Solving the Deployment Crisis with GNU Guix

Christopher Allan Webber & David Thompson

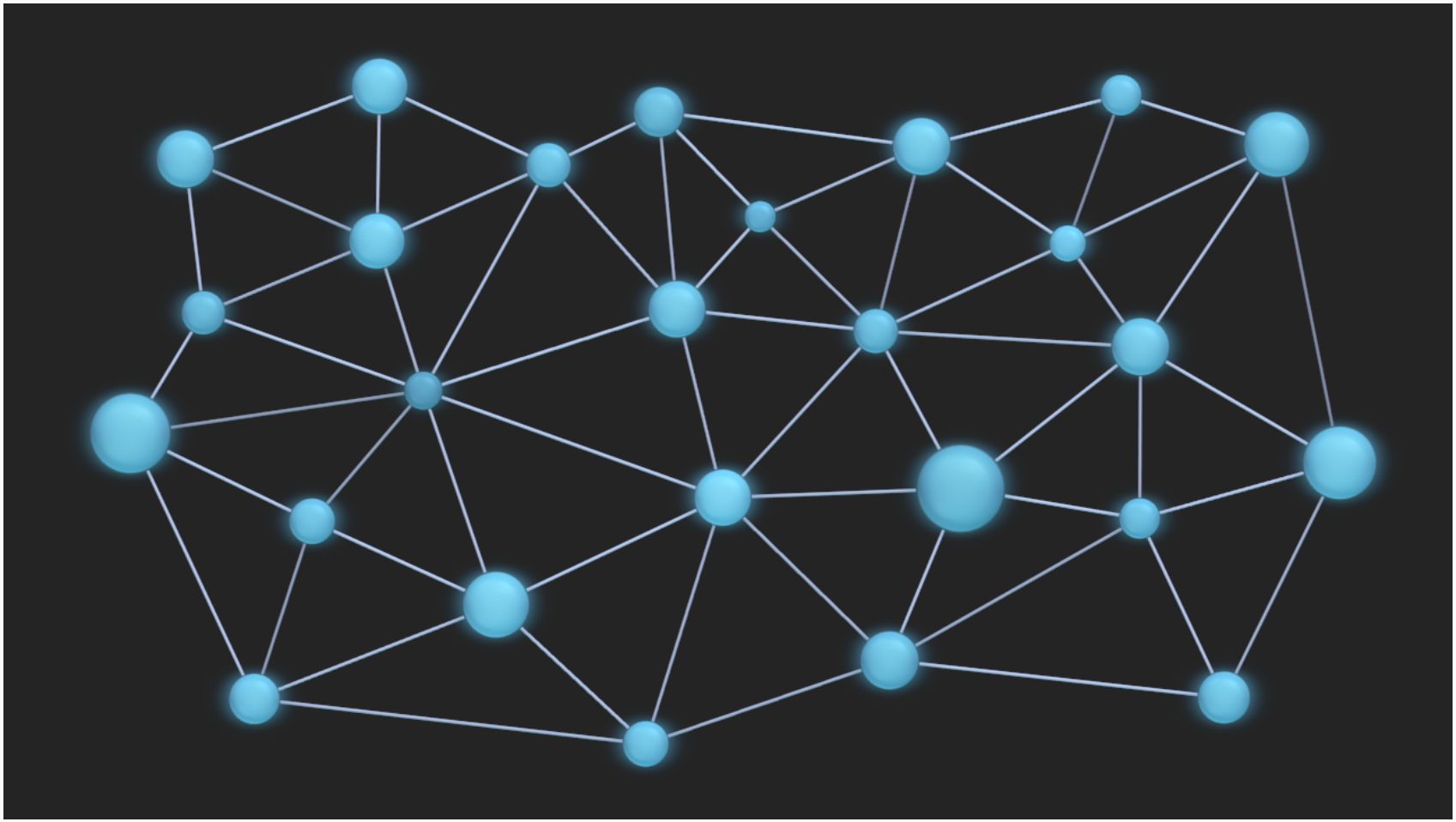
Saturday, March 19th, 2016

Setting the stage

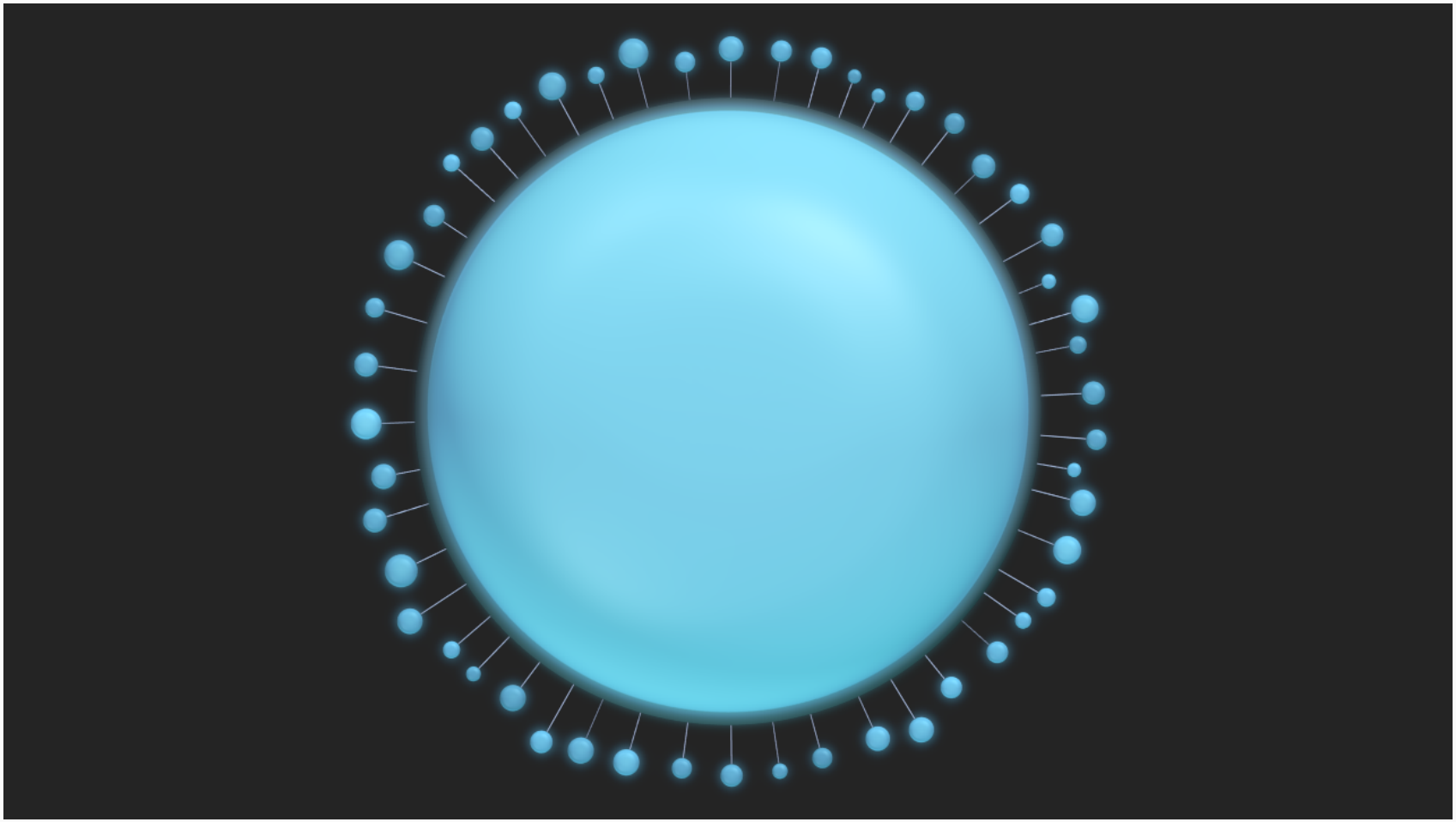
Who are we?

