Legal Compliance in Large Free Software Communities

The case of Debian

Stefano Zacchiroli

Debian Developer
Former Debian Project Leader
OSI Board Director

5 June 2014
OSS Task Force Workshop
OSS trends and state of the art of license compliance
Siemens
Erlangen, Germany

Outline

1. Legal fundamentals of Debian

2. License compliance

3. Looking forward

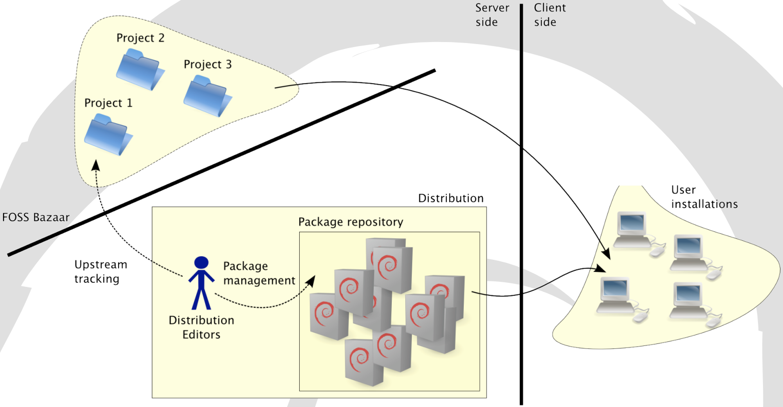
Outline

1. Legal fundamentals of Debian

2. License compliance

3. Looking forward

Prelude: distributions



- ease software management
- key notion: the package abstraction
- offer coherent software collections
- killer application: package managers (& installers)

Once upon a time

> *Fellow Linuxers,*
>
> *This is just to announce the imminent completion of a brand-new Linux release, which I'm calling the Debian Linux Release. [. . .]*
>
> *Ian A Murdock, 16/08/1993*
> `comp.os.linux.development`

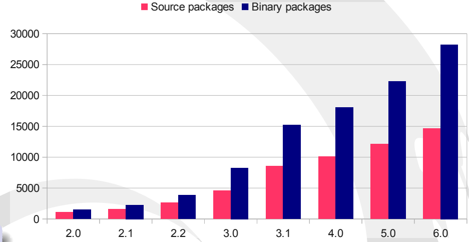
- built collaboratively by software experts
- 1st major distro developed "*openly in the spirit of GNU*"

  FSF-supported for a while

Debian — the operating system

flagship product: Debian stable

- binary distribution
- released every ≈24 months
- 12 hw architectures
- archive-wide security support
  - new: LTS, 5 years



renowned for

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

possibly the largest curated Free Software collection

Web server FOSS market lead (31.2%) — W3 Techs, Jan 2014

Debian — the Project

Common goal:

Create the best, Free operating system.

Debian Social Contract (excerpt) (1997)

1. 100% Free Software
2. give back
3. don't hide problems
5. works that do not meet our Free Software standards

- ≈ 1'000 official members world-wide
- ≈ 4–5'000 contributors
- volunteers, no employees

Debian — the ecosystem

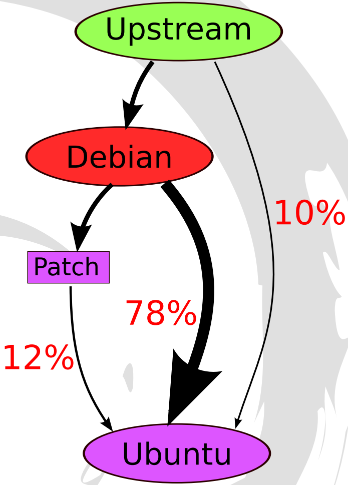
Derivative distribution:

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

Debian: base for ≈140 (48%) distros
— DistroWatch, Jan 2014

Why?

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



Data for Raring Ringtail, Jan 2013,
main + universe

Fundamental #1 — DFSG

To verify the *“100% Free”* promise, you need to define “Free”.
The Debian Free Software Guidelines (DFSG) give such a definition.

