Open source: licences, practices, & tools

Roman Klapaukh

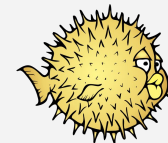Quaisr

If you need licence advice consult a legal service

Open source is cool

Many things are not open source
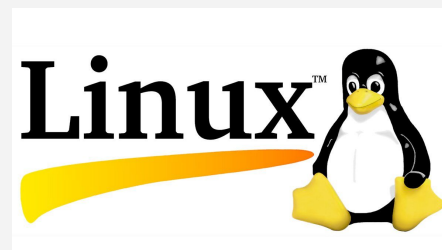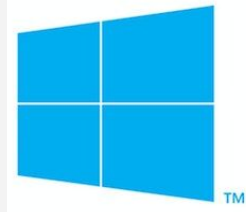
What actually is open source?

- Does it have to be free?

- Does it have to exist online?

- Can you be paid to develop it?

- Is it crowdsourced?

- Is it stuff that's on Github?

What actually is open source?

It varies.

In different contexts it means different things.

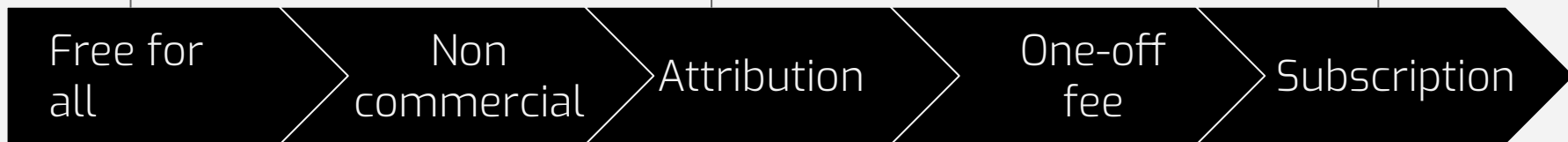# What to do with source code?

How can I distribute my software?

Here is all my code and everything. Use it however you like.

Here is my source code. But remember that I wrote it.

Pay me regularly to continue using the binary

Free for all

Non commercial

Attribution

One-off fee

Subscription

Use it how you like, as long as you don't make money

Buy the binary to keep. Probably pay for updates

Default copyright

For the lifetime of the author plus 70 years

Terms

Copyright - protects a produced work

Trademark - protects an identity, e.g., name / slogan / brand

Patent - protects a novel non-obvious invention

Permissive licenses

Some popular permissive licence families are:

- MIT license
- BSD license
- Apache license

Public domain is the most permissive, but rarely used.

While the details vary these licenses tend to focus on two things:

- Protecting the author from users of their code
- Preserving some notion of the author being the author (though not always)

Copyleft licenses

Weak Copyleft

- Lesser General Public license
- Eclipse public license
- Mozilla public license

These licenses tend to focus on keeping modifications to code open source while still allowing it to be used (as a whole) in non open source applications.

Copyleft license

- General Public License
- Affero General Public License

These licenses tend to focus on ensuring derivatives of licenced code remain open source.

Creative Commons

The Creative Commons are a set of licenses typically for artistic works.

Permissions are combinations of:

- None (public domain)
- Attribution
- Share-alike
- Non-commercial
- No-Derivatives

The Creative Commons recommend not using these for software.

Proprietary Licenses

You don't have to grant people access to your source code at all. Most proprietary licenses are only for binaries.

Updates are often included in subscriptions, but typically limited in the case of one off purchases.

These licenses typically focus on commercial considerations, including potential revenue, cost of distribution, accepted norms, etc.

Example licensing story

Sam, working alone, writes a library for drawing sheep pictures.

She decides to put it up online for other people to use, but wants to minimise the chance that someone can make money from it. So she licenses it under the GPL.

How might it be used?

What about the art that Sam drew as assets for her tool?

Photo: https://flickr.com/photos/stevepj2009/ CC-BY

Dual licensing

Sam has her library up on Github with a GPL licence.

A large company approach her and want to buy a licence to use the library without needing to GPL their code.

What to do? What to do?

Contributor agreements

Since Sam's library is open source, other developers start to contribute code to her project.

Eventually she decides she wants to make the project more open - change to a BSD licence.

But who owns the code?

Contributor licence agreements (or sometimes copyright transfer agreements) are used to allow the owner of a repository to keep their decision making powers over it.

These can be subject to different constraints depending on the organisation.

Git and Github

- Git is GPL
- GitHub is proprietary
- How does that work?

Contact us

Roma Klapaukh
roma@quaisr.io

@quaisrlabs

https://quaisr.io

SimOps for engineering

Icons by Icons8