- Who is David Thompson?
- Who is Christopher Webber?

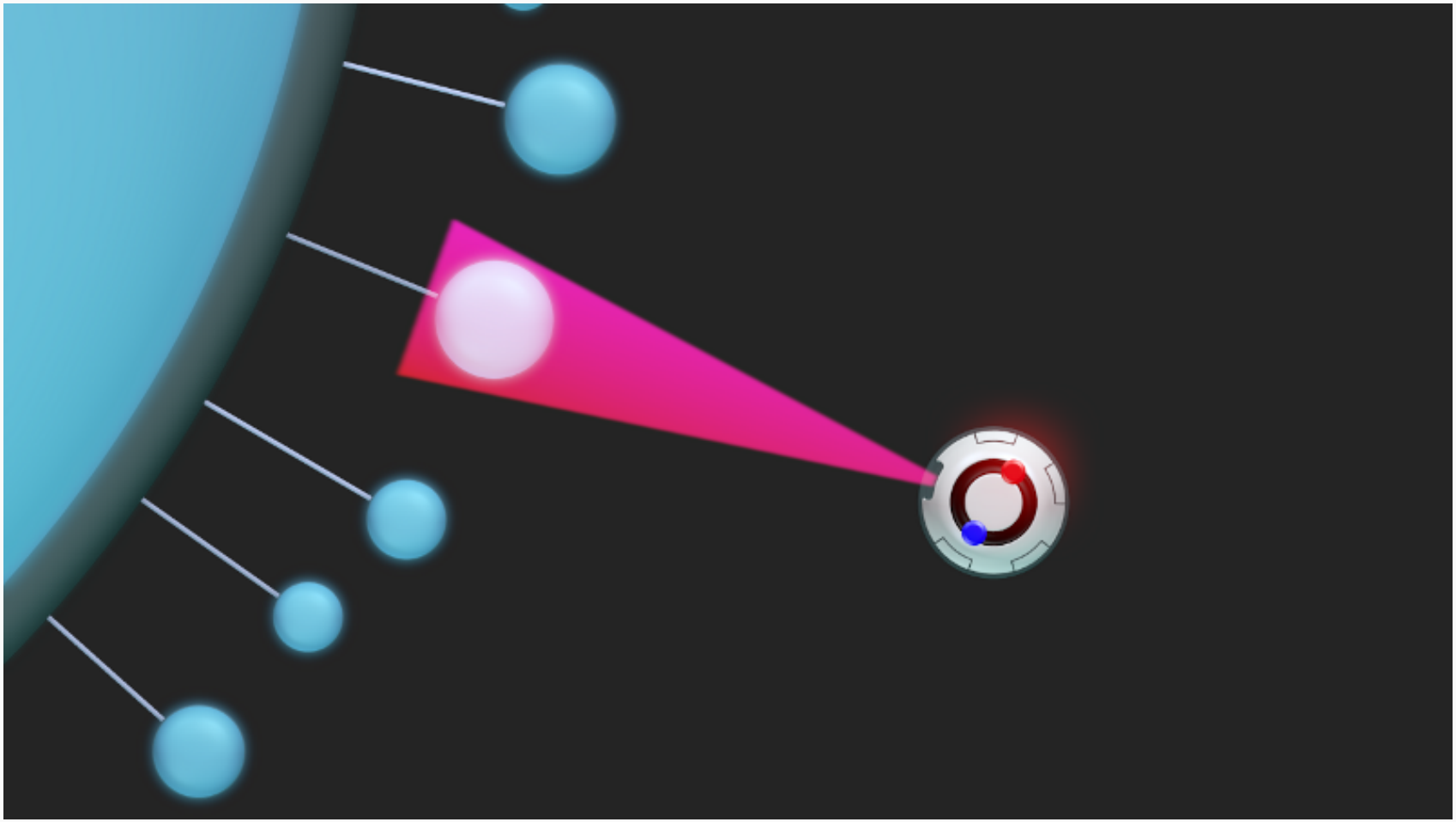
The web we want



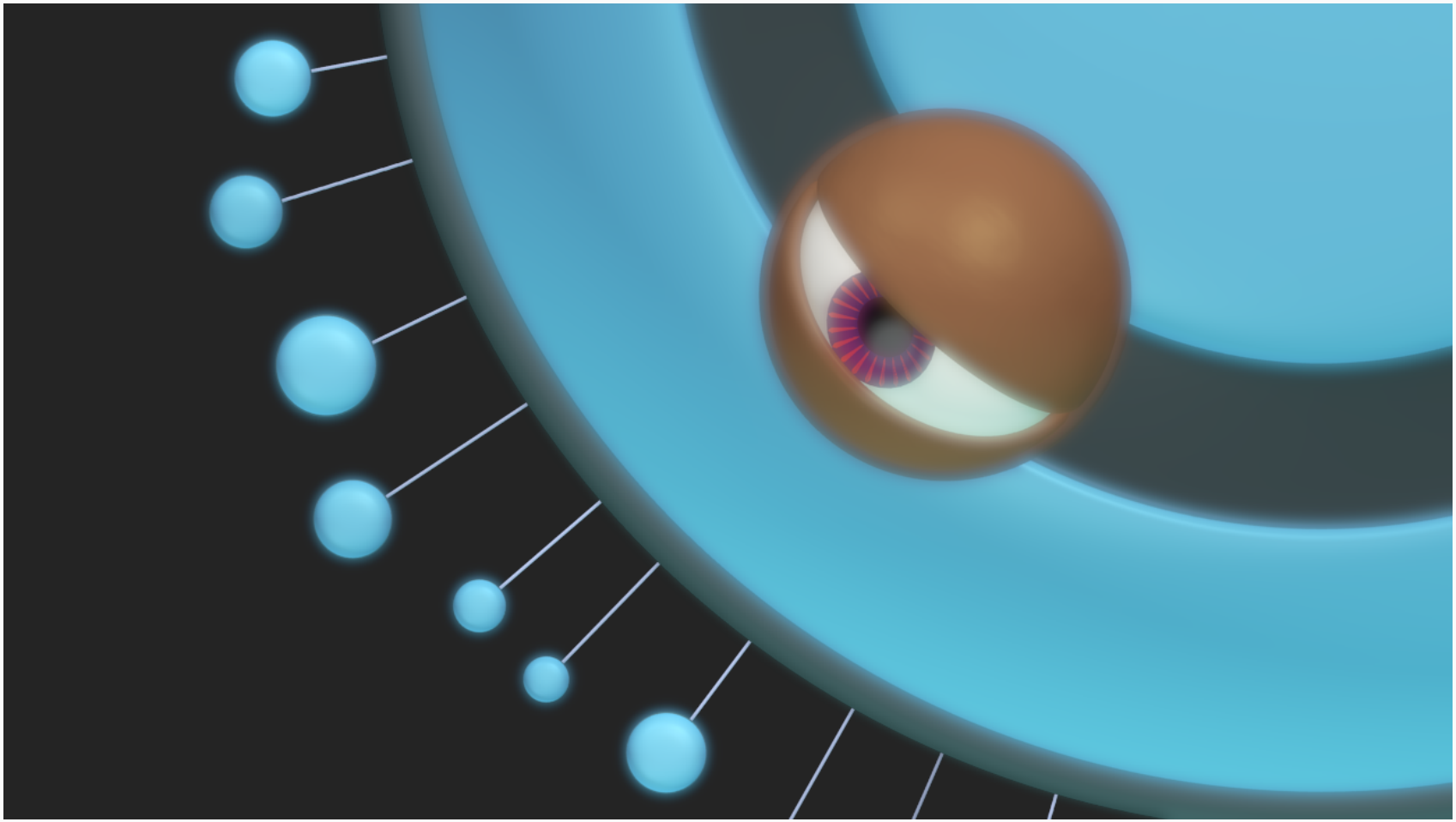
The sad reality (centralization)



