ApacheCon

Behind the Scenes of The Apache Software Foundation

Lars Eilebrecht <lars@apache.org>
Cliff Schmidt <cliffs@apache.org>
Stefano Mazzocchi <stefano@apache.org>

US 2006

What is the ASF?

- ASF stands for Apache Software Foundation
- The ASF is a registered non-profit organization
- Incorporated in the state of Delaware, USA
- The ASF is a volunteer organization

What is the ASF?

- all work is done by volunteers
- nobody gets paid by the foundation
- except a contractor working full-time as system administrator (since April 2006)

Why the ASF?

- to provide a foundation for open, collaborative software development projects by supplying hardware, communication and business infrastructure
- to provide a means for individual volunteers to be sheltered from lawsuits directed at the foundation's projects

Why the ASF?

- to create an independent legal entity to which companies and individuals can donate resources and be assured that those resources will be used for the public benefit
- to protect the »Apache« brand, as applied to its software products, from being abused by other organizations

Bylaws of the ASF

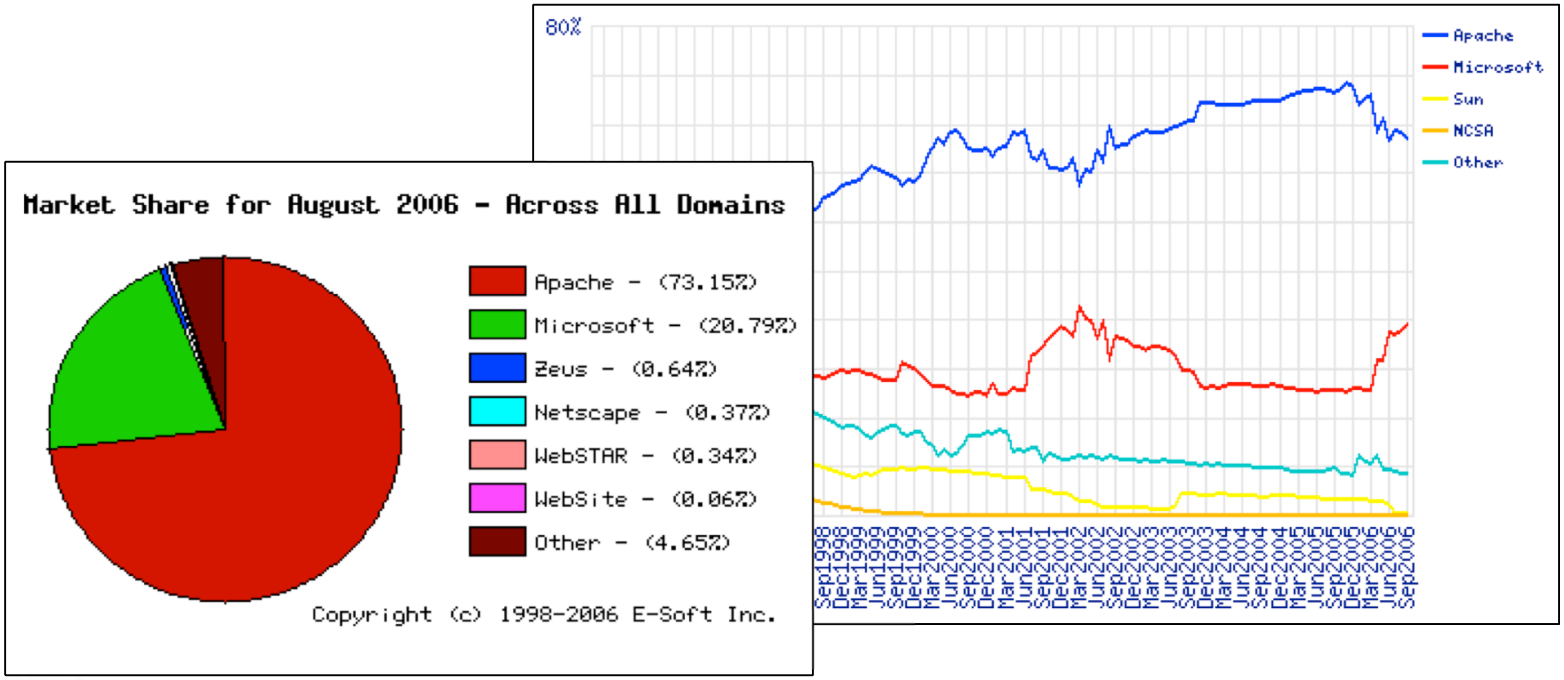
- bylaws can only be changed by the members or board of directory, but not just by the president
- annual vote of a new board of directors
- board members do not need to be residents of the USA
- becoming a new member requires a majority vote of all existing members

