

The Apache Software Foundation

<http://www.apache.org/>

State of Apache

Roy T. Fielding

Chairman, The Apache Software Foundation



Chief Scientist, eBuilt, Inc.

www.eBuilt.com

Mission



- The Apache Software Foundation exists to provide organizational, legal, and financial support for the Apache open-source software projects.
- A membership-based, not-for-profit corporation
 - to ensure that the Apache projects continue to exist beyond the participation of individual volunteers
 - to enable contributions of intellectual property and funds on a sound basis
 - to provide a vehicle for limiting legal exposure while participating in open-source software projects

Stefano Mazzocchi



"Apache is about open collaboration.

It is a place where you can emerge by your own merits, where you can make yourself heard, where you can change the world with your ideas, and feel proud and respected in doing so"

The Apache Vision

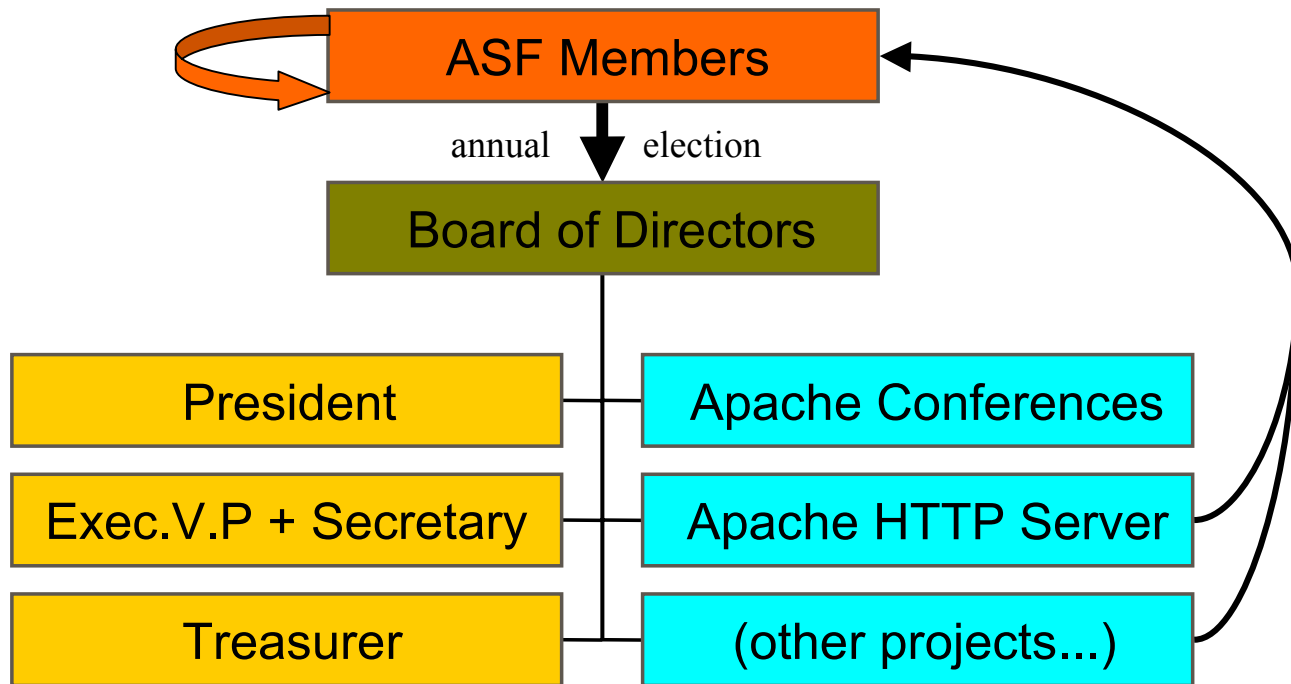


- Real innovation occurs when a contributor is able to implement their own ideas, at their own pace, and for their own purposes
- Effective long-term development and maintenance of a free, commercial-grade, open source product requires an organization that will outlive any single contributor or volunteer
- Community-based development fosters collaboration towards a common goal by supporting the work of individual innovators

The Apache Software Foundation

- When a community is not enough
 - success attracts both good and bad attention
 - liability questions hamper new efforts
 - tax questions prevent non-code contributions
- A nonprofit, membership-based corporation
 - the result of two years careful consideration and planning by the Apache Group membership
 - incorporated in March 1999, effective June 1st
 - like a moose lodge, but without the funny hats
 - www.apache.org/foundation/

