# MessagingComponentsProposal

## Abstract

Commentary:

A short descriptive summary of the
project. A short paragraph, ideally one sentence in length.

The abstract should be suitable for reuse in

the board resolution used to create the official project upon graduation,
as the first paragraph on the podling web site and in the DOAP document.

Examples:

Geronimo will be a J2EE compliant container.

Heraldry will develop technologies around the emerging user-centric
identity space.

Yoko will be a CORBA server.

# Proposal

Commentary:

A lengthier description of the proposal. Should be reasonably declarative. More discursive material should be included in the rationale (or other later sections).

Example:

XAP is to provide an XML-based declarative framework for building,
deploying and maintaining rich, interactive, Ajax-powered web
applications. A basic principal of XAP is to leverage existing Ajax

### Background

Commentary:

Provides context for those unfamiliar with the problem space and history.

Explain terms whose meanings may be misunderstood (for example, where there is not a single widely adopted definition).

This content should be capable of being safely ignored by domain experts. It should probably find an eventual home on the Podling website.

Example (Heraldry):

To provide some background, the Higgins Project is being actively
developed within Eclipse and is a framework that will enable users
and enterprises to integrate identity, profile, and relationship
information across multiple systems. Using context providers,
existing and new systems such as directories, collaboration spaces
...

### Rationale

Commentary:

Explains why this project needs to exist and why should it be adopted by Apache. This is the right place for discursive material.

Example (Beehive):

There is a strong need for a cohesive, easy-to-use programming model
for building J2EE applications. Developers new to Java are forced to
learn a myriad of APIs just to build simple applications; advanced
J2EE developers are forced to write tedious plumbing code; and tools
authors are limited in what they can do to simplify the experience
due to the underlying complexity.

### Initial Goals

Commentary:
A complex proposal (involving multiple existing code bases, for example) may cause concerns about its practicality. A good way to address these concerns is to create a plan that demonstrates the proposal is feasible and has been carefully thought through.

Many projects will not need this section.

Example (Heraldry):

- Expansion of Yadis and OpenID libraries into additional

languages
beyond the existing Python, Ruby, Perl, and PHP libraries

- OpenID authentication specification revision to fix known security considerations, investigate compatibility with the

DIX IETF
proposal, describe Yadis integration, and allow either an URL or
XRI be used as the End User's Identifier

## Current Status

Commentary:
This section (and the contained topics) describes the candidate's current status and development practices. This should be an honest assessment balancing these against Apache's principles and development ideals.

For some proposals, this is a chance to demonstrate understanding of the issues that will need to addressed before graduation. For others, this is a chance to highlight the close match with Apache that already exists. Proposals without an initial code base should just simply state that.

Some proposals name this section criteria (though the term is a little misleading).

- **Meritocracy:**

Commentary:

Apache is a meritocracy.

Once a developer has submitted enough good patches then it should be natural that they are elected to committer. It should be natural that active committers are elected to the project management committee (PMC).

This process of renewal is vital to the long term health of Apache projects. This is the right place to demonstrate that this process is understood by the proposers.

Example (OFBiz):

OFBiz was originally created by David E. Jones and Andy Zeneski in
May 2001. The project now has committers and users from around the
world. The newer committers of the project joined in subsequent
years by initially submitting patches, then having commit privileges
for some of the applications, and then privileges over a larger
range of applications...

Example (Beehive):

We plan to do everything possible to encourage an environment that
supports a meritocracy. One of the lessons that the XMLBeans
committers have learned is that meritocracies don't just evolve
from good intentions; they require actively asking the community
for help, listing/specifying the work that needs to be done, and
keeping track of and encouraging members of the community who make
any contributions...

- **Community:**

Commentary:

Apache is interested only in communities.

Candidates should start with a community and have the potential to grow and renew this community by attracting new users and developers. Explain how the proposal fits this vision.

Example (Beehive):

BEA has been building a community around predecessors to this
framework for the last two years. There is currently an active
newsgroup that should help us build a new community at Apache.