Before the ASF

- in 1995 a group of people got together to exchange software patches for the NCSA web server
- they started to call themselves the Apache Group
- later that year they released their own web server: the Apache HTTP Server

Before the ASF

- in less than 6 months (April 1996) they surpassed the NCSA web server adoption
- market share August/September 2006: 63-73%



Before the ASF

- thanks to its modular design, sister projects for web server modules came to gravitate around the Apache Group
- being leader of the market, the Apache Group started to fear abuse and lack of legal protection
- all this lead to the creation of the Apache Software Foundation in 1999

Why »Apache« as a Name?

- no, it isn’t an attack helicopter ...
- »a patchy« server? No, that's just a pun
- reference to the ASF's development philosophy:

  »Characteristic of both Eastern and Western 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*)

Meritocracy

- govern of merit
- the more merit you have, the more power you get
- no old-men club because of openness

The Chain of Merit

- User
- Committer
- Project Member
- ASF Member

User

- anybody who uses the software
  - passive: does not contribute directly to the project (aka lurker)
  - active: contributes directly to the project (aka contributor), either with code patches, feature requests, bug reports, answering questions on mailing lists, etc. ...

Committer

- an active user that was elected for merit and, for this, receives:
  - write access to the project's source code repositories (SVN)
  - an »apache.org« email address and access to certain ASF server infrastructure

