiquity. If this is true, the development of a vital community will merely be a matter of time.

Core Developers

Currently Alexander Veit is the only core developer of this project.

Alignment

Jaffre will be a new species in the zoo of ASF's remoting technologies.

Known Risks

- It may turn out that people are quite comfortable with already available remoting technologies for the Java platform, so that there's no market for another RPC library.
- It may turn out that Jaffre's simple approach is too simplistic to satisfy the needs of real-world projects.

Orphaned Products

The current Jaffre code base has about 2000 lines of code. An experienced programmer will be able to understand the whole Jaffre code in less than an hour.

So as long as Jaffre will have users, being orphaned is unlikely.

Inexperience with Open Source

The initial committer has some experience with open source communities as a user (mostly ASF and Codehaus).

Homogeneous Developers

With one initial committer developers are as homogeneous as it could be. Given the manageable code base of roughly 2000 lines this should not be a big problem.

Reliance on Salaried Developers

Jaffre currently does not depend on salaried development.

Relationships with Other Apache Products

Jaffre will use Apache Maven as a build system, and Apache Ant to build samples.

A Excessive Fascination with the Apache Brand

The author uses Apache software since eight or nine years, and regards the Apache project as one of the most valuable sources of Java software in the world.

Documentation

Public documentation beyond the [JavaDoc](#) in the source code is not yet available.

Initial Source

The initial source code is available at <http://www.jaffre.org/>.

Source and Intellectual Property Submission Plan

Apache would receive all source and documentation under the Apache Individual Contributor License Agreement V2.0.

External Dependencies

Jaffre does not have any required dependencies other than a JRE.

However, SLF4J is recommended to integrate logging with Jaffre.

Cryptography

Jaffre uses the native capabilities of Java to support TLS.

Required Resources

Mailing Lists

- [jaffre-private](#)
- [jaffre-dev](#)
- [jaffre-commits](#)
- [jaffre-user](#)

Subversion Directory

<https://svn.apache.org/repos/asf/incubator/jaffre>

Issue Tracking

JIRA : [Jaffre \(JAFFRE\)](#)

Other Resources

None

Initial Committers

Alexander Veit [alexander dot veit at jaffre dot org](mailto:alexander.dot.veit@jaffre.org)

Affiliations

None.

Sponsors

We kindly request the Incubator PMC to accept sponsorship for this proposal.

Champion

n.n.

Nominated Mentors

n.n.

Sponsoring Entity

Incubator