

# JSpiritProposal

## Abstract

jSpirit will be a platform to develop efficiently enterprise class applications for SaaS with real Multi-tenant support and cloud deployment. jSpirit is an integration framework for multi-tenant applications with associated developpement tools. The platform tends to be an implementation of the Application layer of a cloud computing stack. The integration framework is clearly multi-tenancy and product line oriented.

## Proposal

jSpirit will provide technical foundation on which application developper will create enterprise software distributed as services. jSpirit will implement global and out-of-box architecture supporting multi-tenancy. As multi-tenancy, we mean architecture that share the same application for multiple client, with support of specifics (by tenant) comportements. The technical foundation will include an integration framework designed for simplify and abstract technical complexity of J2EE for the final developer, a set of tools to industrialize production of applications, a complete (and in the future customizable) applications stack, and a set of methods and recommandation to develop efficiently.

You will need jSpirit if :

- you need build a multi-tenant web application
- you need an out-of-box architectural model
- you want to focus on business code and not on technical concept
- you need respect standards
- you need a complete and integrated framework with coherent API and open-source customizable application stack

## Background

jSpirit was initially developped for a french company who wants to create a multi-tenant SaaS ERP for trading in the agribusiness world. The application is now finished and this company opens the codes of the foundation of this project. At this time, there is no foundation framework whose provide multi-tenancy support so it was a need to develop something like jSpirit. The experience of developping such application point there is a need to have tools and method to do this.

## Rationale

We think there is a strong need of architecture and simplicity in the java world. The multi-tenancy problems are difficult to resolve and the needs of such application will grow in the future. jSpirit will implements out-of-box architecure, a seamless programming model and technical module to simplify developpement. jSpirit goals is to become a concentrate of experience of open-source and advanced J2EE developpers to provide a platform for efficiently develop application in the SaaS and Multi-tenant world.

I add here more explanation about SaaS : We think SaaS is a software distribution business model for software editor. So, don't be surprised to not find specifics functionality for SaaS programming. We think we need some characteristics to achieve SaaS application, these fonctionnality are at least :

- Multi-tenancy enablement for each generic fonctionnality
- [RestFull](#) and Soap Web Services
- Easy and Rich presentation layer
- Scalable and distributed runtime
- High-availability with active-active clustering
- Simple programming model to evolve quickly
- Modular architecture to add plugins with simplicity
- Open and evolvable format for business data integration and export

We will build jSpirit to achieve this characteristics in an open-source environement for several reasons:

- We want respect standard, we know that this is not the case now with the current implementation and we need help in the community to do this, it's very important.
- We want a multi-tenant implementation independant from software editor to avoid vendor code-locking.

## Initial Goals

First goal is to develop users and developer community around the project to ensure quality and usability of the platform. Our open-source experience is not high so we think it's important to relies on a community to make the project live. Second goal is to document the project to be more usable as is. Third goal is to enlarge fonctionnality and make the project more coherent with apache ecosystem.

## Current Status

### Code Base

All the code base is here : [Sourceforge](#). The current code base implements all fonctionnalités below.

## Architecture

- Multi-tiered Architecture out-of-the-box : Implementation of Integration Layer, Business Layer, Client Layer
- Java 5 annotation and auto-injection based lookup of services
- Classpath scanning for auto-discovering components
- Modular and pluggable architecture : automatic activation of modules in the classpath, ready for seamless integration
- Implementation of Long-Conversation pattern, with JTA 2PC support (with Geronimo Transaction Manager), and implicit demarcation (explicit demarcation is always possible)
- [in progress] AOP interceptor on top of each layer

## Integration Layer

- Implementation of abstract integration services and abstract persister based on JPA technology
- Maven plugins for code generation of integration layer from xml description of component business model : generate persistent class, access services, queries, constraints, JPA annotation, lucene indexation of business model
- bean validation integration
- Full Multi-tenancy integration on [EntityManager](#) and Caches
- Multi-tenant Postgresql support

## Business Layer

- Implementation of abstract business services and infrastructure
- Annotation discovering and injection of dependents services
- Multi-tenant replacement of services at runtime
- Simple Asynchronous and distributed business services with Apache ActiveMQ : this is annotation driven

## Client Layer

- JSF 2.0 predefined integration
- Abstract Managed Bean for simple developpement of list and forms
- Integration of restful url for JSF 2
- Multi-tenant interceptor for determining tenant context based on full qualified domain name

## Scheduling

- Distributed and load adaptative voting peer-to-peer scheduler
- voting task execution with Condorcet Method

## Security

- Simple security integration : form login, http basic security
- Multi-tenant support for authentications and authorizations
- peer-to-peer sessions id replications for support max session per user in a cluster
- Regexp filters on urls
- JSF function and bean to manage security on pages

## i18n

- Full i18n support
- Multi-tenancy i18n : overriding label per tenant
- JSF function for accessing labels and locale
- JSF bean for controlling user locale on web page

## Data Import/Export

- XML data importer/exporter customizable by tenant with scripting services
- ready for "open-SaaS" to guarantee application users data integration and recuperation

## Web Services

- Simple export of business services to Soap Web Services with Apache CXF
- [in progress] REStfull web services with Apache Abdera integration (and XStream)
- Atom 1.0 support with Apache Abdera (only GET method now)

## Search

- Indexation of data model
- Simple Query interface for searching in the data model
- Multi-tenant support of the Lucene Indexes

## JCR

- Multi-tenant integration of Apache [JackRabbit](#) : workspaces based
- Implementation of injectable service for [JackRabbit](#) access
- JTA transaction participation