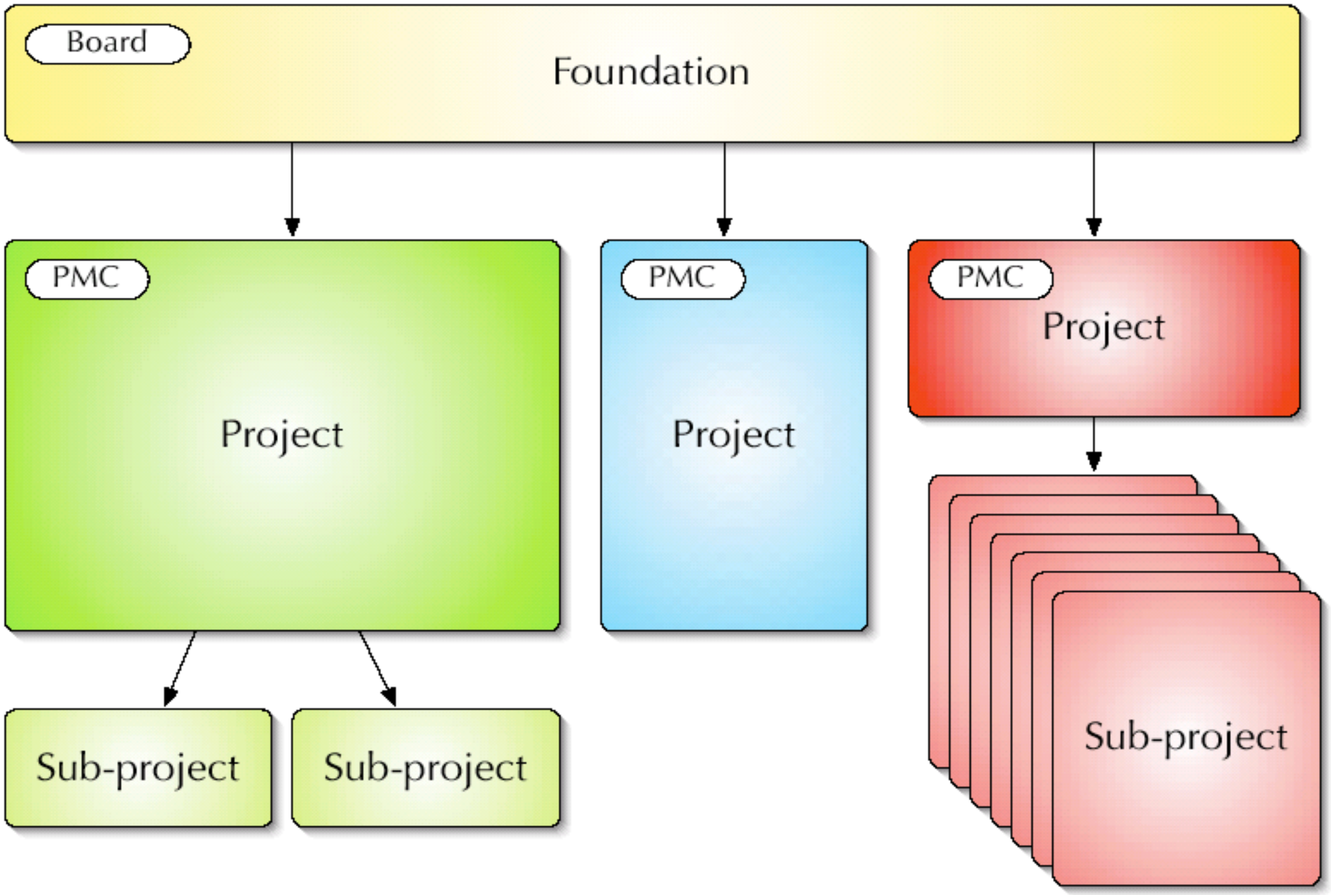
Project Member

- an active committer that was elected for merit in the evolution of the project, for this, receives:
  - the ability to vote
  - the ability to propose other people for committership

ASF Member

- a project member or committer that was elected for merit in the evolution of the foundation
  - shareholder of the foundation
  - can work on foundation projects/teams
  - can propose committers for membership
  - can elect the board
  - can be a candidate for the board election
  - can propose a new project for incubation