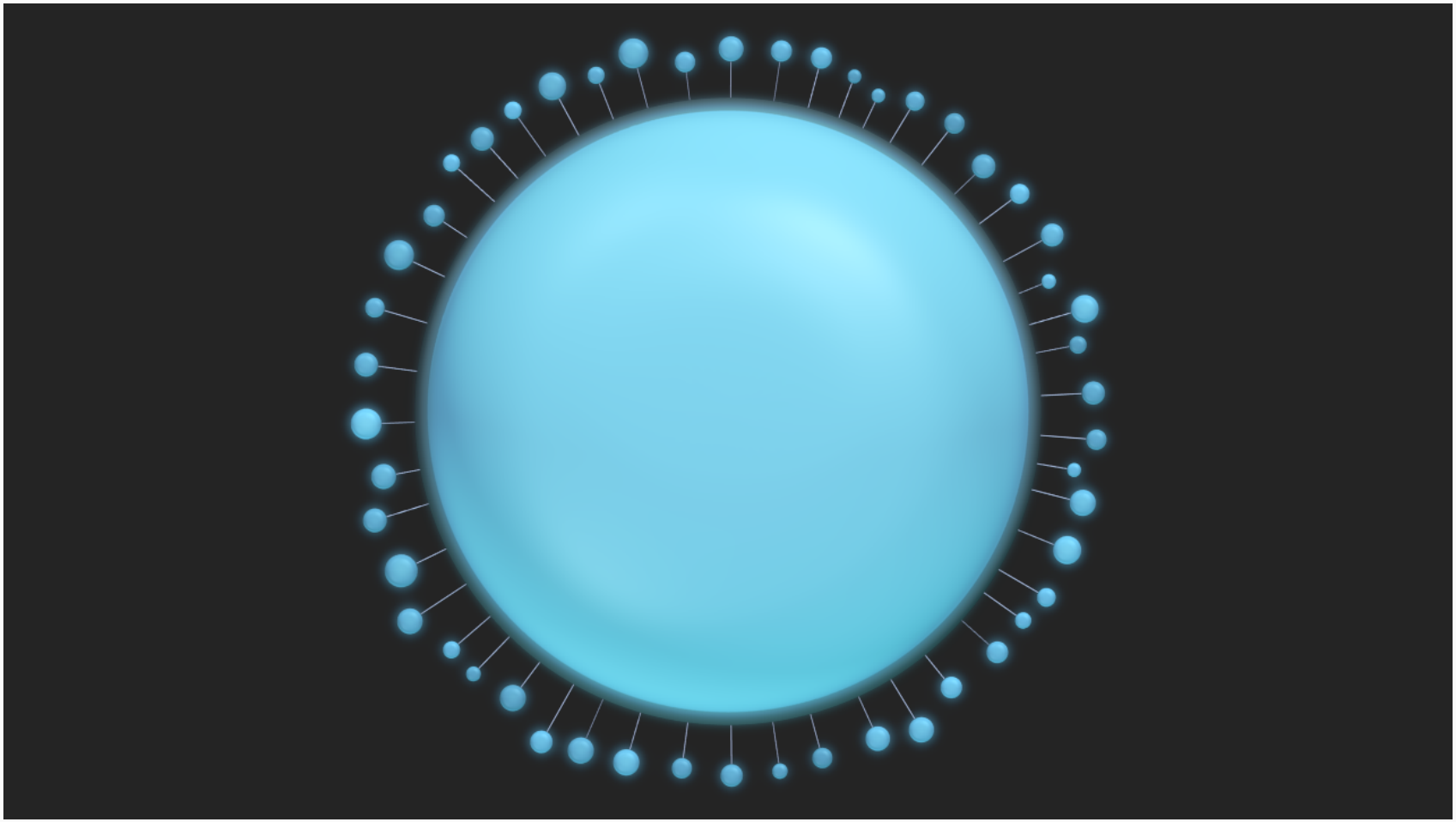
The sad reality (censorship)