Example (WebWork2):

The WebWork 2 community has a strong following with active mailing lists and forums.

Example (WADI):

The need for a full service clustering and caching component in the open source is tremendous as its use can be applied in many areas, thus providing the potential for an incredibly large community.

- **Core Developers:**

Apache is composed of individuals.

It is useful to provide a brief introduction to the developers on the initial committers list. This is best done here (and not in that section). This section may be used to discuss the diversity of the core development team.

Example (ServiceMix)
The core developers are a diverse group of developers many of which are already very experienced open source developers. There is at least one Apache Member together with a number of other existing Apache Committers along with folks from various companies.

http://servicemix.org/Team

Example (WADI)
WADI was founded by Jules Gosnell in 2004, it now has a strong base of developers from Geronimo, Castor, OpenEJB, Mojo, Jetty, ActiveCluster, ActiveMQ, and ServiceMix.

- **Alignment:**

Describe why Apache is a good match for the proposal. An opportunity to highlight links with Apache projects and development philosophy.

Example (Beehive):
The initial code base is targeted to run within Tomcat, but the goal is to allow the framework to run on any compliant Servlet or J2EE container. The Web services component, based on JSR-181, will leverage Axis. The NetUI component builds on top of Struts. The underlying Controls component framework uses Velocity. There are other projects that we will need to work with, such as the Portals and Maven projects.

**Known Risks**

An exercise in self-knowledge. Risks don't mean that a project is unacceptable. If they are recognized and noted then they can be addressed during incubation.

- **Orphaned products**:

A public commitment to future development.

Recruiting a diverse development community and strong user base takes time. Apache needs to be confident that the proposers are committed.

Example (Yoko):
The contributors are leading vendors in this space. There is no risk of any of the usual warning signs of orphaned or abandoned code.

Example (Ivy):
Due to its small number of committers, there is a risk of being orphaned. The main knowledge of the codebase is still mainly owned by Xavier Hanin. Even if Xavier has no plan to leave Ivy development, this is a problem we are aware of and know that need to be worked on so that the project become less dependent on an individual.

Example (Tika):
There are a number of projects at various stages of maturity that implement a subset of the proposed features in Tika. For many potential users the existing tools are already enough, which reduces the demand for a more generic toolkit. This can also be seen in the slow progress of this proposal over the past year.

However, once the project gets started we can quickly reach the feature level of existing tools based on seed code from sources mentioned below. After that we believe to be able to quickly grow the developer and user communities based on the benefits of a generic toolkit over custom alternatives.

- **Inexperience with Open Source:**

If the proposal is based on an existing open source project with a history of open development, then highlight this here.

If the list of initial committers contains developers with strong open source backgrounds then highlight this here.

Inexperience with open source is one reason why closed projects choose to apply for incubation. Apache has developed over the years a store of experience in this area. Successfully opening up a closed project means an investment of energy by all involved. It requires a willingness to learn and to give back to the community. If the proposal is based around a closed project and comes with very little understand of the open source space, then acknowledge this and demonstrate a willingness to learn.

Example (Cayenne):
Cayenne was started as an open source project in 2001 and has remained so for 5 years.

Example (Beehive):
Many of the committers have experience working on open source projects. Five of them have experience as committers on other Apache projects.

Example (Ivy):
While distributed under an open source license, access to Ivy was initially limited with no public access to the issue tracking system or svn repository. While things have changed since then - the svn repository is publicly accessible, a JIRA instance has been setup since june 2005, many new features are first discussed on the forum or JIRA - experience with a true open source development model is currently limited.
However, Maarten has already a good experience with true open development process, and bring his experience to the project.

Example (River):
The initial committers have varying degrees of experience with open source projects. All have been involved with source code that has been released under an open source license, but there is limited experience developing code with an open source development process. We do not, however, expect any difficulty in executing under normal meritocracy rules.

- **Homogenous Developers:**

Healthy projects need a mix of developers. Open development requires a commitment to encouraging a diverse mixture. This includes the art of working as part of a geographically scattered group in a distributed environment.

