The Joy of Human Interaction Over the Internet

or

Developing and integrating free software in a large project for fun and profit

Lars Wirzenius
`liw@iki.fi`

The boring contents slide

So what is this Debian thingy?

- A collection of free software packages, integrated to work as a whole, plus tools to build packages, install and manage the system, and to manage the development and release processes
  - At first Linux kernel only, now also Hurd, FreeBSD
  - Many hardware architectures: i386, Alpha, ARM, IA-64, Motorola 68k, MIPS, PA-RISC, PowerPC, Sparc (and UltraSparc), IBM S/390 and Hitachi SuperH
- Also the project and the people
  - Perhaps especially them

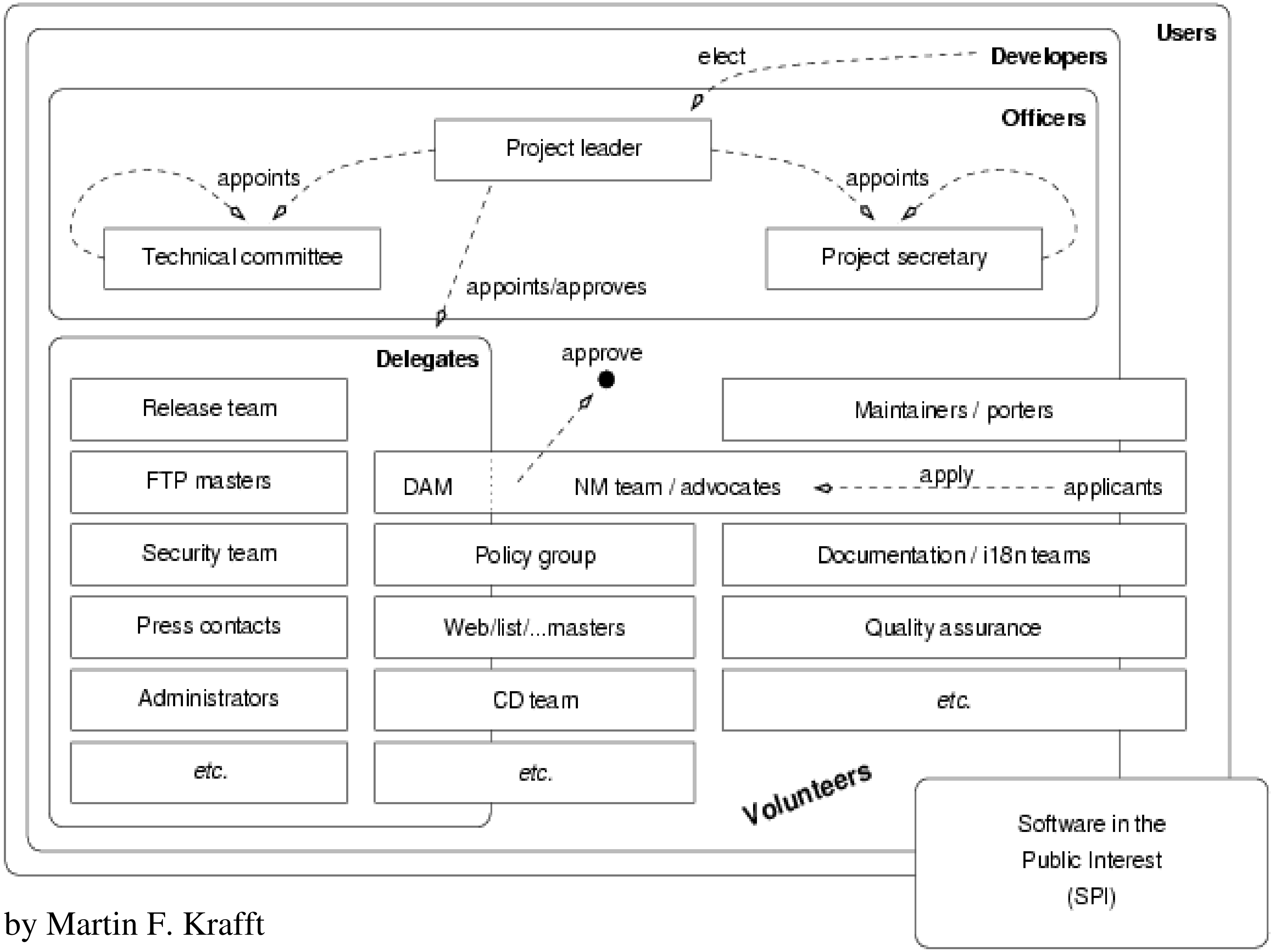
Debian history: 11 years on one slide

- 1993: Linux distribution a new concept
  - boot/root floppies, hex editing, serious hackers only
  - SLS/Slackware was dominant, based on floppies, much room for improvement, fairly closed development
  - Ian Murdock started Debian, others joined
  - actively open development process!
- 2004: More than 15000 packages, about 1000 developers (some inactive)

Debian really likes freedom

- Freedom is important
  - No freedom, no Debian, no fun, no profit
- Debian Social Contract
  - Debian will remain 100% free
  - We will give back to the free software community
  - We will not hide problems
  - Our priorities are our users and free software
  - Works that do not meet our free software standards

Debian organization: the messy graph



Graph by Martin F. Krafft

Debian organization: cats appear later

- Mostly self-organizing volunteers who form teams and groups ad hoc
- The Constitution specifies a formal decision making process for when ad hoc fails
  - Also a few official positions
  - Just enough to keep things running smoothly
- Works for Debian, might not for other projects

Decision, decisions...

- Mostly people work independently alone or in small teams
  - work on one package usually independent of most other packages
- Common issues need general discussion
  - ideology, unclear license issues, project management, integration issues, etc
  - free-form discussions mainly over e-mail (non-real-time, archives); sometimes quite *vigorous* discussions

Debian Policy: this is important

- The real technical foundation of Debian
- Written down decisions of how packages should be built so that the whole is an integrated system