`http://www.debian.org/social_contract#guidelines`

- require the 4 freedoms to uphold
- + distribution specific provisions
- basis for the Open Source Definition
- apply to all sorts of content
  - firmware, documentation (PDFs!), artwork, music, . . .

Fundamental #2 — Governance

Debian Constitution (1998)

Structures and rules for a Free Software-compatible democracy

on paper: pretty formal

- bodies: DPL, delegates, technical committee, secretary, . . .
- procedures: NM process, general resolutions, . . .

in practice: flat, bottom-up, almost anarchic

- teams (100x), maintainers (1'000x)
- all (almost entirely) autonomous in technical decisions

Fundamental #3 — Independence

no (or very little) corporate control over Debian

- no (single) company babysitting us
- living up on: donations, gift-economy
- truly remarkable among “major” distros

drawback: limited access to typical corporate resources

assets (money, hw, IP) held by trusted organizations world-wide

- e.g.: SPI (US), FFIS (Germany), debian.ch, . . .
- to reduce SPOF risk
- there is no “Debian foundation”
  TOs used for fiscal sponsorship, and more

Fundamental #3 — Independence

no (or very little) corporate control over Debian

- no (single) company babysitting us
- living up on: donations, gift-economy
- truly remarkable among “major” distros

drawback: limited access to typical corporate resources

assets (money, hw, IP) held by trusted organizations world-wide

- e.g.: SPI (US), FFIS (Germany), debian.ch, . . .
- to reduce SPOF risk
- there is no “Debian foundation”
  TOs used for fiscal sponsorship, and more

Some consequences

At different scales, these traits apply to most "community-driven FOSS projects".

Some consequences:

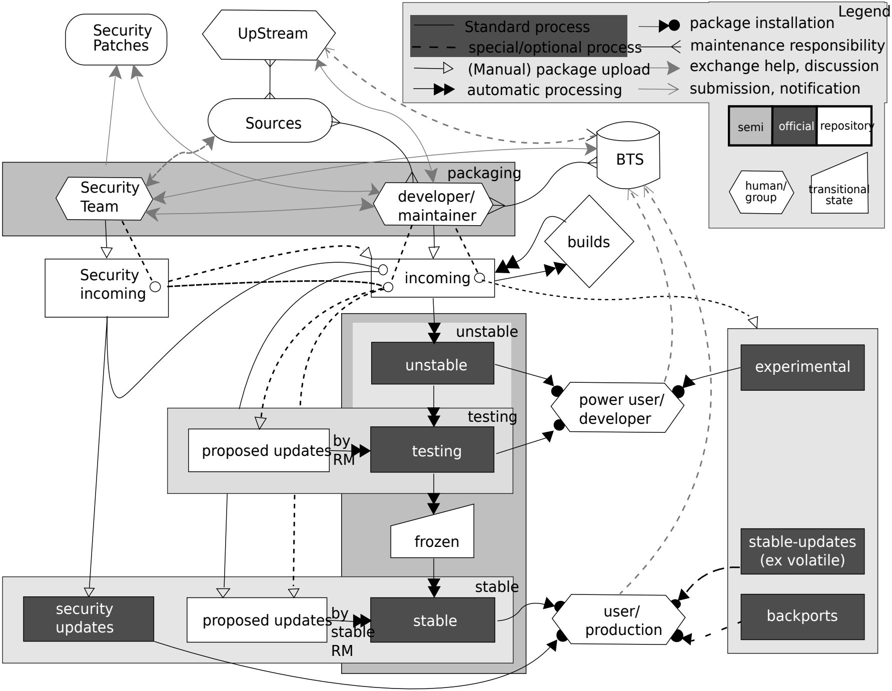
- top-down *"thou shalt not. . . "* doesn't work
- limited access to legal advice
- some "US-centrism"

Outline

1. Legal fundamentals of Debian

2. License compliance

3. Looking forward


Legend
Standard process
special/optional process
(Manual) package upload
automatic processing
package installation
maintenance responsibility
exchange help, discussion
submission, notification
semi
official
repository
human/
group
transitional
state
Security
Patches
UpStream
Sources
BTS
packaging
Security
Team
developer/
maintainer
builds
Security
incoming
incoming
unstable
unstable
experimental
power user/
developer
testing
proposed updates
by
RM
testing
frozen
stable
security
updates
proposed updates
by
stable
RM
stable
user/
production
stable-updates
(ex volatile)
backports