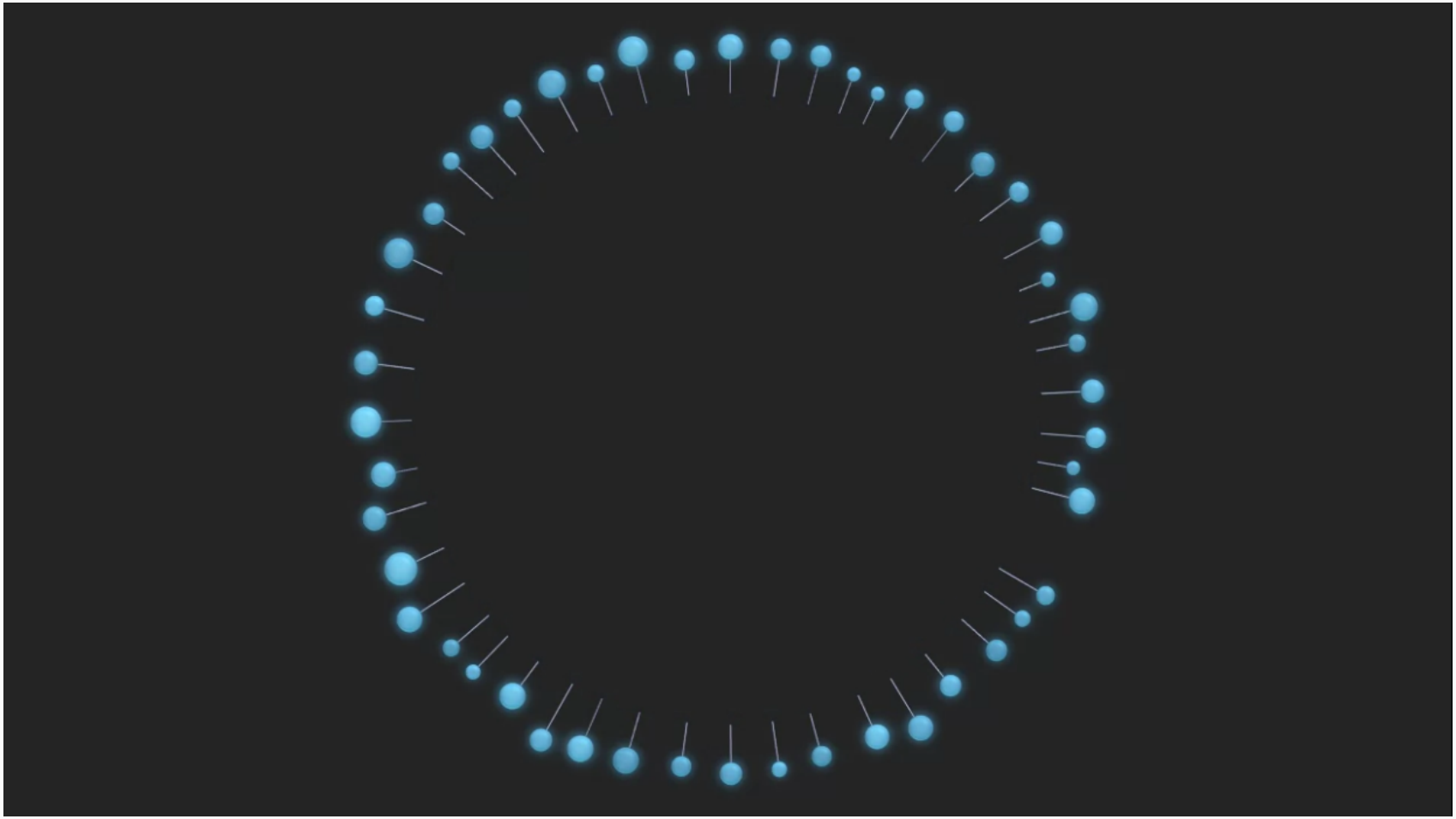
The sad reality (surveillance)



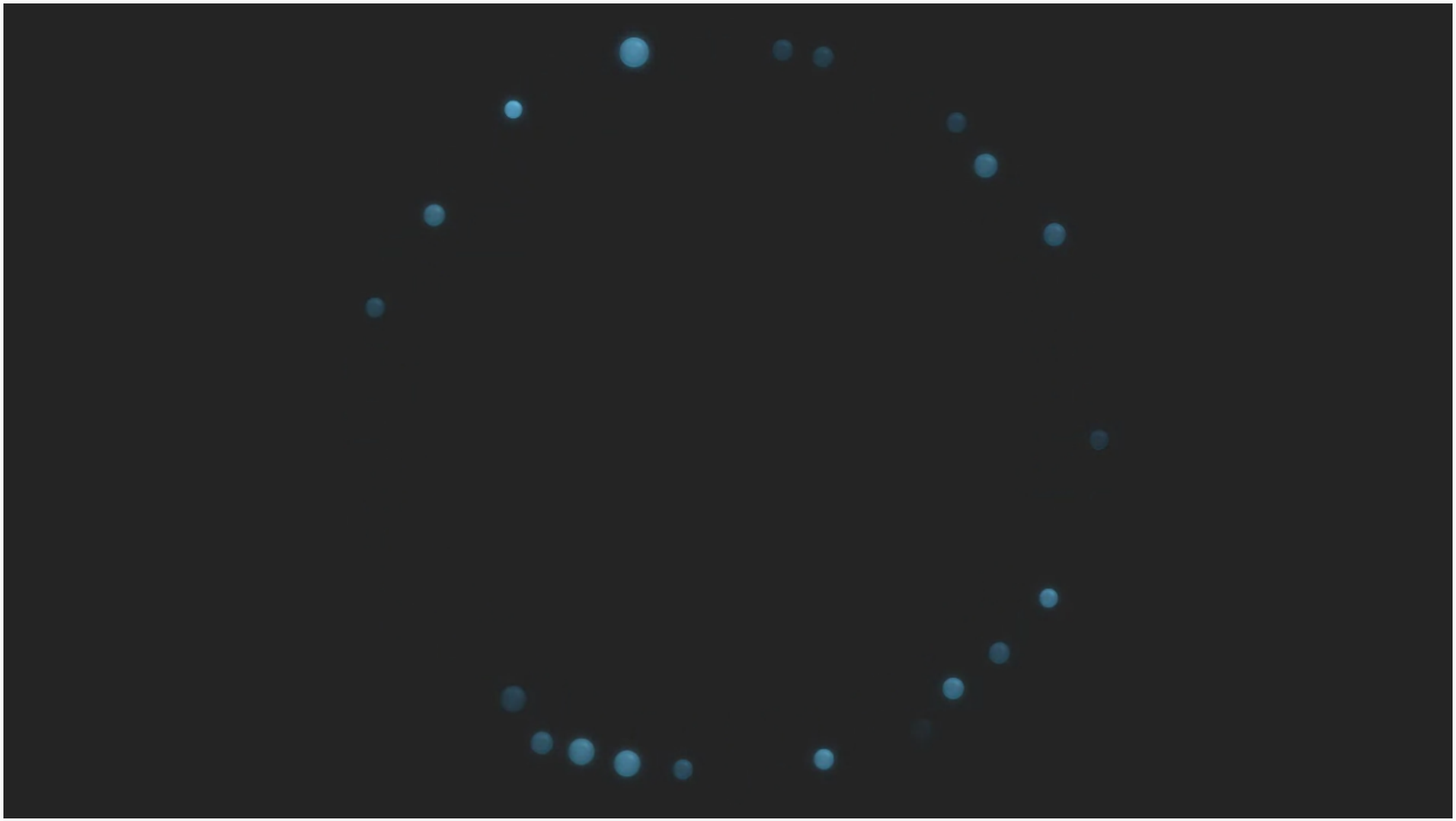
The sad reality (fragility)



The sad reality (fragility)



