Debian in Context
Distributions, Upstreams, and Downstreams

Stefano Zacchiroli

Debian Project Leader

26 August 2011
GNU Hackers Meeting
IRILL — Paris, France

Outline

1. Distributions
2. Downstream
3. Upstream
   - Debian and GNU

Outline

1. Distributions
2. Downstream
3. Upstream
   - Debian and GNU

Reminder: what problems distributions solve?

foo is cool, let's install it!

1. download foo-1.0.tar.gz
   - bonus point: verify checksums and GPG-sig
2. ./configure
   - error: missing bar, baz, . . .
3. foreach (bar, baz, . . .) go to 1
   until (recursive) success
4. make
   - error: symbol not found
5. make install
   - error: cp: cannot create regular file /some/weird/path

now try scale that up to 17'000 sources releasing 3'000 new versions/month

Distributions

Distributions act as intermediaries between upstream software authors and final users. Distributions are meant to ease Free Software life cycle management.

Within distributions scope:

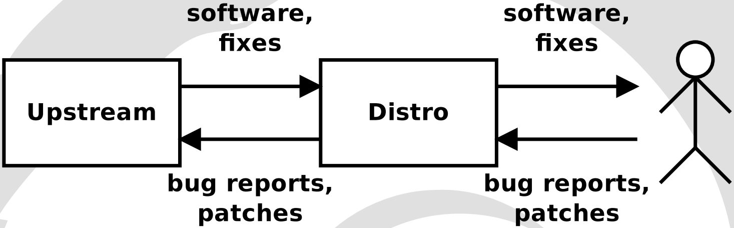
- package management
- trusted sw delivery
- sw integration
- initial installation
- sw packaging
- upstream release tracking
- bug triage and forwarding
- (porting)

Outside distribution scope:

- upstream sw development (but beware of overlaps)
- “shielding” users from upstream and vice-versa

Free Software distribution pipeline (early 90s)

distributions have invented the current way we distribute FOSS:



everybody wins (if done properly)

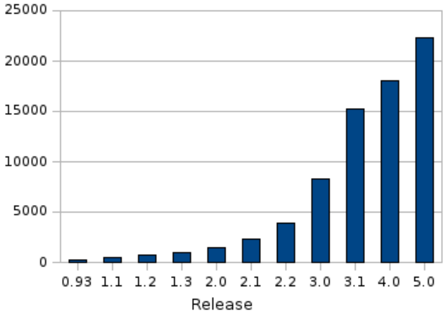
- freedom spreads
- more eyeballs, more shallow bugs
- more potential contributors

inspired by GNU (1983), Debian (1993) has turned distribution editing into a community effort

The Debian distribution

flagship product: Debian stable

- binary distribution
- completely Free (DFSG)
- released every 24 months (≈)
- a dozen architectures
- archive-wide security support (3-3.5 years)



renowned for

ports, stability, packaging system, old hardware support, documentation, smooth upgrades, i18n/l10n, the `testing` suite, runs anywhere, technical policy, package choice, ...

one of the largest GNU/Linux porting platforms

The Debian Project

Common cause:

Create the best, Free operating system.

Debian Social Contract (1997)

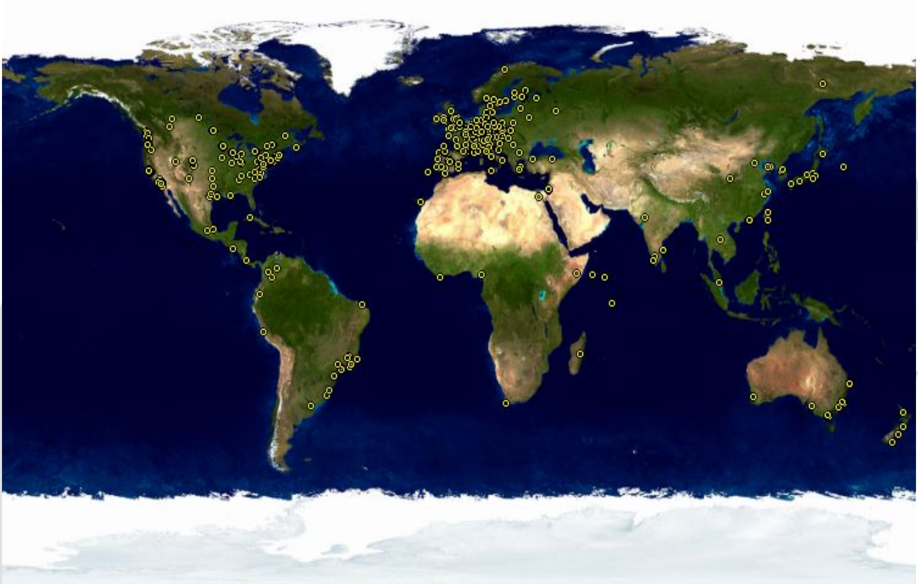
- 100% Free Software
- give back
- don't hide problems
- priorities: users & Free Software