  - where is the mail spool? where are manual pages? which system accounts are there?

Life as a Debian developer

- New maintainer process
  - fairly strict checking of new maintainers
  - quality, security issues
- Build package, upload package, get bug reports, fix bugs, build package, upload package, ...
- Maybe follow and join in on discussions
  - debian-devel, debian-policy, debian-project, ...
- Maybe help users
  - debian-users

What's the point, for a developer?

- Building things is fun
  - Engineering students should know this already...
- Working together with smart people is fun
- The product is useful

The Debian release process

- Three parallel versions of Debian
  - stable: the currently released version
  - testing: supposedly "ready to be released"
  - unstable: where new package uploads go
- Packages flow unstable -> testing -> stable
- Release manager decides when testing is ready to become stable
  - issues: bug counts, installer, security support, mirrors

The Debian release process sucks

- Lots of time between releases
- Getting volunteers to do things is like herding cats, so keeping schedules is hard
- A thousand developers => at least a thousand opinions on where the project should be heading
- Still, we do manage to make releases

Significant Debian lessons (1)

- Make sure things scale up
  - small problems become big problems
  - make things independent, doable in parallel
- Make sure the foundation is good
  - easier to build new stuff if old stuff works well
- Document important things
  - Debian Policy!

Significant Debian lessons (2)

- Automate repetitive tasks when possible
  - all people are lazy and often make mistakes
- Avoid single points of failure, especially for volunteers
- Don't worry about time tables, keep goals realistic
  - Some projects manage time-based releases, though

Significant Debian lessons (3)

- Be open and keep things public
  - bug tracking, mailing lists, decision making
- Make it easy to contribute
  - a project can only succeed in the long run if new people join it
  - Some barrier of entry may be necessary for quality and security

The most significant free software development project lesson of all time

Have fun!

The boring final slide

- http://www.debian.org
- http://liw.iki.fi/liw/texts/debian-lessons