The Foundation Structure

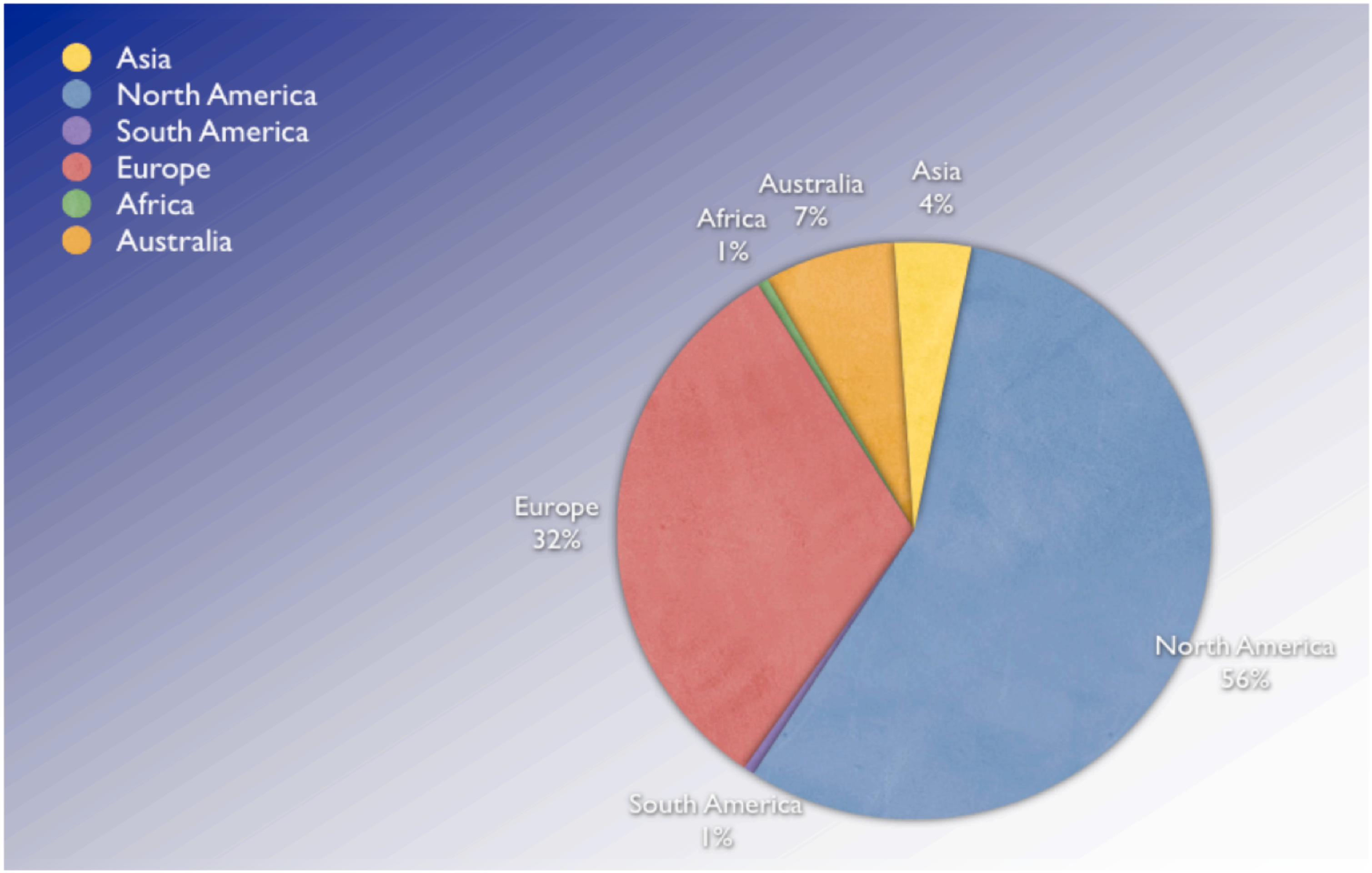


Some Foundation Numbers

- 1239 Committers
- 202 Members
- 42 Officers and 9 Directors

(June 2006)

ASF Member Demographics



The Board of Directors

- manages corporate assets (funds, intellectual property, trademarks, and support equipment)
- allocates corporate resources to the various projects
- 9 members elected every year
- makes no technical decisions!

Project Management Committee

- the PMC manages a project (and its sub-projects) subject to the direction of the board
- each PMC has the faculty of establishing its own set of rules and procedures for day-to-day project management
  ... but many share the same

Communication

- communication is done mainly over asynchronous media (mostly mailing lists)
- all communication is kept archived, so that others can join or have a reference
- this allows geographic distribution
- if people do F2F meetings they are required to post information of what has been discussed to the list.

Decision Making

- lazy consensus: making a decision does not require the whole group to be involved
- voting rules:

yes	abstain	no (veto)
+1	0	-1

- three +1 and no veto required for approval
- a veto requires proposal of an alternative solution or explanation of reason

Contributions

- code contributions
  - contribute code to software projects
- non-code contributions
  - contribute to foundation/auxiliary projects
  - contribute to software projects: documentation, project/release management, community management, etc.

