

Risks and Benefits of Copyright Assignment

Or why you should become somewhat allergic to it ...

Michael Meeks

Michael.meeks@suse.com

*“Stand at the crossroads and look; ask for the
ancient paths, ask where the good way is, and walk
in it, and you will find rest for your souls...” -
Jeremiah 6:16*



Outline: the curves ...

- An Agenda outline
- I am **not** a lawyer (really)
- The effect of © assignment on projects
 - Some graphs showing the removal of assignment
 - Arguments people use for and against doing it
 - Funding Free Software development
- How the FSF loses by condoning and practice it
- An emerging tactical gap in the cloud space
- Questions & Stoning ...

My Agenda:

- I believe to-corporate © assignment is a serious problem in the Free Software world
- So I'd like to persuade you (simultaneously) that:
 - It is the best long-term business / legal strategy for your company to avoid and/or drop © assignment
 - This inevitably means competition, collaboration, and selling support.
 - In the cloud: the lack of an ALGPL pushes people to © assign
 - Collectively, we need to fix that.
 - The FSF harms it's own interest by practising © assignment
 - Which it does only for a few projects anyway

I am not a Lawyer
(in case this is not obvious)

So – what am I doing here ?

Lawyers are clueful businessmen too

Pragmatics rule: reality bites ...

**Legal simplicity and elegance
can have substantial hidden costs.**

The Effect of © Assignment on Free Software projects

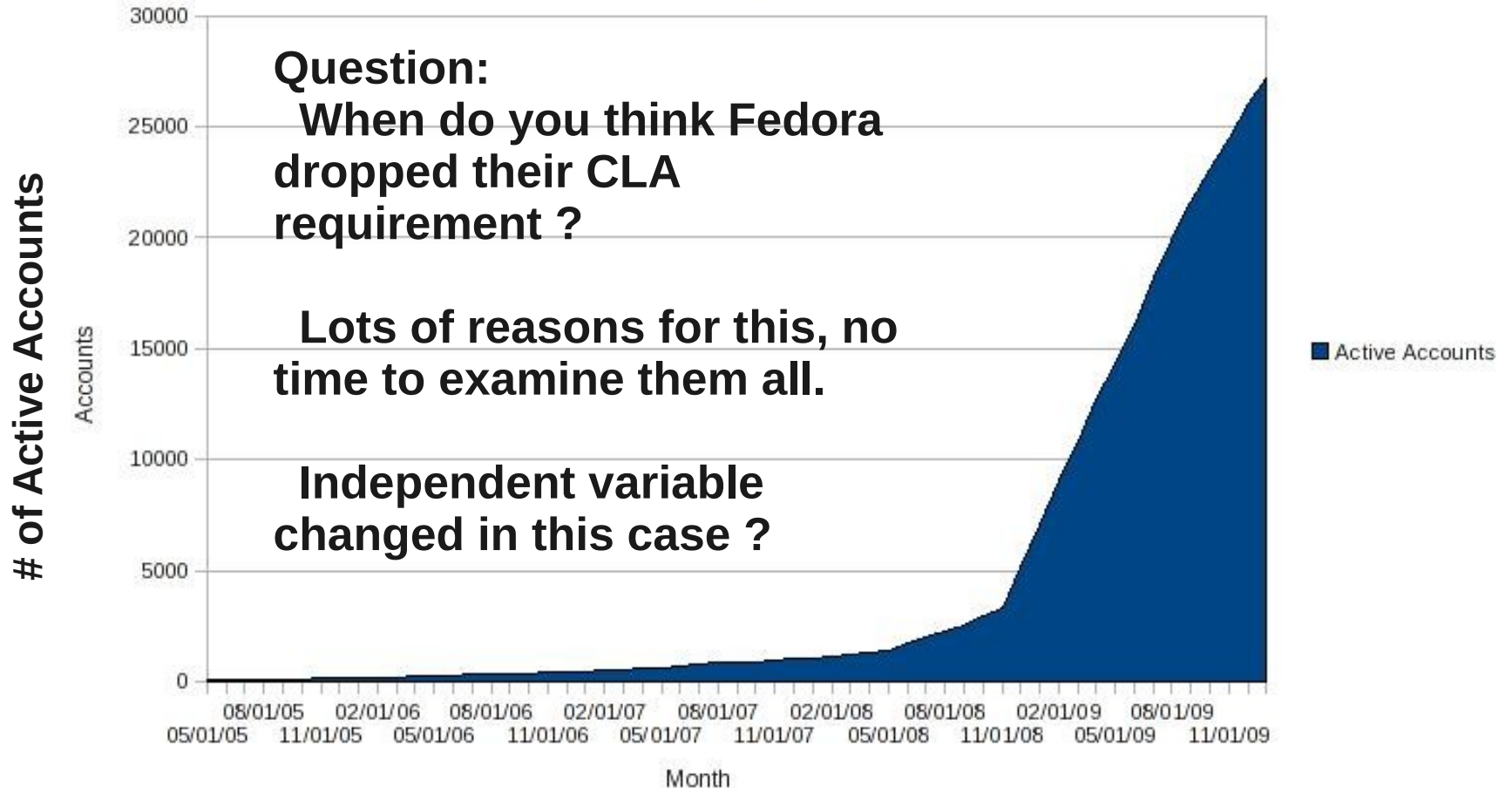
(or, a random selection of graphs and quotes
I found to confirm my pre-existing bias)