Debian Constitution (1998)

Structures and rules of a Free-Software-compatible democracy

Strong ideal motive to join: ≈ 1'000 volunteer members world-wide

The Debian Project (cont.)



What’s so special about Debian?

There are 300+ *active* distributions (distrowatch)
What’s so special about Debian? Why should you care?

1. freedom
2. package quality
3. independence
4. decision making
5. a distro with (several) downstreams

What’s so special about Debian?

There are 300+ *active* distributions (distrowatch)
What’s so special about Debian? Why should you care?

1. freedom
    - rooted in Free Software principles
    - free the bottom up, dogfooding
2. package quality
3. independence
4. decision making
5. a distro with (several) downstreams

What’s so special about Debian?

There are 300+ *active* distributions (distrowatch)
What’s so special about Debian? Why should you care?

1. freedom
2. package quality
   - policy
   - maintainers are experts of the sw they package
3. independence
4. decision making
5. a distro with (several) downstreams

What’s so special about Debian?

There are 300+ *active* distributions (distrowatch)
What’s so special about Debian? Why should you care?

1. freedom
2. package quality
3. independence
   - volunteers
   - no babysitting company
4. decision making
5. a distro with (several) downstreams

What’s so special about Debian?

There are 300+ *active* distributions (distrowatch)
What’s so special about Debian? Why should you care?

1. freedom
2. package quality
3. independence
4. decision making
   - do-ocracy & democracy
   - no imposed decision; “show me the code” discipline
5. a distro with (several) downstreams

What's so special about Debian?

There are 300+ *active* distributions (distrowatch)
What's so special about Debian? Why should you care?

1. freedom
2. package quality
3. independence
4. decision making
5. a distro with (several) downstreams
    - derivatives!

Outline

1. Distributions

2. Downstream

3. Upstream
   - Debian and GNU

Interlude — derivatives how to

Free Software 101

Freedom #2, to redistribute copies

Freedom #3, to improve the program, and release improvements

When applied to distros: derived distributions, AKA derivatives

How?

1. take existing packages and add your extras
2. patch & rebuild packages as needed
3. sync periodically

Derivatives are game changers

Derivatives have changed the way in which distros are made

- derivatives' focus is on customization
- people power is needed "only" for that

everybody wins (if done properly)

- derivative: massive reuse of packaging work
- upstream distro: reach out to new public
  - of users *and* contributors

Debian derivatives

Debian: a base for ≈130 *active* derivatives (distrowatch)

- Linspire, Liurex, Mint, LiMux, Sidux, gnuLinEx, Grml, MEPIS, Xandros, Ubuntu, Univention, Damn Small Linux, Collax, Euronode, Floppix, Gibraltar, Kanotix, Knoppix, PureOS, gNewSense, LMDE, 64 Studio, Elive, Freespire, Jolicloud, Kurumin, Maemo, Neopwn, OpenZaurus, Parsix, Xebian, Hackable:1, aptosid, . . .

Why?

- quality & licensing assurances
- solid base system
- huge package base
- the "*universal* OS", perfect for customizations

The derivatives crisis

Starting 2006-2007, Debian went through a “crisis” wrt derivatives

- some derivatives finally became *very* popular (20x Debian)
- bad feelings and disattended expectations ensued on each side

