Boundaries on freedom and copyleft in decentralized network environments

Christopher Lemmer Webber

website: https://dustycloud.org/
email: cwebber@dustycloud.org
twitter: @dustyweb
fediverse: https://octodon.social/@cwebber
I’ve been doing network freedom stuff for a while

mediagoblin
I’ve been doing network freedom stuff for a while
I’ve also been an advocate and fan of this thing
I think AGPL has been historically a great fit

MediaGoblin, Mastodon, StatusNet all make sense to me as AGPL’ed projects ("Libre Web 2.0")

I find the kinds of protections the AGPL provides desirable

But ...
... but?

- I’m starting to explore some "next generation" network freedom software
- I’m not so sure the boundaries are the same here...
- Copyleft will remain useful / critical, but maybe we move the protection maxima
Let’s step back

Why do we want copyleft?

• User freedom!

• Protect the commons!

And that’s it... right???
The copyleft intentions split

<table>
<thead>
<tr>
<th></th>
<th>Libre Commoner</th>
<th>Proprietary Relicenser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motivation</td>
<td>Protect the commons</td>
<td>Develop income</td>
</tr>
<tr>
<td>Mitigating</td>
<td>Tragedy of the commons</td>
<td>Free Rider Problem</td>
</tr>
<tr>
<td>Wants</td>
<td>Compliance</td>
<td>Non-compliance</td>
</tr>
</tbody>
</table>
Wanting noncompliance is a problem

- I’m highly sympathetic to reducing the free rider problem and funding FOSS... BUT
- Wanting non-compliance means wanting non-freedom
- Copyleft non-compliance has turned out to be insufficient anyway for this group; that’s how we got the SSPL
- We’ll come back to this...
Programs according to FOSS tradition
Let’s talk about something radically different

- Actors allow for massively distributing programs
- Object capabilities allow for massively distributed security
The actor model, eh?

- Imagine *every* object on your program is a microservice
- *Any* of those objects can live anywhere
- Your program as a "society" of experts sending messages to each other, not unlike sending email
[...] However, in many cases you can distribute the GPL-covered software alongside your proprietary system. To do this validly, you must make sure that the free and nonfree programs communicate at arms length, that they are not combined in a way that would make them effectively a single program. [...] — GPL FAQ

What happens when every object communicates with every other object at arm’s length?
ActivityPub *is* an actor model specification
ActivityPub *is* an actor model specification
ActivityPub *is* an actor model specification
ActivityPub *is* an actor model specification
ActivityPub *is* an actor model specification
ActivityPub *is* an actor model specification

ActivityPub is a "social protocol". BUT...
We showed people sending each other messages
Why not parts of a computer program?
We can get as fine grained as we want
Let’s contextualize before things get harder.
Most ActivityPub implementations aren’t taking the actor model this seriously
AGPL is fine/great for eg Mastodon
Ok time to get more complex!
(setq nnmail-split-fancy
  '(or ("X-Spam-Flag" "YES" "Spam")
       ("To" "depaul-lug@linux.depaul.edu" "DGLUG")
       ("To" "chicagolinux-discuss@chicagolug.org" "ChiGlug Discuss")
       ("To" "super-tux-devel@lists.sourceforge.net" "SuperTux")
       ("To" "supertux-team@lists.lethargik.org" "SuperTux")
       ("To" "gridlock@hafd.org" "BreakTheGridlock")
       ("From" "actforchange.com" "Political Campaigns")
       ("From" "ThePeaceAlliance.org" "Political Campaigns")
       ("From" "barackobama.com" "Political Campaigns")
       ("Subject" "Dipierro1" "CPP Class")
       ("To" "plans@pculture.org" "PCF Plans")
       ("To" "discuss@lists.autonomo.us" "Autonomo.us")
       ("To" "cc-staff@lists.ibiblio.org" "CC Staff")))
(setq org-capture-templates
  '(({"t" "Todo" entry
      (file+headline "~/org/life.org" "Various Tasks" )
      "* TODO %?\n %i\n %a" :prepend t :empty-lines 1)
  ("e" "Event" entry
      (file+headline "~/org/life.org" "Events" )
      "* %^{Event}\n %^t\n %i\n %a\n\n%?" :prepend t :empty-lines 1)
  ("W" "Weigh-in (no table)" entry
      (file+headline "~/org/diet.org" "Daily Logs" )
      "* Diet for day %t\n%^{Weight}p"
      :prepend t :empty-lines 1)
  ("b" "Blood pressure" table-line
      (file+headline "~/org/bpressure.org" "Blood pressure table" )
      "| %U | %^{systolic} | %^{diastolic} | %^{pulse} | %^{note} |"
  )
  ("cgt" "Contracting general TODO" entry
      (file+headline "~/org/contracting/contracting.org" "Various Tasks" )
      "* TODO %?\n %i\n %a" :prepend t :empty-lines 1)
  ("co" "Open Tech Strategies" )
  ("cot" "OTS Todo" entry
      (file+headline "~/org/contracting/opentechstrategies.org" "Various Tasks" )
      "* TODO %?\n %i\n %a" :prepend t :empty-lines 1))
(defun org-diet-hacky-jump-to-today ()
  "Hackily jump to today in diet file"
  (interactive)
  (switch-to-buffer (get-file-buffer "~/org/diet.org")))
  (beginning-of-buffer)
  (search-forward-regexp "^| Total")
  (beginning-of-line)
  (previous-line))
Now consider the wants-noncompliance camp

Without the right for private, local modification can someone pressure me to revealing:

- My mail configuration?
- My org-mode setup, revealing my health history?
- My private blocklist?
Object capabilities for safe execution

We can even share and run code safely across the network
This isn’t an abstract for me

Spritely will be exploring this
Is it time for the LAGPL?

What does that mean, what’s the exception?
A library exception? What does that mean for actors?
If you can’t see the remote object, how do you know whether they’re modifying the library or just using it?
But I still want to protect the commons!

Here’s an awful future for TNG networks, featuring lax/permissive licenses:

• I share some cool code on the network

• Someone else picks it up on their machine, modifies it

• They distribute it back to my computer, modified

• Bing! I get a notification that I’m not allowed to run or use or modify that code that came back to me
So we still want to protect the commons

In fact, the GPL protects against the previous scenario!
Is the best LAGPL for TNG networks the GPL?

- Private modification still possible
- Code passed around on network can still be protected
Conclusions

• Actually I’m not confident about my conclusions because I haven’t found enough people thinking about this space

• But AGPL feels good for Libre Web 2.0

• But maybe not for Libre Networks TNG?

• But GPL-level of copyleft still good, necessary
Thanks!

Christopher Lemmer Webber

website: https://dustycloud.org/
email: cwebber@dustycloud.org
twitter: @dustyweb
fediverse: https://octodon.social/@cwebber

Oh, and subscribe to Libre Lounge!
https://librelounge.org