Compliance non-issues

Typical license compliance concerns that do *not* arise in Debian:

- ~~*"release everything but your secret sauce"*~~ — T. Preston-Werner
  - Free Software commitment
  - we *want* to release everything

- ~~copyright assignment / contributor license agreement~~
  within limits though:
  - responsibility waiving (e.g., *post mortem* license upgrades)
  - delegate license enforcement to trusted 3rd parties

Actual “compliance” issues

- keep Debian (main) 100% DFSG-free (mission)
- keep Debian mirrors content re-distributable (legal)
  - non-free is a relevant concern here

debian/copyright

- human readable file that collects all copyright & license notices for any given (source) package[1]
- developers: must fill it in, reviewing upstream notices
- users: for any given (binary) package PKG, will find it under `/usr/share/doc/PKG/copyright`
- popular licenses' full texts are collected under `/usr/share/common-licenses/` and referenced from debian/copyright

- incorrect debian/copyright → release critical bug
- prevention: "user testing" + periodic (in theory) review by package maintainers

[1] `www.debian.org/doc/debian-policy/ch-docs.html#s-copyrightfile`

debian/copyright

- human readable file that collects all copyright & license notices for any given (source) package[1]

- developers: must fill it in, reviewing upstream notices
- users: for any given (binary) package PKG, will find it under `/usr/share/doc/PKG/copyright`
- popular licenses' full texts are collected under `/usr/share/common-licenses/` and referenced from debian/copyright

- incorrect debian/copyright → release critical bug
- prevention: "user testing" + periodic (in theory) review by package maintainers

[1] `www.debian.org/doc/debian-policy/ch-docs.html#s-copyrightfile`

debian/copyright

- human readable file that collects all copyright & license notices for any given (source) package[1]
- developers: must fill it in, reviewing upstream notices
- users: for any given (binary) package PKG, will find it under `/usr/share/doc/PKG/copyright`
- popular licenses' full texts are collected under `/usr/share/common-licenses/` and referenced from debian/copyright
- incorrect debian/copyright → release critical bug
- prevention: "user testing" + periodic (in theory) review by package maintainers

[1] www.debian.org/doc/debian-policy/ch-docs.html#s-copyrightfile

Reviewing notices

how do you distribute the responsibility of reviewing upstream notices to a large, almost anarchic hacker community?

Lesson learned

You don't.

Delegating review to individual maintainers doesn't work at this scale.

not all hackers are equally attentive (or even interested) when it comes to legal matters

Reviewing notices

how do you distribute the responsibility of reviewing upstream notices to a large, almost anarchic hacker community?

Lesson learned

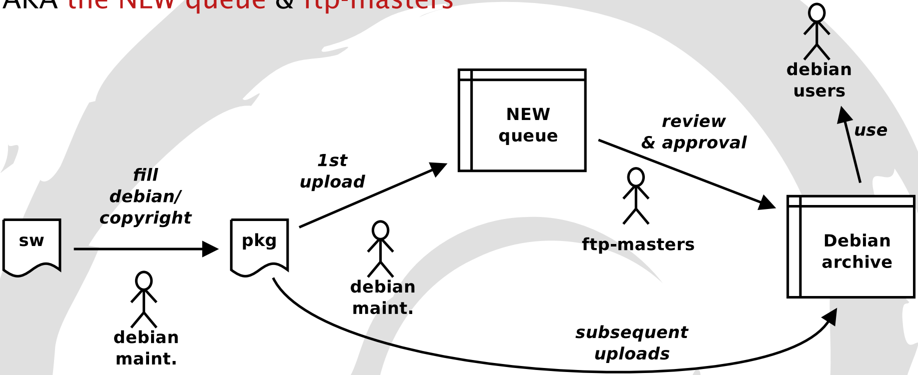
You don't.

