CelixProposal

Abstract

Celix is a OSGi like implementation in C with a distinct focus on interoperability between Java-OSGi and Celix.

Proposal

Celix will be an implementation of the OSGi specification adapted to C. It will follow the API as close as possible, but since the OSGi specification is written primarily for Java, there will be differences (Java is OO, C is procedural). An important aspect of the implementation is interoperability between Java-OSGi and Celix. This interoperability is achieved by porting and implementing the Remote Services specification in Celix. These Services will be made available in separate bundles.

Background

In embedded/realtime situations software is mostly implemented in C. In situations where interoperability and dynamics are important, a good architecture and design principle is needed. OSGi provides such middleware for Java based systems. To be able to have such dynamic environment implemented in C, a OSGi like implementation is needed. Besides a dynamic environment, OSGi also makes it easier to reuse parts of the system, reducing time needed to implement and maintain software. The implementation started with the basics to make it possible to load libraries dynamically, and steadily grown towards an implementation where large parts of the OSGi framework API is implemented. The implementation of Celix is heavily based on Apache Felix (where appropriate it is even a direct port of the Apache Felix code (Java) to C). Since distributed systems are used more and more in services based environments, a scalable and transparent solution is needed for remote communication. The OSGi specification describes remote services, this specification will be a part of the first release. Remote services also make it possible to communicate between Java-OSGi and Celix. To achieve this interoperability, both Java and C implementations of remote services for a common protocol are needed. For local services JNI can be used, for remote services SOAP and/or REST can be used. In the latter case, Apache CXF can be used to implement the Remote Services API in Java.

Rationale

In embedded/realtime/distributed environments there is a need to be able to create dynamic and maintainable software. Currently there are no off the shelf middleware/frameworks for this. Celix tries to provide such a framework. The choice to base the implementation on the OSGi specification makes it possible for a developer to use Celix as well as Java OSGi implementation without much differences and without a steep learning curve. The community and user driven platform created by Apache provides a great base for middleware such as Celix. Also the fact that Celix is based on Apache Felix, and Apache Felix is hosted by Apache, makes it a logical choice to have Apache as home for this project.

Initial Goals

- Provide a basic implementation of the OSGi Framework API
- Provide an implementation of Remote Services to be able to create distributed systems (and Celix <-> OSGi interoperability).
- Build/Expand a community using/developing Celix
- OSGi like implementation in C
- Provide a module/component based software solution for embedded Platforms
  o Real-Time software
  o Distributed systems
  o Provide Services based solution
- Investigate if Apache Portable Runtime can be used for platform abstraction

Current Status

Meritocracy

Celix was created by Alexander Broekhuis. While he is no active committer/participant of Apache projects, he has used Open Source and is well known with it and how a meritocracy works. Currently there are several other possible committers. To be able to create and maintain complex middleware (such as Celix) a good structure is needed. A meritocracy following the rules and traditions of the ASF is a good choice for Celix.

Community

Currently the Celix community exists out of the core developers, and the users integration Celix in an end-user product (all from Thales).

Core Developers

- Alexander Broekhuis (Luminis)
- Jasper Gielen (Humiq)
- Herman ten Brugge (Thales)

Alignment

Celix is heavily based on Apache Felix. Since Apache Felix is an Apache project it makes sense to develop Celix under Apache. Also, Celix is a complex and large middleware project, it makes sense to have a supporting/developing community. Apache provides a solid base, with established processes and rules, to create such community.

Known Risks

Orphaned Products

Celix will be used by Thales, and so there is no direct risk for this project to be orphaned.

Inexperience with Open Source

The committers have experience using and/or working on open source projects. The Apache process is new, but most likely not a problem.

Homogeneous Developers

Currently all committers are from the Netherlands, but they do work for different organizations.

Reliance on Salaried Developers

All committers working on Celix (currently) are paid developers. The expectation is that those developers will also start working on the project in their spare time.

Relationships with Other Apache Products

- Celix is based on Apache Felix
- Using Apache ACE it probably is possible to provision Celix bundles
- For remote services Apache CXF can be used (either SOAP or REST)
- Possibly Apache ZooKeeper can be used for remote service discovery (Apache ZooKeeper vs SLP)
- Possibly Apache Portable Runtime for platform abstraction

An Excessive Fascination with the Apache Brand

Celix itself will hopefully have benefits from Apache, in terms of attracting a community and establishing a solid group of developers, but also the relation with Apache Felix. These are the main reasons for us to send this proposal. We think that a good community is needed to build and maintain large middleware projects, such as Celix. However, even if Celix would not be accepted, development will continue. As such, there is no need to, or reason to, "abuse" the Apache Brand.

Documentation

Currently all documentation and information is stored on a private corporate wiki.. This has to be moved to a public place. (is this part of the process after acceptance, or should this be placed on (eg) the luminis open source server?)

Initial Source

Development of Celix started in the summer of 2010. The source currently is located on a private corporate SVN repository. All the source is available for Open Sourcing and can be found at http://opensource.luminis.net/wiki/display/SITE/Celix

Source and Intellectual Property Submission Plan

Celix is currently developed solely by Luminis employees. All source will be donated to Apache.

External Dependencies

- MiniZip source , zlib license

This source can be included, according to http://www.apache.org/legal/3party.html

Required Resources

Mailing Lists

- celix-dev
- celix-private

Subversion Directory

https://svn.apache.org/repos/asf/incubator/celix

Issue Tracking

JIRA Celix

Other Resources

- CMake

Celix uses Cmake as build environment. CMake generates Make files for building, bundling and deploying Celix. This build environment can also be used by project using Celix, it provides simple methods for creating and deploying bundles to a named target.

- Confluence Wiki

To be able to provide help, documentation, faq etc, a wiki is needed.

Initial Committers

Alexander Broekhuis a.broekhuis@gmail.com

Sponsors

Champion

Marcel Offermans

Nominated Mentors

- Marcel Offermans
- Karl Pauls
- Luciano Resende (lresende AT apache DOT org)

Sponsoring Entity

Celix is a new project and proposed is to release to code under the sponsorship of the Incubator.

Status

The proposal is now ready for a vote.