Europeana Space has received funding from the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, under GA n° 621037

10

Content Space

Software Open Source Licence Chooser

When content has already been opened up and an institution is considering making it available through some kind of software application, it will be necessary to think about the license requirements for this application. There have been examples of museums that let an external developer create an app for one of their exhibitions, forgetting to discuss with the developer, what kind of license this app could be used and re-used under in the future. The consequence of not discussing this might be that such an institution is stuck with a product they are not allowed to modify, with a source code that they cannot access, and are thus locked in by the supplier.

If considerable efforts have already been undertaken to make the content easily accessible, similar measure could be taken with the software ordered. Institutions may be concerned that developers will fear losing business opportunities, or be concerned that their developer name will no longer be associated with what they have created. However, such problems can be solved by attributing the right kind of license to software.

Initially it is important to discuss whether or not your supplier is willing to release the ordered piece of software as an open source product. This means that the created product can be freely used, changed, and shared (in modified or unmodified form) by anyone. Open source software is often made by many people, and distributed under licenses that comply with the Open Source Definition[1]. One of the aspects of this definition, is that the actual product must include source code, and must allow distribution in source code as well as compiled (complete, e.g. the actual app) form. Where some form of a product is not distributed with source code, there must be a

1. See The Open Source Initiative for this definition of Open Source Software: http://opensource.org/

well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge (thus, the code could be requested from the supplier via e-mail or an online form).

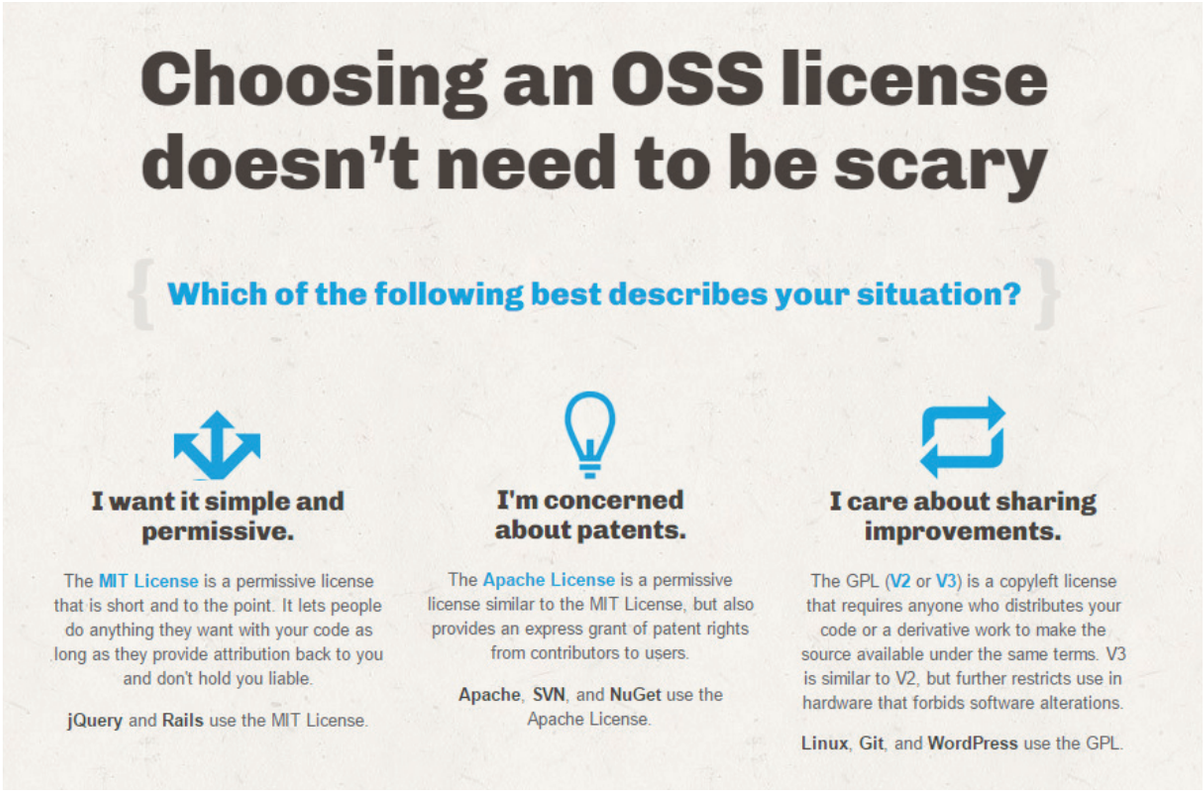
What this enables the subcontracting institution to do, if they have IT-skilled staff, is to enhance the product without having to pay an extra fee to actually obtain the source code. This way, in the case of a museum, for example, it could re-use the app with a bit of customisation for another exhibition, or a whole other purpose. With access to the code, anything is possible. Releasing it as open could also bring on other effects; people with an interest in the program might think of nice add-ons or enhancements. They might even contribute to the software in a way that the content holder had previously imagined. This approach may, therefore, mean work and time saved, a better product, and again, one which is shareable with a community that is much broader than the original institution alone.