Delegating review to individual maintainers doesn't work at this scale.

not all hackers are equally attentive (or even interested) when it comes to legal matters

Reviewing notices (cont.)

AKA the NEW queue & ftp-masters



- 2 tier review process
  peer review—among “legal geeks”—might be a viable alternative
- main purpose: check DFSG free-ness

Quality assurance on debian/copyright

At Debian scale, (semi-)automated QA on license information is highly desirable, e.g.:

- do we link OpenSSL w/ GPL (unwillingly)?
- how many GPLv3-incompatible packages do we have? (2007)
- what happens when `libbdb` is relicensed to AGPL? (2013)
- ...

Approach

Heuristics to cross-check package metadata (Depends, Build-Dep., etc.) with license info → spot candidates for further review.

Requirement: machine-readable debian/copyright

Machine-Readable (M-R) debian/copyright

2007 early versions
2012 version 1.0

http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/

```
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: X Solitaire
Source: ftp://ftp.example.com/pub/games

Files: *
Copyright: Copyright 1998 John Doe <jdoe@example.com>
License: GPL-2+
 This program is free software; you can redistribute it and/or modify it under the terms of the
 GNU General Public License as published by the Free Software Foundation; [snip]
 .
 On Debian systems, the full text of the GNU General Public License version 2 can be found
 in the file '/usr/share/common-licenses/GPL-2'.

Files: complex-1/*
Copyright: Copyright 1998 Jane Smith <jsmith@example.net>
License: GPL-2+ with OpenSSL exception
 [LICENSE TEXT]

Files: complex-2/*
Copyright: Copyright 1998 Jane Smith <jsmith@example.net>
License: GPL-2+ or Artistic-2.0, and BSD
 [LICENSE TEXT]
```

M-R debian/copyright — details

Surface syntax: RFC 822-like “paragraphs”

Header paragraph

```
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: SOFTware
Upstream-Contact: John Doe <john.doe@example.com>
Source: http://www.example.com/software/project
```

Files paragraphs and globbing

```
Files: *
Copyright: 1975-2010 Ulla Upstream
License: GPL-2+

Files: debian/*
Copyright: 2010 Daniela Debianizer
License: GPL-2+

Files: debian/patches/fancy-feature
Copyright: 2010 Daniela Debianizer
License: GPL-3+

Files: */*.1
Copyright: 2010 Manuela Manpager
License: GPL-2+
```

M-R debian/copyright — details

Surface syntax: RFC 822-like “paragraphs”

Header paragraph

```
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: SOFTware
Upstream-Contact: John Doe <john.doe@example.com>
Source: http://www.example.com/software/project
```

Files paragraphs and globbing

```
Files: *
Copyright: 1975-2010 Ulla Upstream
License: GPL-2+

Files: debian/*
Copyright: 2010 Daniela Debianizer
License: GPL-2+

Files: debian/patches/fancy-feature
Copyright: 2010 Daniela Debianizer
License: GPL-3+

Files: */*.1
Copyright: 2010 Manuela Manpager
License: GPL-2+
```

M-R debian/copyright — details

Surface syntax: RFC 822-like “paragraphs”

Header paragraph

```
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: SOFTware
Upstream-Contact: John Doe <john.doe@example.com>
Source: http://www.example.com/software/project
```

Files paragraphs and globbing

```
Files: *
Copyright: 1975-2010 Ulla Upstream
License: GPL-2+

Files: debian/*
Copyright: 2010 Daniela Debianizer
License: GPL-2+

Files: debian/patches/fancy-feature
Copyright: 2010 Daniela Debianizer
License: GPL-3+

Files: */*.1
Copyright: 2010 Manuela Manpager
License: GPL-2+
```

M-R debian/copyright — details (cont.)

Verbatim license and factorization

```
Files: src/js/foobar/*
License: weird-license
 [LICENSE TEXT]

Files: src/js/editline/*
Copyright: 1993, John Doe
           1993, Joe Average
License: MPL-1.1

Files: src/js/fdlibm/*
Copyright: 1993, J-Random Corporation
License: MPL-1.1

License: MPL-1.1
 [LICENSE TEXT]
```

M-R debian/copyright — details (cont.)