ASF Corporate Structure



ASF Members

Doug MacEachern
Stefano Mazzocchi
Aram Mirzadeh

Thies C. Arntzen

Stig Bakken

Brian Behlendorf

Stas Bekman

Ryan Bloom

Eric Cholet

Ken Coar

Mark Cox

James Duncan Davidson

Lars Eilebrecht

Ralf S. Engelschall

Roy T. Fielding

Tony Finch

Pier Paolo Fumagalli

Dean Gaudet

Dirk-Willem van Gulik

Andi Gutmans

Ask Bjørn Hansen

Brian Havard

Ben Hyde

Jim Jagielski

Manoj Kasichainula

Ed Korthof

Alexei Kosut

Martin Kraemer

Ben Laurie

Rasmus Lerdorf

Daniel Lopez Ridruejo

Chuck Murcko

Sameer Parekh

David Reid

Sam Ruby

Wilfredo Sanchez

Sascha Schumann

Cliff Skolnick

Marc Slemko

Greg Stein

Jon S. Stevens

Bill Stoddard

Zeev Suraski

Paul Sutton

Randy Terbush

Jim Winstead

Projects



- Web Server (httpd)
- Jakarta (nee Java-Apache)
 - Tomcat, Ant, Regexp, Slide, Struts, Taglibs, ...
 - Jetspeed, Turbine, Alexandria, Jyve, Avalon, ...
- XML/Apache
 - Xerces, Xalan, Cocoon, FOP, Xang, SOAP
- Perl/Apache Integration
 - mod_perl, Apache::
- Conferences
- Foundation

Apache HTTP Server Project



- A common goal
 - To maintain an open source, secure, efficient and extensible server that provides HTTP services in sync with non-proprietary World Wide Web standards
- Apache Group (HTTP Server PMC)
 - Self-selected volunteers that guide the project and perform most of the development work
- Current status
 - #1 server
 - *Over 10 million web sites*
 - *65% of the public Internet sites*
 - 31 Apache Group members

Open Source Incubation

- httpd is the prototype project
 - but how much of httpd development is universal?
 - we only support **collaborative** development
- Java-Apache, Jakarta, XML, Perl, and TCL
 - incubator projects
 - give participants time to get used to Apache rules
 - need to spin-off projects when they are mature
- Tomcat, Cocoon, Jetspeed
 - active projects with broad base of developers
 - it is now time to spin them off

Roy's ASF License Philosophy

- We want our software to be usable and modifiable by the largest number of people for any purpose they see fit
 - Open source derives benefit by trusting community
 - Any forced contribution is a worthless contribution
 - This implies a license with very few restrictions on redistribution and use
 - ASF license is based on earlier BSD license
- “A license can ruin a perfectly good piece of software.” – Jon Stevens

What we won't redistribute



- Code with viral conditions
 - stickiness of their license must not cause our code to become sticky, nor can it place additional constraints on what can be combined with our code
- Code that excessively limits modifications
 - everything must be maintainable by us and our users
 - “derived work” must be based on source code
 - license grants on modified work must be limited to what was actually modified by the contributor
 - modifications do not imply public distribution

ASF Conferences



- ApacheCon Europe 2000
 - London, England
 - October 23-25, 2000
 - www.apachecon.com

- ApacheCon 2001
 - Santa Clara Convention Center, California
 - April 4-6, 2001

Apache Futures



- The future is what you make of it
 - all Apache work is based on someone needing something at some point and just doing it
 - all Apache technical decisions are discussed and implemented on public mailing lists
- Open Source == Shared Custom Software
 - community is more important than code
 - but the code is needed to form a community
- We'll distribute the future when it's ready

Questions for the Developers?



Why call it Apache?

- No, it isn't an attack helicopter
 - unless you happen to work for Microsoft
- A Patchy Server? (the outside joke)
- A reference to our development philosophy:
“Characteristic of both Eastern and Western Apache, with the exception of the Kiowa Apache, was the lack of a centralized tribal organization. The band, an autonomous collection of small local groups within a given locality, was the primary political unit as well as the primary warring and raiding unit. The strongest headman of the local groups was recognized as an informal chief, and several bands might be united under one leader. Chieftainship was thus not generally hereditary.” *Encyclopaedia Britannica*

Collaboration Techniques

- Collaborative development requires
 - at least one common goal
 - | *but not all goals need to be common*
 - a means for communication
 - | *both public and private*
 - a shared information space
 - | *access to past communication (organizational memory)*
 - | *access to past and current products*
 - coordination
 - | *to make all of the above possible*
 - | *requires an architecture with good separation of concerns*

Development Process



- Trusted “core” developers
 - write access to the CVS repository (shared space)
 - work on what they want, when they want
 - release pressure expands what they want to include the common goal of a complete distribution
- Development issues and conflicts
 - choosing among features and alternative fixes
 - avoiding server bloat
 - setting project direction
- Voting resolves conflicts

Conflict Resolution via Voting

- We needed a way for the group to make a decision without requiring the whole group to be involved in making every decision
- Small quorum consensus
 - votes: +1 = yes, 0 = *shrug*, -1 = no/veto
 - three +1 and no veto required for approval
 - lazy consensus when consensus is likely
- Emphasizes code review
- Attempts to minimize interruptions in the flow