Starting with a homogenous community does not prevent a project from entering incubation. But for those projects, a commitment to creating a diverse mix of developers is useful. Those projects who already have a mix should take this chance to highlight that they do.

Example (Beehive):
The current list of committers includes developers from several different companies plus many independent volunteers. The committers are geographically distributed across the U.S., Europe, and Asia. They are experienced with working in a distributed environment.

Example (River)
Since the Jini Technology Starter Kit has been mainly developed to date by Sun Microsystems, the vast majority of initial committers to the project are from Sun. Over the years, Sun has received bug fixes and enhancements from other developers which have been incorporated into the code. Our plan is to work with these other developers and add them as committers as we progress. There are three other initial committers (non Sun): Bill Venners, Dan Creswell, and Mark Brouwer. Bill is the lead of the Service UI API work, Dan has been involved with much Jini-based development, including an implementation of the JavaSpaces service called Blitz <http://www.dancres.org/blitz/>, and Mark is veteran of much Jini-based development, including commercial work at Virgil <http://www.virgil.nl> as well as leading the open source Cheiron <http://www.cheiron.org> project.

Example (Ivy):
With only two core developers, at least they are not homogenous! Xavier and Maarten knew each other only due to their common interest in Ivy.

- **Reliance on Salaried Developers**:

A project dominated by salaried developers who are interested in the code only whilst they are employed to do so risks its long term health.

Apache is about people, not corporations. We hope that developers continue to be involved with Apache no matter who their current employer happens to be.

This is a right place to indicate the initial balance between salaried developers and volunteers. It's also good to talk about the level of commitment of the developers.

Example (OpenJPA):
Most of the developers are paid by their employer to contribute to this project, but given the anticipation from the Java community for the a JPA implementation and the committers' sense of ownership for the code, the project would continue without issue if no salaried developers contributed to the project.

Example (River):
It is expected that Jini development will occur on both salaried time and on volunteer time, after hours. While there is reliance on salaried developers

(currently from Sun, but it's expected that other company's salaried developers will also be involved), the Jini Community is very active and things should balance out fairly quickly. In the meantime, Sun will support the project in the future by dedicating 'work time' to Jini, so that there is a smooth transition.

Example (Wicket):
None of the developers rely on Wicket for consulting work, though two - Martijn and Eelco - are writing Wicket In Action (publisher Manning) in their spare time. Most of the developers use Wicket for their day jobs, some for multiple projects, and will do so for a considerable while as their companies (specifically Topicus, Cemron and Teachscape) choose Wicket as their development framework of choice.

*

**Relationships with Other Apache Products:**

Apache projects should be open to collaboration with other open source projects both within Apache and without. Candidates should be willing to reach outside their own little bubbles.

This is a an opportunity to talk about existing links. It is also the right place to talk about potential future links and plans.

Apache allows different projects to have competing or overlapping goals. However, this should mean friendly competition between codebases and cordial cooperation between communities.

It is not always obvious whether a candidate is a direct competitor to an existing project, an indirect competitor (same problem space, different ecological niche) or are just peers with some overlap. In the case of indirect competition, it is important that the abstract describes accurately the niche. Direct competitors should expect to be asked to summarize architectural differences and similarities to existing projects.

Example (OpenJPA):
Open JPA will likely be used by Geronimo, requires some Apache products (regexp, commons collections, commons lang, commons pool), and supports Apache commons logging.

Example (River)
Currently the only tie to Apache projects is the starter kit's use of the Ant build tool. There are potential future ties (http server, database backend, etc)that will be explored.

- **A Excessive Fascination with the Apache Brand:**

Concerns have been raised in the past that some projects appear to have been proposed just to generate positive publicity for the proposers. This is the right place to convince everyone that is not the case.

This is also the right place to build bridges with the community after past misdemeanors (for example, if any of those associated with the proposal have - in the past - sort to associate themselves with the Apache brand in factually incorrect ways) and promise good conduct for the future.