Take #1 – Fedora ...

Carefully borrowed from [Olav Vitters](#)

Fedora Active Accounts

May 2005-Feb 2010



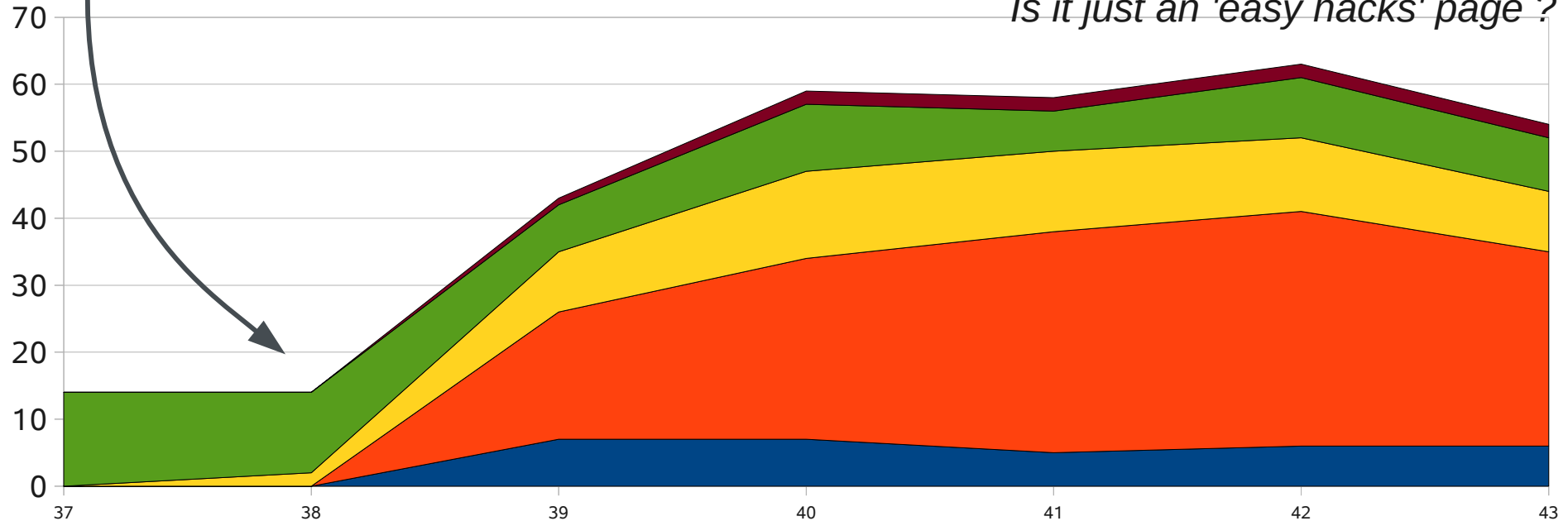
Take #2 - LibreOffice

Some numbers from our launch generated by my with gitdm ...

- At launch – sudden explosion of interest & contribution
 - One factor is clearly – dropping © assignment
 - Freedom from vendor control / ownership of franchise etc.