Debian’s feelings

- *“they are not giving back!”*
- *“they are taking all the credit!”*

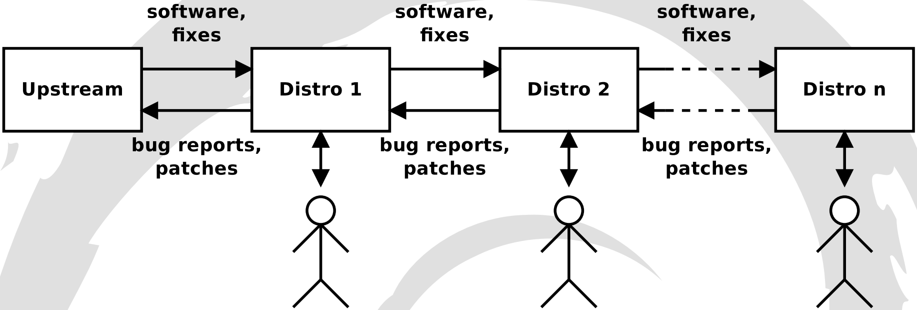
Derivatives’ feelings (guesswork)

- *“Debian is not easy to work with”*
- *“Debian is hostile to us”*

Net result: very little collaboration, fork risk

sounds familiar to any upstream in the room?

Free Software distribution pipeline (mid 2000s)



Upstream

software, fixes

bug reports, patches

Distro 1

software, fixes

bug reports, patches

Distro 2

software, fixes

bug reports, patches

Distro n

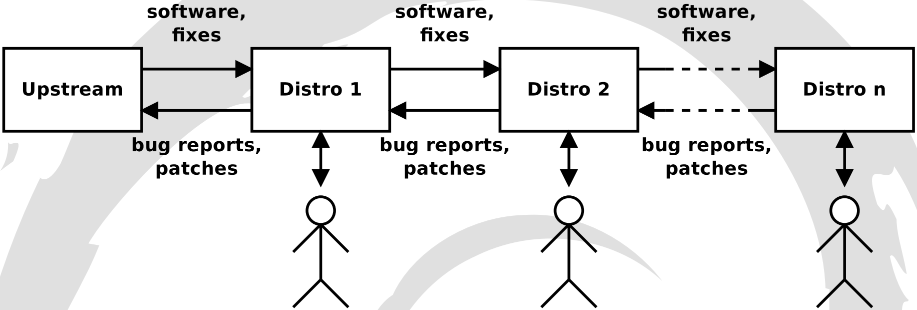
That’s wonderful!

- freedom spreads
- more eyeballs, more shallow bugs
- more potential contributors

But.

- should be sustainable
- and benefit Free Software as a whole

Free Software distribution pipeline (mid 2000s)



That’s wonderful!

- freedom spreads
- more eyeballs, more shallow bugs
- more potential contributors

But.

- should be sustainable
- and benefit Free Software as a whole

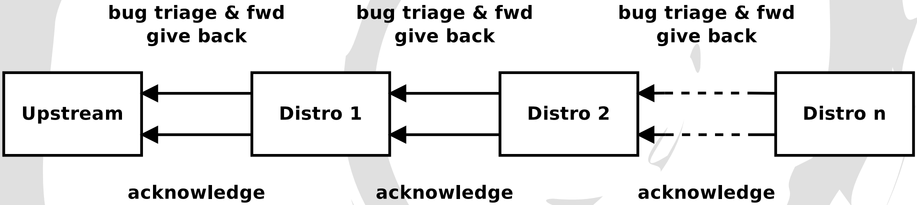
The role of downstream

Share (or at least understand) our view:

Free Software is bigger and more important
than Debian, or any other distro or project

Free Software golden rules and the distro pipeline

1. push changes upstream, i.e. reduce patch flow viscosity
2. give credit where credit is due



Collaboration with derivatives

Derivatives Front Desk

- contact point and discussion place
- emergence of a critical mass of DDs interested in collaboration with derivatives

http://wiki.debian.org/DerivativesFrontDesk

Debian dErivatives eXchange (DEX)

- short-lived cross-distro projects to merge back changes
- visible progress

http://dex.alioth.debian.org

Derivatives Census

- gather detailed information about derivatives
- useful to Debian and for relationship development

http://wiki.debian.org/Derivatives/Census

Outline

1 Distributions

2 Downstream

3 Upstream
- Debian and GNU

How to make distro packagers happy

Upstream guide

`http://wiki.debian.org/UpstreamGuide`

- pristine source — we *have to* rebuild everything
  - in a tarball
  - without debian/
- copyright & licenses
  - clearly stated copyright & license, with per-source-file breakdown
  - DFSG-free licenses only
- sane versioning scheme
  - we *will* totally order it!

How to make distro packagers happy (cont.)