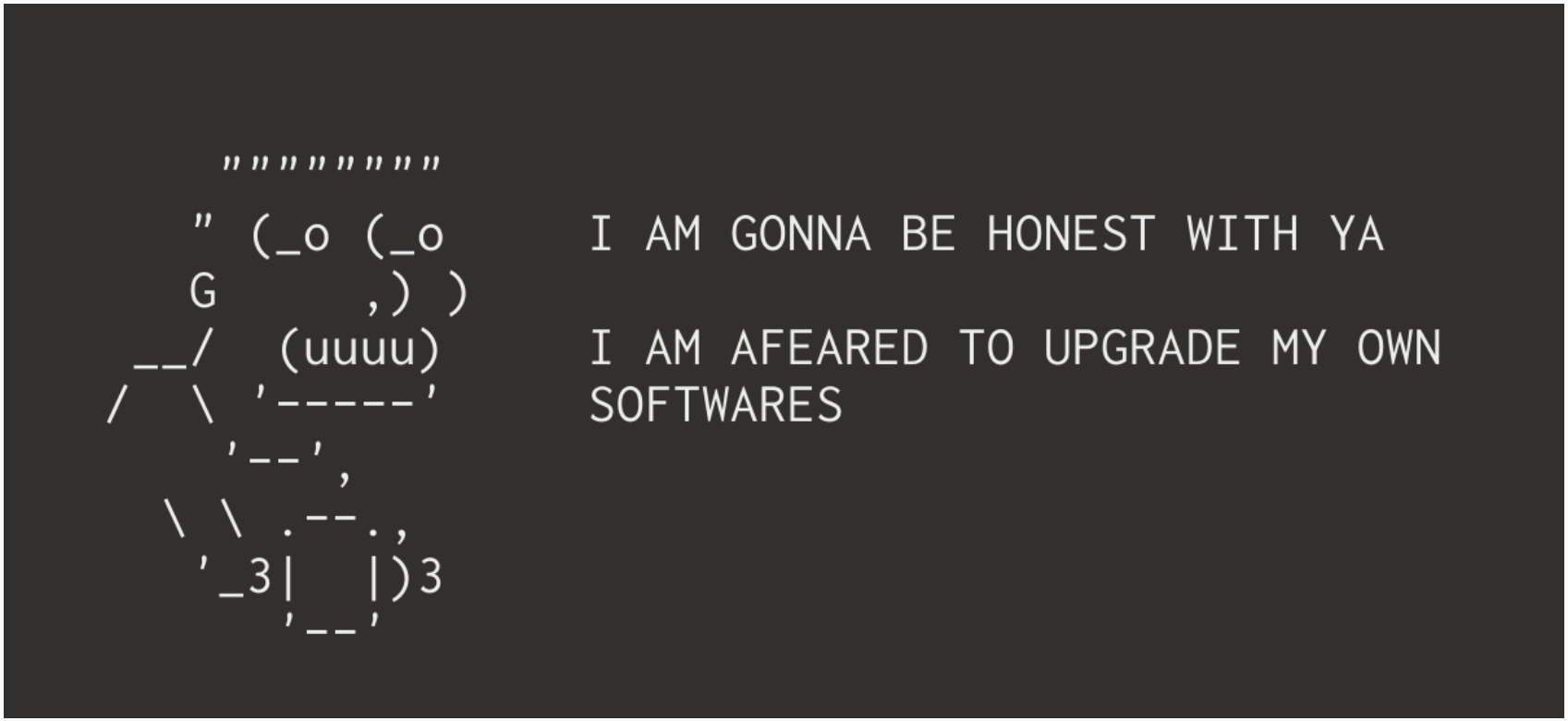
The sad reality (fragility)



The sad reality (fragility)

Sounds great! But???

Stuff is complex to run



Dependent on phase of the moon

One Language Package Manager Per Child

Have fun managing configuration

So... docker??? (Or something like it?)

Easy for users! “I already built this for you, just pull it down and use it!”

Maybe not :(

Distro-sized static compiling considered hazardous

- Extremely heavy: throws away dynamic linking
- Hard to introspect, rebuild
- Analysis of Docker Hub: over 70% have medium vulnerabilities, 30-40% high (shellshock, heartbleed) vulnerabilities [1]
- Reproducible? Not in the sense of `https://reproducible-builds.org/`
- Docker's DSL is not expressive
- Still dependent on "phase of the moon" of distributions!

[1] `http://www.banyanops.com/blog/analyzing-docker-hub/`

Unfortunately, it's not just Docker

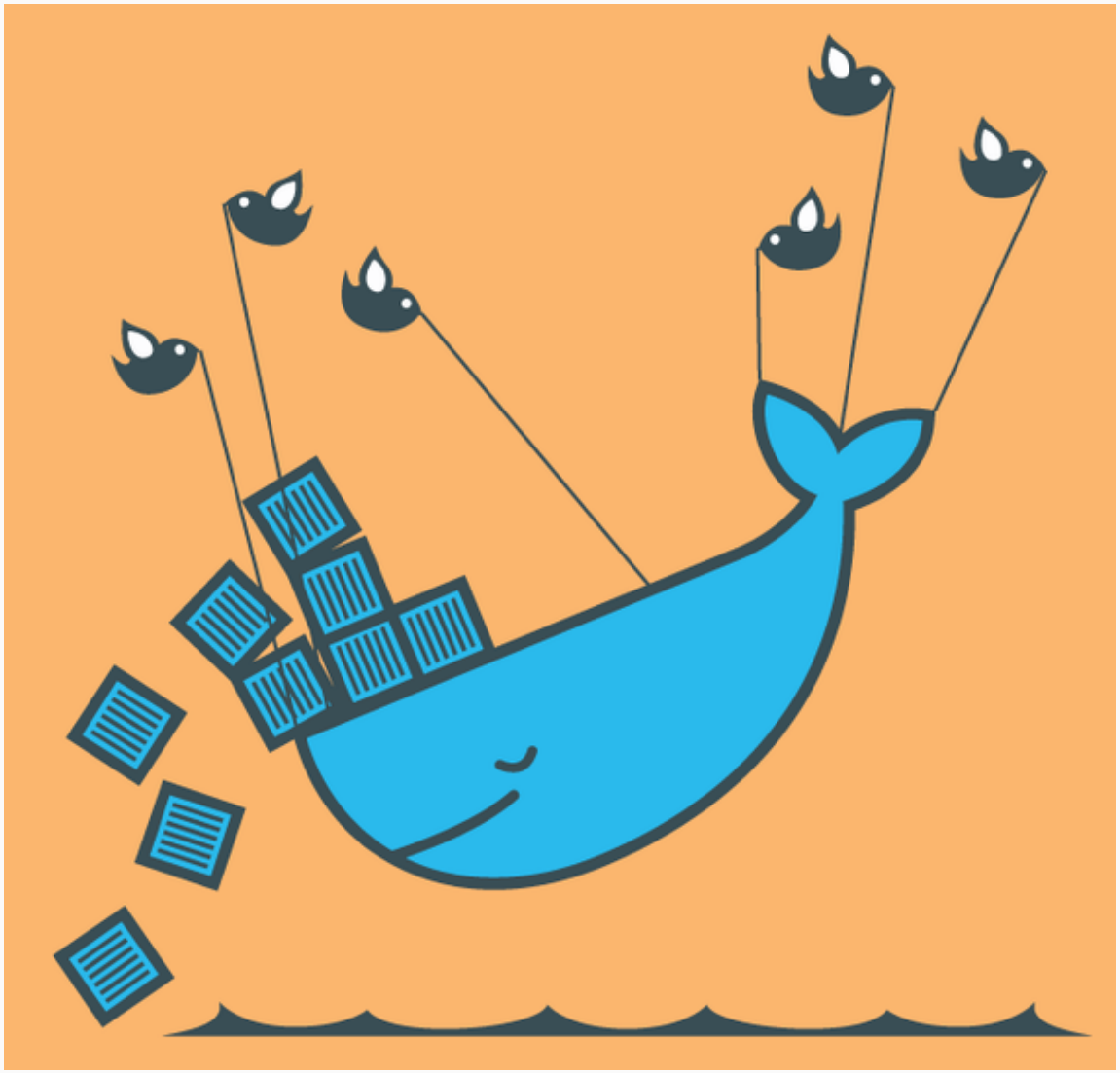
- CoreOS
- xdg-app
- Qubes
- Snappy
- egads!

We are losing the ability to reason about free software!

A policy issue disguised as a technical issue?