Example (CeltiXfire):
While we expect the Apache brand may help attract more contributors, our interests in starting this project is based on the factors mentioned in the Rationale section. However, we will be sensitive to inadvertent abuse of the Apache brand and will work with the Incubator PMC and the PRC to ensure the brand policies are respected.

Example (Wicket):
The ASF has a strong brand, and that brand is in itself attractive. However, the developers of Wicket have been quite successful on their own and could continue on that path with no problems at all. We are interested in joining the ASF in order to increase our contacts and visibility in the open source world. Furthermore, we have been enthusiastic users of Apache from the earliest hour (remember JServ anyone?), and feel honored at getting the opportunity to join the club.

Example (OpenJPA):
We think that Open JPA is something that will benefit from wide collaboration, being able to build a community of developers and committers that outlive the founders, and that will be embraced by other Apache efforts, such as the Geronimo project.

**Documentation**

References to further reading material.

Examples (Heraldry):
[1] Information on Yadis can be found at:
http://yadis.org

http://www.openidenabled.com

[2] Information on OpenID can be found at:
http://www.openid.net
http://www.openidenabled.com

The mailing list for both OpenID and Yadis is located at:
http://lists.danga.com/mailman/listinfo/yadis
...

**Initial Source**

Describes the origin of the proposed code base. If the initial code arrives from more than one source, this is the right place to outline the different histories.

If there is no initial source, note that here.

Example (Heraldry):
OpenID has been in development since the summer of 2005. It currently has an active community (over 15 million enabled accounts) and libraries in a variety of languages. Additionally it is supported by LiveJournal.com and is continuing to gain traction in the Open Source Community.

Yadis has been in development since late 2005 and the specification has not changed since early 2006. Like OpenID, it has libraries in various languages and there is a large overlap between the two communities. The specification is...

**Source and Intellectual Property Submission Plan**

Complex proposals (typically involving multiple code bases) may find it useful to draw up an initial plan for the submission of the code here. Demonstrate that the proposal is practical.

Example (Heraldry):

- The OpenID

specification and content on openid.net from Brad Fitzpatrick of Six Apart, Ltd. and David Recordon of VeriSign, Inc.

- The domains openid.net and yadis.org from Brad Fitzpatrick of

Six Apart, Ltd. and Johannes Ernst of NetMesh, Inc.

- OpenID libraries in Python, Ruby, Perl, PHP, and C# from JanRain, Inc.
  ...
- Yadis conformance test suite from NetMesh and

VeriSign, Inc.

We will also be soliciting contributions of further plugins and patches to various pieces of Open Source software.

**External Dependencies**:

External dependencies for the initial source is important. Only some external dependencies are allowed by Apache policy. These restrictions are (to some extent) initially relaxed for projects under incubation.

If the initial source has dependencies which would prevent graduation then this is the right place to indicate how these issues will be resolved.

Example (CeltiXfire):
The dependencies all have Apache compatible licenses. These include BSD, CDDL, CPL, MPL and MIT licensed dependencies.

**Cryptography**

If the proposal involves cryptographic code either directly or indirectly, Apache needs to know so that the relevant paperwork can be obtained.

**Required Resources**:

- **Mailing lists:**
  The minimum required lists are

private@{podling}.incubator.apache.org (for confidential PPMC discussions) and dev@{podling}.incubator.apache.org lists. Note that projects historically misnamed the private list pmc. To avoid confusion over appropriate usage it was resolved that all such lists be renamed.

If this project is new to open source, then starting with these minimum lists is the best approach. The initial focus needs to be on recruiting new developers. Early adopters are potential developers. As momentum is gained, the community may decide to create commit and user lists as they become necessary.

Existing open source projects moving to Apache will probably want to adopt the same mailing list set up here as they have already. However, there is no necessity that all mailing lists be created during bootstrapping. New mailing lists can be added by a VOTE on the Podling list.

By default, commits for {podling} will be emailed to commits@{podling}.incubator.apache.org. It is therefore recommended that this naming convention is adopted.

Mailing list options are described at greater length elsewhere.