If a supplier agrees with releasing the tendered program as open source software, then as with publishing the content, she will have to choose an appropriate open source license for her product. Two main strands can be identified:

- Permissive license types: these only describe minimal requirements about how the software can be redistributed. Such licenses therefore make no guarantee that future generations of the software will remain free: if the intention is to re-use this licensed code in another programme and make that product proprietary, this can be freely undertaken. Examples of permissive free software licences are the MIT License and the BSD licenses.
- Copyleft license types: these are more 'share-alike' in nature. When a program is released that is based on or uses copyleft licensed software, it will have to be made available on terms no more restrictive than the copyleft license of the software originally used. It will thus be harder to make a product proprietary, if a copyleft component has been used. Another difference between permissive and copyleft, is that when the software is being redistributed (either modified or unmodified), permissive licences permit the redistributor to restrict access to the modified source code, while copyleft licenses to do not allow this restriction. An example of a copyleft licence is the GNU General Public License[2].

2. More info on the different types of licensing can be found on http://en.wikipedia.org/wiki/Permissive_free_software_licence

Between all available licenses in each category there are many options of choice. Some guidance is definitely useful. GitHub, the biggest code repository, also understood this: “It’s easy to get caught up in code. Sharing your code isn’t everything, though: it’s also important to tell people how they can use that code”[3]. They created ChooseALicense.com to help developers make an informed choice. The website shows a breakdown of what is required, what is permitted, and what is forbidden for each license[4].



Most of the open source software released by GitHub has been placed under the MIT license. It is a popular permissive license for a number of reasons, among these[5]:

- Its license text is short: anyone can read and understand exactly what it means without needing a legal background.
- Enough protection is offered to be relatively sure there will not be any claims if something goes wrong when another developer uses your code (or part).

---

3. See https://github.com/blog/1530-choosing-an-open-source-license

4. For content licensing, Creative Commons provides a similar simple wizard to help content authors select an appropriate CC license: http://creativecommons.org/choose

5. See http://tom.preston-werner.com/2011/11/22/open-source-everything.html

So although OS licenses do allow for a great deal of freedom, they are not the same as releasing a work into the public domain. Permissive licences often do stipulate some limited requirements, such as that the original authors must be credited (attribution). If a work is truly in the public domain, this is usually not legally required. Attribution may still be considered an ethical requirement. Continued proper attribution is also one of the things that MIT licenses also require – just like a CC-BY.

```
/*!
 * Copyright 2014 Drifty Co.
 * http://drifty.com/
 *
 * Ionic, v0.9.23-alpha
 * A powerful HTML5 mobile app framework.
 * http://ionicframework.com/
 *
 * By @maxlynch, @helloimben, @adamdbradley <3
 *
 * Licensed under the MIT license. Please see LICENSE for more information.
 *
 */;

// Create namespaces
window.ionic = {
  controllers: {},
  views: {},
  version: '0.9.23-alpha'
};
(function(ionic) {

  var bezierCoord = function (x,y) {
    if(!x) x=0;
    if(!y) y=0;
    return {x: x, y: y};
  };

  function B1(t) { return t*t*t; }
  function B2(t) { return 3*t*t*(1-t); }
  function B3(t) { return 3*t*(1-t)*(1-t); }
  function B4(t) { return (1-t)*(1-t)*(1-t); }
```

Example of copyright info in MIT-licensed code, see top

Anyone building upon this programme should keep the copyright information intact and add its credit line. Other than it being a license requirement, it is also a matter of courtesy and ethics. This way, even though the code has been opened up for anyone to re-use, they will still see who made it or contributed to it.

In recent years, the European Commission have been enthusiastic in the support of open licensing. They released the EUPL or European Union Public License[6]. The EUPL was first intended to distribute the EC's own software. However, here would

6. For more information on the EUPL, see https://joinup.ec.europa.eu/community/eupl/og_page/eupl

then be no direct benefit of creating a new OS license, when several exist already. The EC had some specific requirements, currently not covered by the existing ones[7]:

- The licence should have equal legal value in many languages;
- The terminology regarding intellectual property rights had to be conformant with European law requirements;
- To be valid in all Member States, limitations of liability or warranty had to be precise, and not formulated "to the extent allowed by the law" as in most licences designed with the legal environment of the United States in mind;
- In addition, distribution of software should avoid the exclusive appropriation of the software even after improvement by a third party (therefore, the EUPL is a "copyleft" licence).

Can such openness then still be associated with using an OS program for commercial purposes? Yes; all Open Source software can be used for commercial purpose. It can even be sold. Services can be sold based on the code, or tailored customisation and maintenance work can be offered[8]. There are also other ways this would impact the business potential of an open source project. With adequate communication about the product created, and uptake by interested users and contributors, it might be possible to form a community around the program. It might become known as (or in part) a standard piece of code, gain in visibility and increase the long term sustainability of the work. Ideally, an institution could build a commercial service ecosystem around it.

7. See https://joinup.ec.europa.eu/software/page/eupl/introduction-eupl-licence

8. See http://opensource.org/faq#profit and http://opensource.org/faq#commercial