If it's too hard to build, run and modify software, what does this this mean for user freedom?

And so here we are

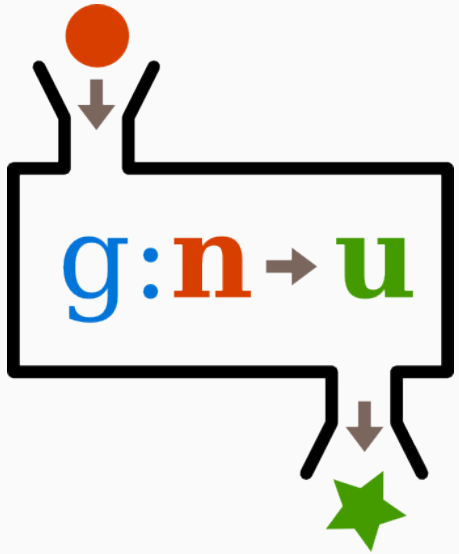


Enter Guix & GuixSD!

Enter Guix & GuixSD!

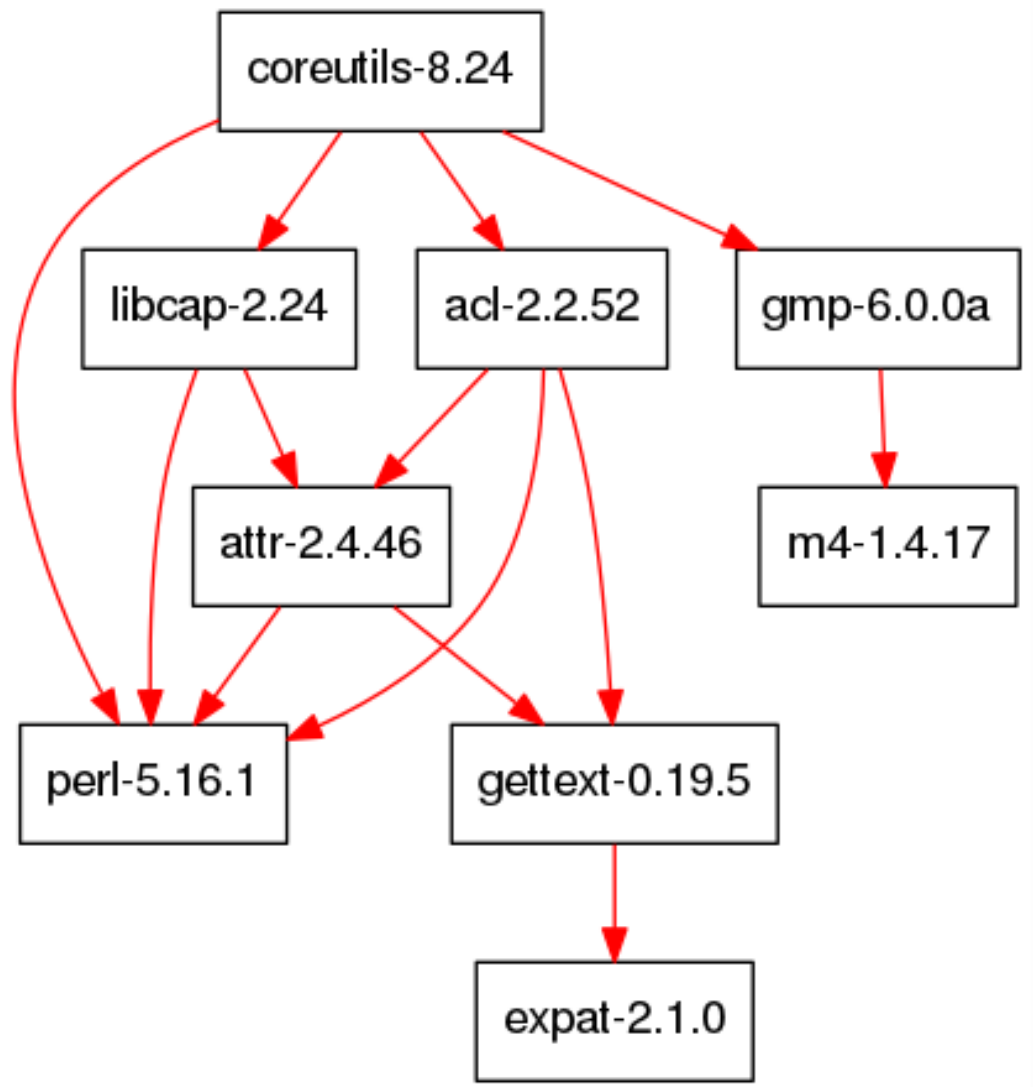
GuixSD

Functional package management



- Precisely specified dependencies
- Reproducible builds (same input, same output)
- Built packages are immutable
- Unprivileged package management
- Atomic upgrades and roll backs

Packages capture the full dependency graph



Your profile, my profile

- Unpriviliged package management (no root privileges needed!)
- Each user may install software without conflicting with other users or the system
- Each user may have many profiles

Guix gives users practical freedom to use different sets of packages for different tasks.