Software Projects

- main purpose is to create, maintain and distribute software
- Current top-level projects:
  - Ant, APR, Beehive, Cocoon, DB, Directory, Excalibur, Forrest, Geronimo, Gump, HTTP Server, iBATIS, Jackrabbit, Jakarta, James, Lenya, Logging, Lucene, Maven, MyFaces, Apache-Perl, Portals, Shale, SpamAssassin, Struts, Tapestry, Apache-TCL, Tomcat, Web Services, Xerces, Apache-XML, XMLBeans, XML Graphics

Auxiliary Projects

- they do not distribute software, but help the foundation in specific domains
  - Incubator
  - Infrastructure
  - Security
  - Public Relations
  - Conference Planning
  - Legal
  - Java Community Process

Apache Incubator Project

- What is the Apache Incubator project?
- Should your project come to Apache?
- How does one propose a project?
- What happens during incubation?
- How do projects graduate?

Apache Incubator Project

- formed in October 2002
- entry point for all new software projects
- provides guidance for new committers
- focus on:
  - addressing legal issues with contributed IP
  - building healthy community around the project
- incubating projects are not officially endorsed

Incubator Disclaimer

"Apache xxxx is an effort undergoing incubation at the Apache Software Foundation (ASF), sponsored by the Incubator project. Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects. While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF."

Warning Signs

- Do you plan to use a hand-crafted license for your code?
- Are you now wondering ‘what code?’
- Do you have any inclination to give away your employer’s IP without authorization?
- Do you view community participation as annoying overhead?
- Do you think open source is where projects go to die?
- Are you often accused of being a “control freak”?
- Is your end-game to stamp “Apache” labels everywhere you can to get more press?

Before Proposing

- Can you license your code under Apache License?
- Do you have a code base?
- Can you sign the Contributor License Agreement?
- Do you already have a community?
- Are you committed to the code and community?
  - not just development, but community management
- Are you okay with not having ultimate control of the development of your code base?
- Are you prepared to follow Apache's marketing guidelines?

Pre-Proposal Guidelines

- explore potentially related communities
  - consider posting to related dev lists for feedback
  - find ASF member <u>champion</u> and <u>mentor</u>
    - (three mentors is helpful, but one dedicated one is critical)
  - find a potential host in a Top Level Project
    - or propose project as TLP itself
- gather the project's initial committers
  - should represent multiple interests

Pre-Proposal Guidelines

- clean up your code

- consider licensing issues
  - license your code under an open source license
  - ensure you have the rights to do this first
  - consider the licensing of any upstream dependencies
  - consider the impact on downstream dependencies

Proposal: Using the Template

- rationale / description
- criteria
  - meritocracy, community, developers, alignment
- warning signs:
  - orphaned, inexperience with OSS,
  - homogeneous, reliance on salaried developers
  - no ties to other products
- scope of project
- initial source
- resources requested
- initial committers: history and number
- sponsor

Proposal: Posting (option A)

- post proposal on Wiki
- send email with rev # or inline copy
  - to: general@incubator.apache.org
    subj: [PROPOSAL] {projectname}
    cc: {potentially sponsoring project}
- join in ~3-30 days of discussion
  - clarify proposal
  - look for interest in community
    - Potential contributors / initial committers
    - Future projects to work with
    - Project mentor

Proposal: Posting (option B)

- proposal subproject of an existing TLP:
  - propose the project directly to the TLP
    - to the developer / user community, or
    - privately to the TLP's Project Management Committee
  - if voted and accepted by the TLP,
    - work with the sponsoring TLP and the Incubator PMC to get your project started.
- public decision making is always preferable,
  - some may be concerned with keeping proposal discussions out of the press before final

Post-Proposal

- vote (when discussion winds down):
  - sponsoring PMC members cast binding votes
  - all members of community encouraged to vote opinion
  - votes run about 3 days
  - acceptance: positive consensus with no binding vetos
- get started!
  - software grant and CLAs signed and faxed in
  - accounts, mailing lists, Bugzilla/Jira, SVN, Wiki
  - confirm your community is ready to move development
  - http://www.apache.org/dev/project-creation-form.txt
  - import code into Apache SVN

Life in the Incubator

- learn
  - subscribe to general@apache!
- legal
  - all code written by committers is under Apache License
  - watch third-party code licenses and crypto export
- logistics
  - lots of docs out there to help
  - ask for help: mentor, incubator, sponsoring project
- community management
  - self-sustaining users for support & requirements
  - scalable and diverse developer community