Upstream guide

http://wiki.debian.org/UpstreamGuide

- no convenience copies
- be clear about external dependencies
- “sane-minded” build system
  - stick to the well-known one
- have and use an open bug tracking system
- test suites
  - both build-time and run-time welcome

How to make distro packagers happy (cont.)

Upstream guide

`http://wiki.debian.org/UpstreamGuide`

after initial packaging:

- we always welcome upstream-downstream collaboration!
  - with the package maintainer
  - with porters
- resources:
  - bug tracking system at `http://bugs.debian.org/`
  - package tracking system at `http://packages.qa.debian.org/` allow subscription to Debian events related to your software
  - patch tracker at `http://patch-tracker.debian.org`

Debian an GNU: let's talk

Mutual feedback experiment for GHM 2011:

> *I would like [. . .] to present how we feel about GNU as an upstream, in case we have anything more to say about GNU than what we have to say about other upstreams [. . .] if you are maintaining a GNU package in Debian, I'd like to hear from you. Please mail me comments about your relationship with GNU as an upstream, whether it is good or bad [. . .]*
>
> — *myself*
>
> *http://lists.debian.org/debian-devel/2011/06/msg00518.html*

- good participation: 15 maintainers replied offering articulated feedback

Ludovic Courtès has done the same, switching sides

GNU as a Debian upstream — the good

With a 4:1 ratio, Debian maintainers feel that relationships with GNU are good, generally better than with other upstreams.

Some highlights:

- very responsive upstream
- *“the fact that they care about licensing helps [. . .] maintaining GNU software is usually less work”*
- several GNU maintainers are Debian users — *“it helps”*
- good relationships with some GNU-backed derivatives, active in the derivatives front desk

GNU as a Debian upstream — the bad

Usual suspect:

- GFDL with invariant sections is not DFSG-free
  - reported by most Debian maintainers
  - breaks pristine tarball assumption
  - brings non-free to Debian users of (some) GNU manuals

Dark spots: (most hint at heterogeneity in maintenance practices)

- unmaintained GNU software / unresponsive maintainers
  - hint: encourage adoption of open bug tracking systems
- perceived lack of transparency in some technical statements
- copyright assignment might induce reluctance to give back
- lack of central security contact / approach to security varies a lot

Misc / minor:

- *"savannah interface needs a serious facelift"*

GNU as a Debian upstream — the bad

Usual suspect:

- GFDL with invariant sections is not DFSG-free
  - reported by most Debian maintainers
  - breaks pristine tarball assumption
  - brings non-free to Debian users of (some) GNU manuals

Dark spots: (most hint at heterogeneity in maintenance practices)

- unmaintained GNU software / unresponsive maintainers
  - hint: encourage adoption of open bug tracking systems
- perceived lack of transparency in some technical statements
- copyright assignment might induce reluctance to give back
- lack of central security contact / approach to security varies a lot

Misc / minor:

- *“savannah interface needs a serious facelift”*

GNU as a Debian upstream — the bad

Usual suspect:

- GFDL with invariant sections is not DFSG-free
  - reported by most Debian maintainers
  - breaks pristine tarball assumption
  - brings non-free to Debian users of (some) GNU manuals

Dark spots: (most hint at heterogeneity in maintenance practices)

- unmaintained GNU software / unresponsive maintainers
  - hint: encourage adoption of open bug tracking systems
- perceived lack of transparency in some technical statements
- copyright assignment might induce reluctance to give back
- lack of central security contact / approach to security varies a lot

Misc / minor:

- *“savannah interface needs a serious facelift”*

Debian and GNU goals

Striking similarities in goals

- an *operating system*
- *UNIX*-like

  initially narrower (Linux-only) scope in Debian; largely aligned nowadays
- granting fundamental user *Freedom*s

Different and complementary means to achieve it:

- GNU by <u>developing</u> OS components
- Debian by <u>assembling</u> existing components

We have a lot more in common
than what we disagree on.

Debian and GNU goals

Striking similarities in goals

- an *operating system*
- *UNIX*-like

  initially narrower (Linux-only) scope in Debian; largely aligned nowadays
- granting fundamental user *Freedoms*

Different and complementary means to achieve it:

- GNU by developing OS components
- Debian by assembling existing components

We have a lot more in common
than what we disagree on.

Thanks!

Questions?

Stefano Zacchiroli
leader@debian.org

http://upsilon.cc/zack
http://identi.ca/zack