License specification

- ontology of license short names
- minimal license algebra:
  - unary postfix “-*v*” modifier for versions
  - unary postfix “+” modifier for “or later” provisions
  - unary postfix “with *exn*” modifier for predefined exceptions (currently 2: GPL Font and OpenSSL exceptions)

    ```
    License: GPL-2+ with OpenSSL exception
    ```

  - binary infix “or” for multiple licensing
  - binary infix “and” for files containing contributions under different licenses, e.g.:

    ```
    License: GPL-2+ or Artistic-2.0, and BSD-3-clause
    ```

M-R debian/copyright — example

Example (Debian copyright file for LibreOffice 4.1.1)

Available at:

`sources.debian.net/src/libreoffice/1:4.1.4-2/debian/copyright`
`sources.debian.net/src/libreoffice/latest/debian/copyright` (current)

- real-life, large example
- 77 license blocks, 30 of which are distinct
- 1427 lines
  - ≈ 200: globbing and copyright notices
  - ≈ 600: verbatim inclusion of unknown (to the ontology) licenses
  - ≈ 500: verbatim inclusion of known licenses, but not popular enough (in Debian) to be shipped under `/usr/share/common-licenses/` (e.g. CDDL, MPL)

M-R debian/copyright — coverage

Potential: huge corpus of thrice reviewed copyright/license notices for popular Free Software projects.

Archive coverage of machine-readable debian/copyright files:[2]

date	release	source packages	archive coverage
Feb 2011	Squeeze	≈ 2’800	19%
May 2013	Wheezy	≈ 7’400	42%
Jan 2014	*unstable*	≈ 9’700	46%
May 2014	*unstable*	≈ 12’200	55%

[2] note: all (100%) Debian packages have a debian/copyright file, but not all are in the machine-readable format yet

M-R debian/copyright vs SPDX

General features[3]

	SPDX	M-R debian/copyright
target	companies / BOMs	hackers
syntax	extensional	intensional
readability	machine ~~& human~~	machine & human
writability	machine	machine & human

[3] my take, YMMV

M-R debian/copyright vs SPDX (cont.)

License short names

- M-R debian/copyright: ≈30 licenses (w/o versions/variants)
  SPDX: ≈100 licenses (ditto)
  - Debian only lists DFSG-free licenses (e.g., no CC-BY-ND)
  - Debian includes Debian-specific variants (e.g., GFDL-NIV)
  - Debian only lists “popular” licenses; others are verbatim

- license name compatibility
  - collaboration Debian ↔ SPDX working group (*circa* 2010)
  - either party agreed to some renaming
  - Debian added equivalences to the license algebra
    - e.g., GPL-2.0=GPL-2
  - one exception: Zope vs ZPL (?)

M-R debian/copyright vs SPDX (cont.)

License short names

- M-R debian/copyright: ≈30 licenses (w/o versions/variants)
  SPDX: ≈100 licenses (ditto)
  - Debian only lists DFSG-free licenses (e.g., no CC-BY-ND)
  - Debian includes Debian-specific variants (e.g., GFDL-NIV)
  - Debian only lists "popular" licenses; others are verbatim
- license name compatibility
  - collaboration Debian ↔ SPDX working group (*circa* 2010)
  - either party agreed to some renaming
  - Debian added equivalences to the license algebra
    - e.g., GPL-2.0=GPL-2
  - one exception: Zope vs ZPL (?)