The PPMC

- Podling Project Management Committee
- Under the supervision of the Incubator PMC
- Helps new folks understand the role of a PMC
- Mentors are part of the PPMC
- Often includes all committers
  - some projects have made PPMC membership a separate level of merit
- Releases & Graduation still require PMC vote

Restrictions on Podlings

- Releases
  - requires Incubator PMC vote on general@
  - include disclaimer text (see earlier slide)
  - file name includes ‘incubating’
  - some hesitancy over 1.0 versions
- Press Releases
  - name
    - project still referred to as “Apache {name}”
    - reference incubation (URL or “undergoing incubation”)
  - content
    - not about acceptance into incubation
    - ASF will not issue press releases, but reviews others

Graduating from the Incubator

- When the community/mentors think it's ready:
- ensure status file is updated
  - legal and community in good shape?
- call for vote of the PMC on the incubator list
- ensure destination is agreed upon
- some factors: (IMHO, in decreasing importance)
  - demonstration of active, open, transparent, and meritocratic community
  - demonstration of familiarity with Apache processes (release, legal, logistics)
  - active user community
  - diversity among committers (still very important)

Life after Graduation

- logistics
  - move project site URL / remove incubator branding
  - move mailing lists, source control, etc.
- subproject issues
  - who joins sponsoring project PMC
- TLP issues
  - determine PMC members; vote for chair
  - vote on proposed resolution for Board
  - consider crafting a charter for new project
- give back!
  - pass your wisdom on!

Key Points of Apache Legal Docs

- Apache License v2.0
- Contributor License Agreement
  - individual and corporate
- Software Grant

Apache License v2.0: 1. Definitions

- Read/refer to them
  - e.g “Derivative Works”, “Contributor”
- Definitions
  - *Contribution* = “…any work of authorship, including…additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner…”
  - *Submitted* = “…any form of electronic, verbal, or written communication sent to the Licensor…including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems…but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution.”

Apache License v2.0: 2. Grant of Copyright License

- Perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable license to all copyright rights
- May sub-license
- Applies to both Source and Object form

Apache License v2.0:
3. Grant of Patent License

- Perpetual, worldwide, non-exclusive, royalty-free, irrevocable license to all patent rights
- May transfer the Work
- “those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted.”
- Patent termination upon instituting patent litigation over Work against any entity.

Apache License v2.0:
4. Redistribution

- Applies to:
  - source/object, with/without modification
- Conditions:
  - give copy of Apache License
  - prominent notice of change for each modified file
  - source must retain all IP notices
  - include all relevant attribution NOTICEs
- Handling original additions/modifications:
  - may add your copyright notice to your modifications
  - offer under other licenses

Apache License v2.0:
5. Submission of Contributions

- Contribution is licensed under Apache License
  - ..unless separate agreement executed

Apache License v2.0:
6. Trademarks

- No license to trademarks other than to:
  - describing origin of work
  - reproducing content in NOTICE file

Apache License v2.0: 7. Disclaimer of Warranty

- Unless required by law, Work on a “AS IS” basis, including:
  - TITLE,
  - NON-INFRINGEMENT,
  - MERCHANTABILITY,
  - FITNESS FOR A PARTICULAR PURPOSE

Apache License v2.0:
8. Limitation of Liability

- Unless required by law, Contributors are not liable for damages arising as a result of using or not able to use the Work

Apache License v2.0: NOTICE and LICENSE files

- NOTICE file
  - notifies user of required attributions
  - does not add additional terms to the license
  - primarily used for “About” boxes and start-up messages
- LICENSE file
  - copy of Apache License v2.0
  - copy of any additional licenses for included non-Apache software or link to additional licenses within distribution