Example (Beehive):

- private@beehive.incubator.apache.org (with

moderated subscriptions)

- dev@beehive.incubator.apache.org
- commits@beehive.incubator.apache.org

- **Subversion Directory:**
  It is conventional to use all lower case,

dash-separated ⛔ directory names. The directory should be within the incubator directory space (http://svn.apache.org/repos/asf/incubator).

Example (OpenJPA):

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

- **Git Repositories:**
  It is conventional to use all lower case, dash-separated ⛔ repository names. The repository should be

prefixed with incubator and later renamed assuming the project is promoted to a TLP.

Example (Blur):

https://git-wip-us.apache.org/repos/asf/incubator-blur.git

- **Issue Tracking:**
  Apache runs JIRA and Bugzilla. Choose one. Indicate the name by which project should be known in the

issue tracking system.

Example (OpenJPA):
JIRA Open-JPA (OPEN-JPA)

- **Other Resources:**
  Describe here any other special infrastructure requirements necessary for the

proposal. Note that the infrastructure team usually requires a compelling argument before new services are allowed on core hardware. Most proposals should not require this section.

Most standard resources not covered above (such as continuous integration) should be added after bootstrapping. The infrastructure documentation explains the process.

**Initial Committers**

List of committers (stating name and an email address) used to bootstrap the community. Mark each which has submitted a contributor license agreement (CLA). Existing committers should use their apache.org email address (since they require only appropriate karma). Others should use the email address that is (or will be) on the CLA. That makes it easy to match CLAs with proposed committers to the project.

It is a good idea to submit CLAs at the same time as the proposal. Nothing is lost by having a CLA on file at Apache but processing may take some time.

Note this and this. Consider creating a separate section where interested developers can express an interest (and possibly leave a brief introduction) or ask them to post to the general list.

Example (OpenJPA):
Abe White (awhite at bea dot com)
Marc Prud'hommeaux (mprudhom at bea dot com)
Patrick Linskey (plinskey at bea dot com)
...
Geir Magnusson Jr (geirm at apache dot org) *
Craig Russell (clr at apache dot org) *

**Sponsors**

Little bit of a controversial subject. Committers at Apache are individuals and work here on their own behalf. They are judged on their merits not their affiliations. However, in the spirit of full disclosure, it is useful for any current affiliations which may effect the perceived independence of the initial committers to be listed openly at the start.

For example, those in salaried positions whose job is to work on the project should list their affiliation. Having this list helps to judge how much diversity exists in the initial list and so how much work there is to do.

This is best done in a separate section away from the committers list.

Only the affiliations of committers on the initial bootstrap list are relevant. These committers have not been added by the usual meritocratic process. It is strongly recommended that the once a project is bootstrapped, developers are judged by their contributions and not by their background. This list should not be maintained after the bootstrap has been completed.

- **Champion:**
  The Champion is a person already associated with Apache who leads the proposal

process. It is common - but not necessary - for the Champion to also be proposed as a Mentor.

A Champion should be found while the proposal is still being formulated. Their role is to help formulate the proposal and work with you to resolve comments and questions put forth by the IPMC while reviewing the proposal.

- **Nominated Mentors:**

Lists eligible (and willing) individuals nominated as Mentors [definition] for the candidate.

Three Mentors gives a quorum and allows a Podling more autonomy from the Incubator PMC, so the current consensus is that three Mentors is a good number. Any experienced Apache community member can provide informal mentorship anyway, what's important is to make sure the podling has enough regularly available mentors to progress smoothly. There is no restriction on the number of mentors, formal or informal, a Podling may have.

- **Sponsoring Entity**:
  The Sponsor is the organizational unit within

Apache taking responsibility for this proposal. The sponsoring entity can be:

- the Apache Board
- the Incubator
- another Apache project
  The PMC for the appropriate project will decide whether to sponsor (by a vote). Unless there are strong links to an existing Apache project, it is recommended that the proposal asks that the Incubator for sponsorship.

Note that the final destination within the Apache organizational structure will be decided upon graduation.