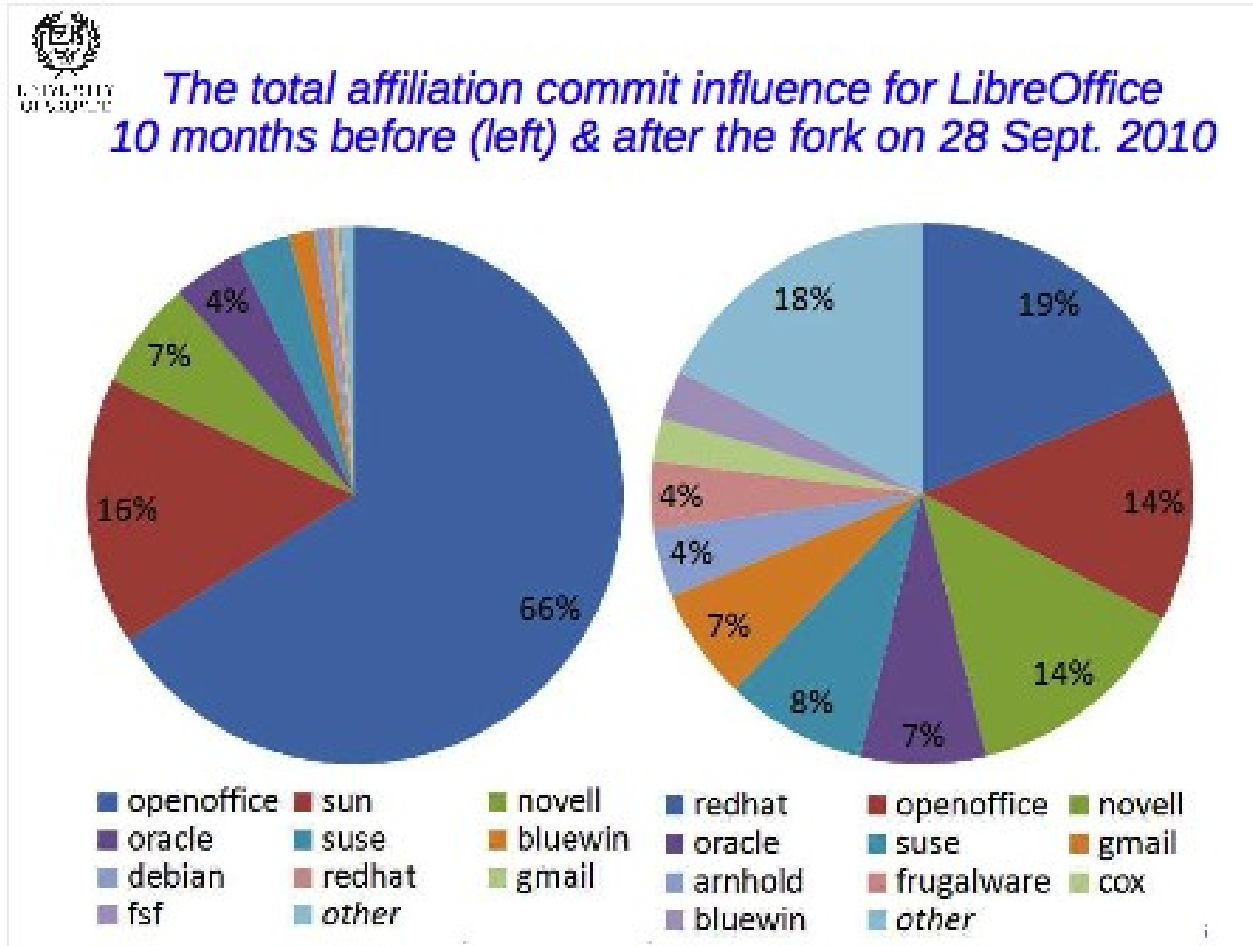
■ Known contributors ■ New contributors ■ Novell
■ Oracle ■ Redhat

*Day #1 nearly same code
Is it just an 'easy hacks' page ?*



Take #2 – LibreOffice

Ask someone else – an academic study of diversity ...



My comment:
It should also be noted the commit rate dramatically increased in this time.

Gamalielsson, J. and Lundell, B. (2011) *Open Source communities for long-term maintenance of digital assets: what is offered for ODF & OOXML?*, in Hammouda, I. and Lundell, B. (Eds.) Proceedings of SOS 2011: *Towards Sustainable Open Source*, Tampere University of Technology, Tampere, ISBN 978-952-15-2718-0, ISSN 1797-836X.