M-R debian/copyright — implementations

- lintian: thorough “lint”-like tool to check packages against Debian Policy
  - syntactic checks about M-R debian/copyright
  - e.g., `http://lintian.debian.org/tags/unused-license-paragraph-in-dep5-copyright.html`
- licensecheck2dep5
  - licensecheck (part of devscripts): bare bone, header-based license detector
  - licensecheck2dep5 (part of cdbs): convert licensecheck’s output to M-R debian/copyright
- dh-make-perl: create M-R debian/copyright out of CPAN metadata
- Config::Model::Dpkg CPAN Perl module
  - full implementation (syntax + semantics)
  - prototype bidirectional SPDX converter

Outline

1 Legal fundamentals of Debian

2 License compliance

3 Looking forward

SPDX adoption in Debian

None yet.
None foreseen (yet?).

some issues:

- maintainers: writing SPDX by hand? (out of question)
- maintainers: reading SPDX (ditto)
  - i.e., SPDX as a derived product of something hacker-readable
- archive: generating SPDX for Debian packages? (*cui prodest*)
- maintainers: use upstream SPDX to generate debian/copyright?
  - sure, but upstream SPDX adoption is lacking
- maintainers: use *3rd party* SPDX to generate debian/copyright?
  - extra party to trust
  - which SPDX repository? and where are they?

Synergies SPDX ↔ M-R debian/copyright

M-R debian/copyright

- widespread, due to package format popularity
- thoroughly reviewed (for a community)

SPDX

- main industry standard to convey license information
- good for machines
- hard sell to hackers (writing) and users (reading)

M-R debian/copyright as a friendly way to read and write SPDX

debian/copyright→SPDX

- assumption: trust
- expand wildcards
- compute checksums
- distribute license info

SPDX→debian/copyright

- group files by licenses and directory
- synthesize wildcards

Next step: embrace the idea, write reference converters

Synergies SPDX ↔ M-R debian/copyright

M-R debian/copyright

- widespread, due to package format popularity
- thoroughly reviewed (for a community)

SPDX

- main industry standard to convey license information
- good for machines
- hard sell to hackers (writing) and users (reading)

M-R debian/copyright as a friendly way to read and write SPDX

debian/copyright→SPDX

- assumption: trust
- expand wildcards
- compute checksums
- distribute license info

SPDX→debian/copyright

- group files by licenses and directory
- synthesize wildcards

Next step: embrace the idea, write reference converters

Use case #1: Debian as a SPDX consumer

Issues with current Debian compliance process:

- debian/copyright bootstrap is costly
- further releases are less scrutinized (stability assumption)

Widespread SPDX adoption could help, e.g.:

If we trust upstream

- all upstream releases will come with SPDX (. . .)
- first release: generate debian/copyright (and then review it)
- at each new release: automatic check for changes

If we do not trust upstream

- “forges” will provide SPDX for all projects (. . .)
- at each new release: lookup (by project name or checksum) SPDX data and double-check upstream data

Use case #1: Debian as a SPDX consumer

Issues with current Debian compliance process:

- debian/copyright bootstrap is costly
- further releases are less scrutinized (stability assumption)

Widespread SPDX adoption could help, e.g.:

If we trust upstream

- all upstream releases will come with SPDX (. . .)
- first release: generate debian/copyright (and then review it)
- at each new release: automatic check for changes

If we do not trust upstream

- "forges" will provide SPDX for all projects (. . .)
- at each new release: lookup (by project name or checksum) SPDX data and double-check upstream data

Use case #1: Debian as a SPDX consumer

Issues with current Debian compliance process:

- debian/copyright bootstrap is costly
- further releases are less scrutinized (stability assumption)

Widespread SPDX adoption could help, e.g.:

If we trust upstream

- all upstream releases will come with SPDX (. . .)
- first release: generate debian/copyright (and then review it)
- at each new release: automatic check for changes

If we do not trust upstream

- “forges” will provide SPDX for all projects (. . .)
- at each new release: lookup (by project name or checksum) SPDX data and double-check upstream data

Use case #2: Debian as a license knowledge base

Compliance tools/services are currently quite tied to centralized, non-transparent dataset.

Debian: not as big as GitHub + SourceForge + . . . , but:

- good proxy of popular FOSS
- long release history (20+ years)

What if we turn distros into a large federated dataset for compliance? In Debian:

- mass convert M-R debian/copyright → SPDX
- add lookup APIs (e.g., to `http://sources.debian.net`)
- building block: M-R debian/copyright → SPDX converter

FOSS communities have different compliance needs than industries.
There are synergies to be found on both tools and datasets.

Thanks!

Questions?

Stefano Zacchiroli
zack@debian.org

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