## Mail

- Injectable mail services out-of-box

## Reporting

- Report module on top of the business layer
- based on Castor XML and Apache FOP
- Pluggable Reporting Provider architecture
- Multi-tenant report replacement at runtime

## Tools

- Set of Maven archetype mapped on architecture to create one project by layer

## Planned Functionnality

- Maven Plugin for code generation supporting Apache Cassandra without interface modification
- More presentation module (Tapestry, ...)
- Create others multi-tenant interceptor based on other methods
- Support more algorithms for scheduling for the distributed scheduler
- Security : Services Access Authorization
- Eclipse plugins for MDA enablement, XML schema recognition, and more
- Integration of Business Rules Engine with multi-tenancy
- Integration of BPM/Workflow Engine with multi-tenancy
- Simple Cloud deployment

## Meritocracy

We think meritocracy is the best method to conduct project. It give involvement to procude quality products. The responsibility of individuals in project should grow proportionnaly to their implication. So their responsibility must represent their involvement and the quality of the work produced. In other words, we think we can improve a project quality with motivated persons who wants contribute to set the future path of the project. We think developping open-source software is an adventure and we need adventurer !!

## Community

As said above, building community is the first initial goal for the project. The actual community is actually composed of two developpers and a software company in France who uses jSPirit as a foundation for his product. We will encourage suggestions, contributions, and other feedback because we think we need stay near the users and developper community to satisfy their attents. That's why we need a free license. We wants promote exchange between persons to improve the ideas and the quality of the software.

## Core Developers

There is two developpers at the moment : Grégoire Rolland, J2EE Architect from 7 years, founder of the jSPirit project. Franck Hervy, Java developper from 8 years, contributor. We are conviced by the need of participate to free software, but our experience is minimal (but we want learn !)

## Alignment

We think apache foundation is a good match for jSPirit. jSPirit is targeted tu use Tomcat and Geronimo in the future. jSPirit uses already a lot of Apache projects and we think we need collaboration with this project to produce the best platform we can. This project are ActiveMQ, Abdera, CXF, Velocity, Maven, Commons, Geronimo, [JackRabbit](#), Lucene, FOP, ...And there are other project we will need to work with (Cassandra, OpenEJB, OpenJPA). We think there could be good interaction between projects and jSPirit could fit in the ecosystem.

## Known Risks

## Orphaned products

There is a risk the product became orphaned due to the small numbers of committers, that's why we need to build a community. Grégoire Rolland wants to build his career around this project and Frank Hervy is very interested. The company which already uses jSpirit could diminish the risk of orphaned product.

## Inexperience with Open Source

The initial committers have low experience with open source projects. All have been involved with source code that has been released under an open source license, but there is no experience developing code with an open source development process. But we are very motivated to execute our development under the meritocracy rules. As said above we are really open to this new experience, that's why we propose this project for incubation.

## Homogenous Developers

All developers work in different companies, we are in France, sure, the geographical region is limited. But we already have joint propositions from other countries. We think this project could interest a large panel of developers and we want to encourage this.

## Reliance on Salaried Developers

Frank Hervy is volunteer for developing jSpirit. Grégoire Rolland is currently unemployed but wants to make a business around this project and support this in the long term. The project will continue without salaried developers.

## Relationships with Other Apache Products

Relationship with other Apache Products exists today as dependent libraries as said above, and we want to encourage these and work together to improve the quality of each. Dependencies include ActiveMQ, Abdera, CXF, Velocity, Maven, Commons, Geronimo, [JackRabbit](#), Lucene, FOP. We want to develop the relationships with Cassandra, OpenEJB, OpenJPA and Geronimo, to build a high quality platform for developing enterprises applications. We are also very interested by the Cloud initiative of the Apache Labs.

## A Excessive Fascination with the Apache Brand

We hope the Apache Brand will attract users and contributors around this project. We know ASF since 10 years and we use products daily. We think the products are all quality and open-source respectively. We recognize the effort of the ASF in the open-source ecosystem and we want to add our sweat to this effort. However we can continue this adventure alone but this will be harder and less motivating than belong to a large and recognized community.

## Documentation

There is not much documentation now (it's the second initial goal), but you can find some materials here :

- [https://sourceforge.net/projects/jspirit/files/doc/saas\\_rml.pdf/download](https://sourceforge.net/projects/jspirit/files/doc/saas_rml.pdf/download) (in french)
- [https://sourceforge.net/projects/jspirit/files/doc/jspirit\\_presentation\\_en.pdf/download](https://sourceforge.net/projects/jspirit/files/doc/jspirit_presentation_en.pdf/download) (in english)

## Initial Source

jSpirit is in development since September 2008 by Grégoire Rolland. The development started in a French company (Husson Ingenierie, Périgueux, <http://husson-info.fr/>) when the developers needed a platform for building multi-tenant applications. The source was liberated in April 2010.

## External Dependencies

The dependencies all have Apache compatible licenses. These include LGPL, BSD, CDDL, CPL, MPL and MIT licensed dependencies.

## Cryptography

jSpirit could depend now or in the future from cryptographic code.

## Required Resources

### Mailing lists

- jspirit-private (with moderated subscriptions)
- jspirit-dev
- jspirit-commits
- jspirit-user

## Subversion Directory

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

## Issue Tracking

JIRA jSpirit (JSPIRIT)

## Initial Committers

- Grégoire Rolland (grolland dot jspirit at gmail dot com)
- Franck Hervy (hervy dot franck at free dot fr)

## Interested developpers

Here a list of the interested developper :

- Niranjan Shukla (niranjan dot shukla at gmail dot com)

## Sponsors

### Champion

We are actually looking for a Champion.

### Nominated Mentors

We also need Mentors.

### Sponsoring Entity

We gracefully ask the Incubator to be our sponsor.