Take #3 – MySQL ...

Taking the words from the horse's mouth ...

- Guaranteed lack of diversity:

*"Similarly, as Prof. Moglen agrees, **it is a fact** that MySQL was almost fully developed by employees of MySQL Ab and later Sun's MySQL division"*

Monty Program AB [response](#) to Commission

- Which takes us to the argumentation on economics:

*"Without MySQL Ab being empowered by ownership of the core, we would not have Oracle MySQL !
Therefore we should encourage this model"*

Or should we ?

**A sketch argument for to-Corporate ©
assignment in Free Software**
(With some thoughts interspersed)

Be grateful for greater traction ...

In the short term, VC and sales/marketing can dominate a space...

*“If you look at projects with **lots of traction**, only **assignment based projects** are a success. We should be **grateful to the companies that do all the heavy lifting (alone).**”*

A counter-suggestion:

*“**Linux** is the paradigmatic successful Free Software project. With **no corporate ownership aggregation**. And there are many others. Indeed – a **single company** doing all the heavy lifting is a **sign of risk**, and **failure is one acquisition away.**”*

- ▼ Try also: Mozilla & WebKit, glibc, gcc, binutils, udev, systemd, GNOME, KDE, perl, python, Apache*, distros.



It's necessary for un-sexy projects ...

There is some class of software that needs 'special' corporate help

*“The only way **companies** can continue to **invest** in working on **non-sexy** projects is if they can **'own'** them”*

A counter-suggestion:

*“We **must not** let **companies**, who are **uncertain** of their continued **investment**, to **exclude others** from co-owning and building out key pieces of our stack”*

Counter examples:

*almost our whole software stack is not owned by companies:
Mozilla & Webkit, GNOME & KDE, Linux, util-linux, udev, glibc,
gcc, binutils, perl, python, Apache*, dbus, etc. etc.*

Personally ~all (cleanly written) software is quite sexy



The OpenSPARC project speaks:

Are we certain that there are no 3rd party patent licenses required ?

- **Disclaimer: this is a project I know – nothing – about: outside of our sphere of business.**
 - **OpenSPARC** – ~GPLv2 **only** + 'Shared' CA
 - *All contributors to OpenSPARC are required to sign the Contributor Agreement ("CA") before contributing code to the community. This enables a **single entity** to **represent** the aggregated code base and gives the '**community**' flexibility to act as a whole to changing situations.*
- SPARC or OpenSPARC ?
 - Safest to proprietary license from Oracle ... ?
 - Needs a 'Welcome' sign with:
 - “Academics, non-profits & partners welcome”



“Why does [Company] have a contributor Agreement ?” - from their clear & helpful **FAQ**

The OCA protects the integrity of the code base, which in turn protects the development community and the project's users.

*For [Company]-sponsored projects (i.e. projects that require an OCA), [Company] **acts on the community's behalf** in the event of any legal challenge.*

- [Intriguingly this commitment is not mentioned in the OCA.]

*This is **in keeping** with how other code stewards, including the **Free Software Foundation**, the **Apache Software Foundation** and the **Eclipse Foundation** operate. In order to represent a code base against legal challenges, [Company] **needs** to have copyright ownership of **all the code** in that project.*

- [Worth noting that Apache and Eclipse do not aggregate © ownership !]



Sometimes useful for re-licensing ...

*Consolidating ownership of the code also allows for the **possibility of relicensing** the whole code base **should that become desirable**. Having the ability to license code under a different license can be a useful tool, and not having that flexibility may be a drawback.*

- [though proprietary licensing is their invariable business practise]
- [Although a “Plus” license or steward would be a better choice]

*Without aggregated copyright, [Company] would have to contact and obtain permission from every single contributor in order to license the code under a different license. **Consolidating ownership in this way is a common practice in various open source communities.***

- [Everyone else is doing it – even the FSF !]



How does the Contributor Agreement help me ? ...

*It allows [Company] to **sponsor** the projects to which you want to contribute, while retaining the ability to offer commercial licenses. **Without this** ability, [Company] **could not responsibly open source** code bases such as OpenJDK or MySQL that represent hundreds of millions of dollars of investment in infrastructure, development, and governance. Moreover, the OCA lets [Company] **protect community members** (both developers and users) from hostile intellectual property litigation should the need arise.*

- [Assertion that XYZ would not exist without this: would that be bad ?]