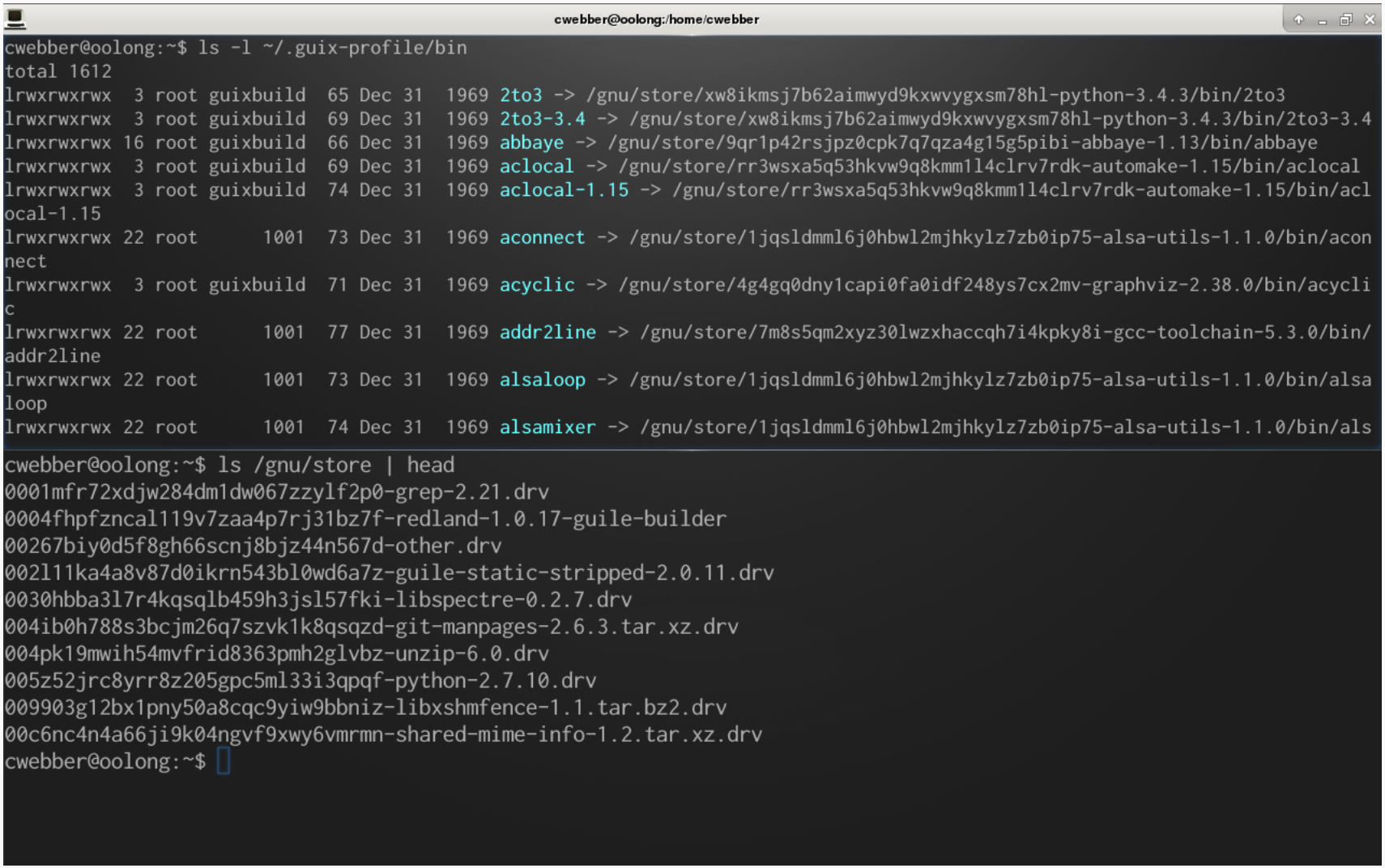
Transactional upgrades and rollbacks

Bad upgrade? No problem!

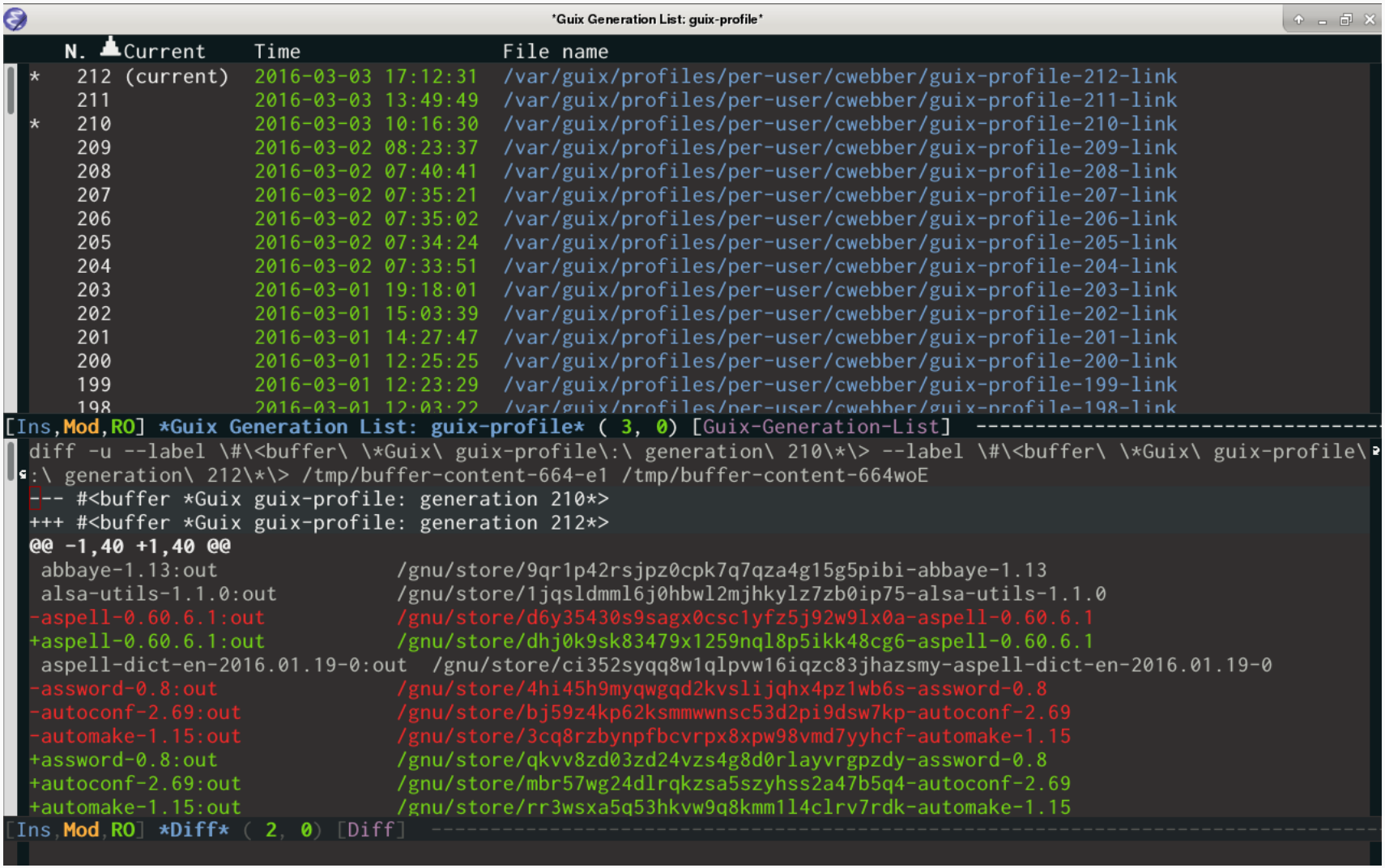
```
guix package --roll-back
```

Congratulations, you are now a time wizard!

Like Git, for your operating system!



Keep the history until you don't need it



Development environments

The `guix environment` tool can be used to quickly create development environments.

Like Python's `virtualenv`, but for anything.

Development environments

```
$ which irb
which: no irb in (/run/current-system/profile/bin)
$ guix environment --ad-hoc ruby ruby-nokogiri
$ which irb
/gnu/store/q2ldaivsnfdmvlxnc7hlw5skc9f9xw5g-profile/bin/irb
$ irb
irb(main):001:0> require 'nokogiri'
=> true
irb(main):002:0> Nokogiri
=> Nokogiri
```

Guix is written in Scheme

grep.scm

```
(define-public grep
  (package
    (name "grep")
    (version "2.21")
    (source
     (origin
       (method url-fetch)
       (uri (string-append "mirror://gnu/grep/grep-")
            version ".tar.xz")
       (sha256
        (base32
         "1pp5n15qwxrw1pibwjhhgsibyv5cafhamf8lwzjygs6y00fa2i2j"))
       (patches (list (search-patch "grep-CVE-2015-1345.patch")))))
    (build-system gnu-build-system)
    (synopsis "Print lines matching a pattern")
    (description
     "grep is a tool for finding text inside files.  Text is found by
matcing a pattern provided by the user in one or many files.  The pattern
may e provided as a basic or extended regular expression, or as fixed
strigs.  By default, the matching text is simply printed to the screen,
howeer the output can be greatly customized to include, for example, line
numbrs.  GNU grep offers many extensions over the standard utility,
inclding, for example, recursive directory searching.")
    (license gpl3+)
    (home-page "http://www.gnu.org/software/grep/")))
```

[Ins] grep.scm (26, 0) [Scheme] Git-master

Guix is a library

All data structures, procedures, etc. are exposed as Guile APIs.