Contributor License Agreement

- It’s a license – not copyright assignment!
  - “You reserve all right…in and to your Contributions.”
  - applies to all present and future contributions
- Grants copyright and patent license to Foundation and recipients
  - same rights granted to users in the Apache License, including the patent termination clause
- Legally entitled to grant license (see cCLA)
  - submission must disclose any known restrictions (e.g. patent/trademark licenses) – update if turns inaccurate
- No warranty being made

Corporate Contributor License Agreement

- Allows employer to license directly with ASF,
  - rather than have developers grant rights on behalf of their corporation
- Authorizes employees to contribute employer’s IP
- Employee-committers still sign individual CLAs
  - how would Apache have license to a contribution that was actually the employee’s own IP?

“Software Grant” License

- Similar copyright license as CLA
- Similar patent license as CLA
- Licensor legally able to grant license and will notify Apache of any inaccuracies
- Disclaimer of warranty and limitation of liability

Auxiliary Projects

- they do not distribute software, but help the foundation in specific domains
  - Incubator
  - Infrastructure
  - Security
  - Public Relations
  - Conference Planning
  - Legal
  - Java Community Process

The Apache Security Team

- security@apache.org
- the central point of contact regarding potential security vulnerabilities in Apache Software
- manages vulnerability reports and coordinates efforts between projects
- reports are forwarded and handled by project-specific security teams (e.g., HTTP Server and Tomcat)

Apache Public Relations Team

- press@apache.org / fundraising@apache.org
- deals with public relations, trademark licensing, and fund-raising for the foundation
- common forms of donation:
  - money (check or via paypal.com), tax-deductable in the USA
  - buying ASF gear, e.g., T-Shirts
  - bandwidth and hardware

Apache Conference Committee

- concom@apache.org
  feedback@apachecon.com
- organizes ApacheCon and similar events
- Upcoming Apache Conferences:
  - ApacheCon Europe 2007
    April 30 – Mai 4
  - ApacheCon US 2007
    November

The Java Community Process

- represents the ASF in the Java Community Process, responsible for the maintenance and evolution of the Java Platform

ASF Infrastructure

- the ASF is a virtual organization: no offices or building
- only physical existence is the technical infrastructure that runs its services
  - web services
  - source code repositories
  - mail services
  - issue/bug tracking services
  - distribution mirroring

The Infrastructure Team

- manages the ASF infrastructure 24/7
- 16 servers located at 4 co-locations (mainly San Francisco and Amsterdam)
- ASF has contracted one full-time system administrator (since April 2006)

Some Infrastructure Numbers

- Web servers:
  - about 50 requests per second
  - about 4 million hits per day
  - about 5 TByte per month

(average numbers)

ASF Rack,
UnitedLayer Co-Location

Some Infrastructure Numbers

- Mail servers:
  - handle user accounts and more than 280 public lists with a total of about 100.000 subscribers
  - about 1 Mio connects per day (inbound)
  - about 300.000 incoming messages per day
    - Rejected (RBLs): about 700.000 per day
    - Filtered (spam): about 20.000 per day
    - Filtered (virus): about 7.000 per day

More Infrastructure Numbers

- Source Code Repository: 70 GB
- Binary Repository
  - Current 8.5 GB
  - Archive 30 GB
- Mirror servers:
  - currently 282 mirrors in 55 countries
  - about 450 GByte per month of rsync traffic
  - about 2 Mio file transfers per month

Web Sites and Mailing Lists

- Public- and Foundation-wide Mailing Lists:
  - announce@apache.org
  - community@apache.org
  - announce@apachecon.com
  - general@incubator.apache.org
  - legal-discuss@apache.org
    (available to committers, others read-only)
- Web Site Resources:
  - http://www.apache.org/foundation/
  - http://www.apache.org/licenses/
  - http://incubator.apache.org

The Future of the ASF

- 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
  - all you have to do to start is subscribe and participate

»*The best way to predict the future is to invent it.*«
(Dr. Alan Kay)

Any Questions?

Cliff Schmidt <cliffs@apache.org>
Lars Eilebrecht <lars@apache.org>