*The **contributor agreement** also includes an "**open source covenant**", or a promise that a contribution will **remain available** from Oracle as free and open-source software for as long as Oracle continues to distribute the contribution.*

- [These are just a ~meaningless marketing commitment: to a public revision control system for old versions]



The OpenSolaris story ...

How your contribution can 'remain available' ...

- Brian Cantrill's Fork Yeah! / Illumos [talk at Lisa](#) Dec. 2011
- Friday August 13th 2010 internal / memo

"The reason that this is shameful, the reason that this is reprehensible is that a **social contract** was formed with the community, and there are **folks in this room** that had **source code** that was **contributed back** under that **copyright assignment**, and that source code was **now being made proprietary**.

"That is **reprehensible**, that is *****ting** in the pool of open-source, and it is disgusting corporate behavior.

"Sadly it is behavior like this that **forces** us to be **cynical** and **suspicious**, it is a body-blow for open source.

"The worst thing was, not only was it shameful - it was cowardly: because this was **never publicly announced**, Oracle has not publicly announced once ! that they are stopping contributions to OpenSolaris, they **simply, silently stopped**.



Then it gets re-hashed left & right ...

Magnifying the confusion between CA's and © agreements

*As far as contributor agreements go: **FSF has one**, Apache has one, Eclipse has one, Mozilla has one, and ... Sun has had one for a while. - Dalibor Topic*

*Moreover, the CLA **lets us protect** community members (both developers and users) from hostile intellectual property litigation should the need arise. This is in line with how **other free software stewards** like the **Free Software Foundation - FSF defend projects** (except with **the FSF, there's no shared copyright** but instead you completely sign it over to the FSF) – Neo4j FAQ*

*In order to **protect** ClearFoundation and its projects, we ask that developers who contribute **intellectual property** (including patches and source code) **agree** to a contributor agreement. **The Free Software Foundation (FSF)** has a concise *explanation* on why this is **standard practice** for open source projects. - ClearFoundation*

[links to: <http://www.gnu.org/licenses/why-assign.html>]

And many more ... - the FSF is the FAQ poster-child



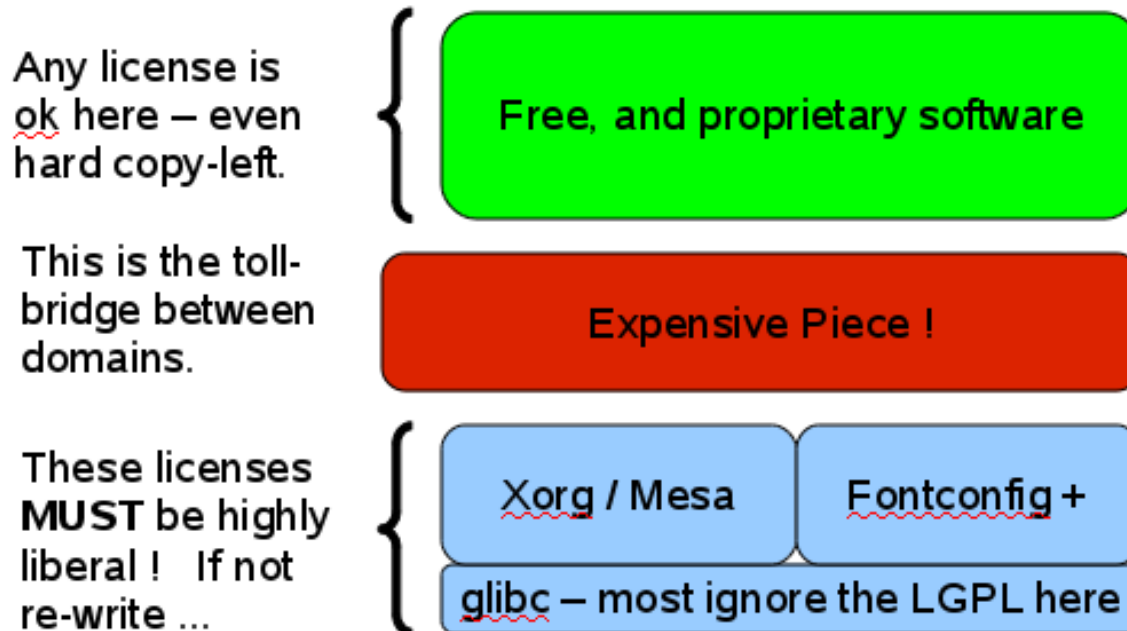
A quick economic view of
Corporate © assignment.