Using these APIs, we've implemented:

- Declarative config management (Puppet, Chef, Salt, Ansible, etc.)
- Universal language packaging (PyPI, RubyGems, ELPA, CRAN, etc.)
- Local dev environments (virtualenv, rvm, rbenv, nvm, etc.)
- Local VM creation (Vagrant)
- Linux containers (Docker, rkt, lxc, etc.)

The world is yours to hack!

Full system configuration management

```
(operating-system
  (host-name "izanagi")
  (timezone "America/New_York")
  (locale "en_US.UTF-8")
  (bootloader (grub-configuration (device "/dev/sda")))
  (file-systems (cons (file-system
                        (device "root")
                        (title 'label)
                        (mount-point "/")
                        (type "ext4"))
                      %base-file-systems))
  (users (list (user-account
                (name "dave")
                (comment "David Thompson")
                (group "users")
                (supplementary-groups '("wheel" "netdev" "audio"
                                        "video" "cdrom"))
                (home-directory "/home/dave"))))
  (packages (cons* adwaita-icon-theme avahi gnome-terminal
                   htop less man-db ncurses nss-certs
                   openssh wicd unzip rsync xfce
                   %base-packages))
  (services %desktop-services)
  (name-service-switch %mdns-host-lookup-nss))

U:@---  izanagi.scm    Bot (52,0)    Git:master  (Scheme Guix Guile/A Paredit Proj
```

Package importers

```
$ guix import pypi pyglet
(package
  (name "python-pyglet")
  (version "1.2.4")
  (source
   (origin
     (method url-fetch)
     (uri (pypi-uri "pyglet" version))
     (sha256
      (base32
       "0i9la03pm51swv2z8f17bx7qz2yjfxg6hn7i9c42s81bryxzyqlz"))))
  (build-system python-build-system)
  (inputs
   `(("python-setuptools" ,python-setuptools)))
  (home-page
   "http://pyglet.readthedocs.org/en/pyglet-1.2-maintenance/")
  (synopsis
   "Cross-platform windowing and multimedia library")
  (description
   "Cross-platform windowing and multimedia library")
  (license bsd-3))
```

What's in `(/gnu/)store`?

guix deploy

- Remote cluster management tool
- Provision bare metal, VMs, containers
- Complete replacement for Chef, Puppet, Ansible, Salt, etc.

Recipe based deployments? With a GUI?

- Select pre-built recipes for MediaGoblin, Wordpress, etc.
- “Don’t repeat yourself“ configuration
- Could we make a simple web interface for this?

Sandboxed applications

- Why should Solitaire have access to your GPG/SSH keys?
- Limit web browser's access to your home directory

Easy installs?

- Graphical distro installer
- Graphical interface for:
  - system configuration
  - user-specific packaging
  - easy roll-backs

Support for difficult languages that web developers use

NodeJS, we're looking at you! [2]

Also, Java. [3]

Send help.

[2] http://dustycloud.org/blog/javascript-packaging-dystopia/

[3] http://lists.gnu.org/archive/html/guix-devel/2016-02/msg01396.html

Wrapping up

State of Guix

- “It’s still beta!”
- But probably more stable than most devops stuff
- A delight to run (we use it!)
- Easy to develop and get involved in
- ≈3350 packages
- ≈20-30 contributors per month and increasing

There's more to do (and you can help!)

- Test drive Guix or GuixSD and send feedback
- Write GuixSD system services
- Improve documentation
- Improve containers, add packages, add importers, ...

Talk with the developers via IRC at `#guix` on Freenode or via email at `guix-devel@gnu.org` or `help-guix@gnu.org`

A short story

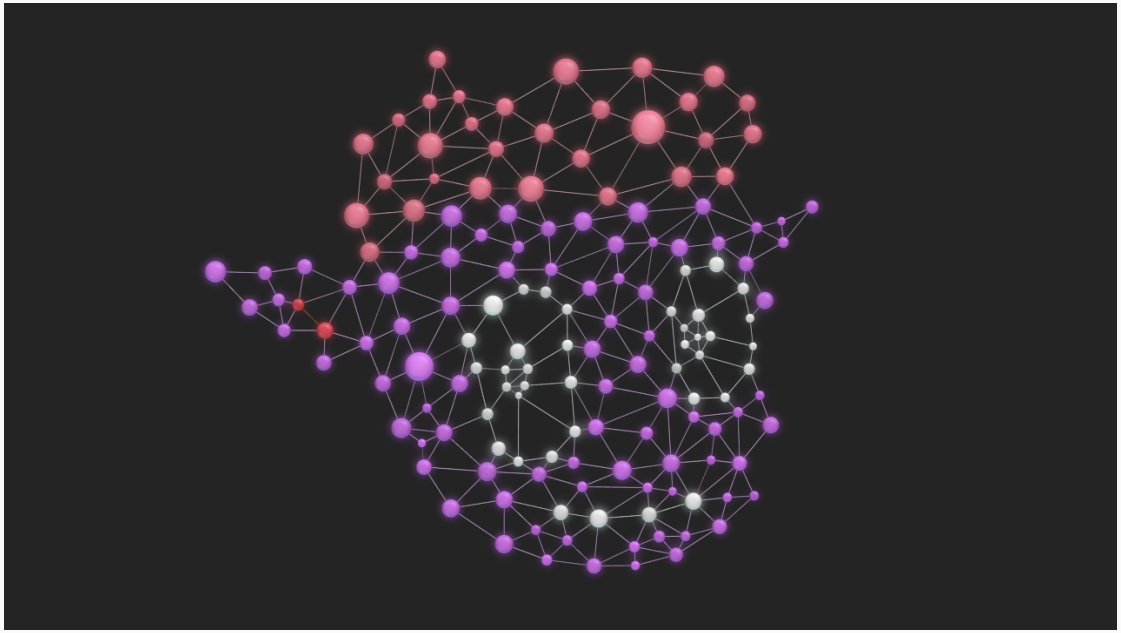
Credits (p.1)

- Moon image from *Le Voyage dans la lune* (A Trip to the Moon), public domain. Retrieved from: `https://en.wikipedia.org/wiki/A_Trip_to_the_Moon#/media/File:Le_Voyage_dans_la_lune.jpg`
- Tar pit image by Ray Bouknight, CC BY 2.0: `https://www.flickr.com/photos/raybouk/8341369957`
- Rena runs aground (container fail image), CC BY 2.0 `https://www.flickr.com/photos/nzdefenceforce/6386334175/`

Credits (p.2)

- Caminandes video screenshot by Blender Institute, CC BY 4.0 http://www.caminandes.com/
- Chemical warehouse image from Pixabay, CC0 https://pixabay.com/en/warehouse-chemistry-industry-629641/
- GuixSD logos by Luis Felipe López Acevedo, CC BY-SA 4.0 http://www.gnu.org/software/guix/graphics/
- Docker + Twitter image by Karen Rustad
- Slight snippet from Guix (grep package), GPLv3 or later

Thanks! Questions?



More on Guix: `https://gnu.org/software/guix`