Free Software's magic bullet:

Code re-use, building up-wards not outwards etc.

- **But** Corporate © assignment **doesn't scale**, is rigid, inflexible, mgmt. makes **very poor** tech. decisions **very slowly**, alliances are incredibly expensive

The unfortunate taxonomy of an assignment business model



Other Sample Economic problems

Where there is a shared code-base all can share them ...

- Corporate © assignment creates and sustains a single rent-seeking owner
 - It fails to attract a community that shares your development cost.
 - Yet it gives away your entire product nearly for free ...
 - single entity always fails to efficiently exploit all niches/price-points
- Celebrated “OpenCore” 3rd party Patent trapping:
 - “Buy the specially indemnified version !”*
- Tactically 'bad' licensing is **required** to drive proprietary revenue instead of adoption
 - What %age of tech. / internet startups fail ? - 30%? 50%?
 - What %age of them exit to 'big' companies with no clue ?
 - Leaving us all with un-alterable / poor tactical choices.

How the FSF loses by advocating
and practising © assignment

The FSF and © assignment ...

And/or how we all loose because of this ...

- Assignment to the FSF is **far** more **benign** than to a Corp.

But

- The FSF does not itself require © ownership to protect projects
 - There is little-to-no need for ownership to defend the license.
- An indirect **consequence** of Corporate © assignment is to **kill Plus** (“Or later version”) **licensing**
 - This adds competitive pressure to the FSF – companies can switch to a variety of other licenses since they own the rights.
 - Ergo – FSF is used as a poster child for a practice that **does not** benefit it, or **long-term** software freedom. Worse – it can **tactically kill the future...**
 - Can anyone name any company that © **assigns** & uses an FSF plus license ?
- The FSF helps to drive the industry in an unfortunate direction.



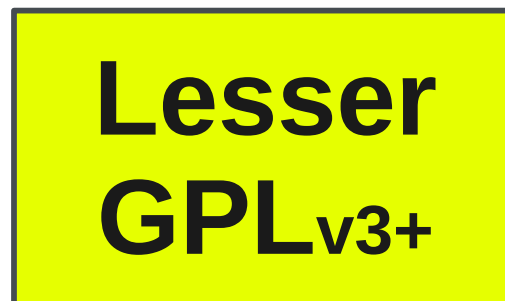
A simple visualization of goodness:

for Free Software purists / enthusiasts

- The simple model we used to explain to people:



Good



Bad



Seldom
mentioned /
only if you
really must ...

RMS – yet again well ahead of his time:

“Why your next library should be GPL” etc.

And of course v3+ is better/stronger/safer than v2.

Some visualizations of goodness:

Dangerous brand combination with © assignment

- A bit more complicated: Notice the loss of the 'plus'

GPLv2 only
© to Oracle

LGPLv2 only
© to Nokia

**Apache/
BSD**
**+License to
foundation**

I posit that the presence of © assignment turns the good, and wholesome restrictions of a license (when applied to all), into a perverse economic incentive to concentrate control, and strangle development of a project – also enabling wide, non-obvious avoidance of GPL.

The ALGPL in the Cloud ...

Who do we want to compete with ?

I want to compete with Proprietary Software ...

- Competing with Proprietary Software built on top of a shared Free Software core is reasonably easy.
 - Competing with tactically useless / competitor excluding / pseudo-Free-Software involves a re-write first.
- What does that mean in the cloud ?

**ALGPL
Web core**



Proprietary Plugin

ALGPL Alternative



- Best for your business to drop the CLA – perhaps with proprietary extras.
- To get code re-use, we need an ALGPL
- FSF needs to re-think it's accidental? corporate © cheerleader role

Thank you.



Oh, that my words were recorded, that they were written on a scroll, that they were inscribed with an iron tool on lead, or engraved in rock for ever! I know that my Redeemer lives, and that in the end he will stand upon the earth. And though this body has been destroyed yet in my flesh I will see God, I myself will see him, with my own eyes - I and not another. How my heart yearns within me. - Job 19: 23-27