

# Software Licences

## HSF Recommendations

John Harvey / CERN

24 June 2015

---

# Summary for iFB meeting

- ❑ A Technical Note (TN) is being drafted that aims to give background information and make recommendations to help developers select a suitable software license for their project
- ❑ The current version draws heavily on the work done by a Task Force at CERN that met over a period of >12 months and issued a report containing guidelines for CERN projects in Jan 2012
  - 3 CERN lawyers participated in this work
- ❑ The draft has been circulated to the Startup team for comment
- ❑ These slides constitute an executive summary
- ❑ The full note can be found here
  - <http://hepsoftwarefoundation.org/content/technical-notes>
- ❑ It would be appropriate to start a WG of those interested in contributing to the TN

# Disclaimer

- ❑ The draft has been circulated to the HSF Startup Team and has provoked quite some reaction, often not agreeing with the formulation of the recommendations I will describe
  - this is *work in progress*
- ❑ The report will undoubtedly be revised and integrate new contributions in the coming weeks – your input is welcome

# Principle of openness

- Philosophy of openness is enshrined throughout our field as exemplified by making the results of our experimental and theoretical work generally available
- The goal of the HSF is to facilitate development of reliable and long-lived software products through collaborative open-source software development.
  - **Open-Source Software (OSS)** *is computer software with its source code made available with a licence in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose.*
- **Recommendation** : Unless forbidden due to external constraints, such as collaboration agreements, software should always be made available as **open-source**

# Safeguarding intellectual property

- ❑ **Copyright** grants the creator exclusive rights to its use and distribution. It is a form of *intellectual property*, and is often shared among multiple authors, each of whom holds a set of rights to use or licence the work. These rights frequently include reproduction, control over derivative works, distribution, and "moral rights" such as attribution.
- ❑ **Recommendation** : Software must contain in the notice a statement acknowledging the copyright owner(s) and license

# Software License

- A typical software licence grants an end-user permission to use one or more copies of software in ways where such a use would otherwise potentially constitute copyright infringement of the software owner's exclusive rights under copyright law.
- **Recommendation** : The open-source licences used for software “hosted” by the HSF should be widely used licences approved by the **Open Source Initiative (OSI)**

# Why OSI-approved licenses?

- There are ~70 OSI-approved-licenses in total and it is a stated goal of the OSI to avoid proliferation of new licenses
  - 2001: CERN submitted its own (permissive) license for EDG project that was approved by OSI
  - 2004: CERN submitted a new license for EGEE project which was **not** approved (Apache v2 was used instead)
- Legal experts are reluctant to analyze non-OSI-approved licenses
- The 9 widely-used licenses approved by the OSI are:
  - Apache-2.0
  - BSD-3-Clause, BSD-2-Clause
  - GNU General Public license (GPL), GNU Lesser GPL (LGPL)
  - MIT
  - Mozilla Public License 2 (MPL 2.0)
  - Common Development and Distribution License 2 (CDDL-1.0)
  - Eclipse Public license (EPL-1.0)

# Strong Copyleft Licenses

- ❑ Strong Copyleft is a general method for making a program free, and requiring all modified and extended versions of the program to be free as well
- ❑ Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it.
- ❑ The spirit behind a Copyleft licence is the creation of an open community of users or developers where the licencees are encouraged not only to improve, correct, complement and integrate the software they receive but also to make available these enhancements to the entire community
- ❑ A good example is the GNU General Public License (GPL).



# Weak Copyleft License

- ❑ This is similar to Copyleft except that the user may include, unmodified, the open source software in a larger programme and release the larger work under a licence different from the initial open source licence.
- ❑ In other words, the user is not obliged to provide the full source code of its larger work
- ❑ A good example is the GNU Lesser General Public License (LGPL)
- ❑ This targets libraries of software, which are designed to be incorporated unchanged into larger programs.
- ❑ It is frequently used when there is concern that the obligation to release the source of a work incorporating unchanged the GPL-licensed software would seriously hamper its wide adoption (e.g. GNU C library, ROOT)

# Permissive Licenses

- ❑ These licences allow the distribution under a different licence of any work incorporating the Open Source Software as well as of any modified versions or any work incorporating modified versions. This different licence may be a proprietary licence.
- ❑ They do not require a derivative work of the software, or modifications to the original, to be distributed using the same licence (unlike copyleft licences)
- ❑ Good examples are Apache 2.0, two variants of BSD, and MIT

# License usage in open source projects\*

- GPL – 35%
- LGPL – 7%
- MIT – 20%
- Apache 2 – 16%
- BSD (3-clause) – 6%
- all others - < or = 2%

★ source :

<https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses>

# Further recommendations

- ❑ The first option is to use a strong copyleft license and the **recommendation** is to use the most recent version of GPL i.e. GPL v3
- ❑ A weak copyleft license can be used for program libraries and other cases where the primary objective is the rapid adoption of the software. For this LGPL v3 is **recommended**
- ❑ A permissive free software license can be used when constraints are imposed by existing agreements, such as those applied by external funding bodies. The **recommendation** is to use one of the widely used licenses i.e. Apache 2.0, BSD-2, BSD-3, MIT
- ❑ **Recommendation:** When contributing to an existing project, release your modified versions under the same license as the original work

# Commercial Exploitation

- ❑ Any software distributed under one license may also be distributed under one or more different licenses.
  - dual or multiple licensing
- ❑ A frequent use case is the public release of the software under a Copyleft license (GPL) and at the same time a bilateral agreement between the owner and a 3<sup>rd</sup> party for commercial exploitation of the software.
- ❑ Dual licensing is unnecessary in the case a permissive license is used

# Dealing with large collaborations

- ❑ Consideration can be given to creating a collaboration agreement that defines the choice of license and rules for governing the way decisions are taken
- ❑ In such cases long term maintenance can be more easily guaranteed if Copyright is transferred to one of the partner institutes, typically the prime distributor of the software.

# Draft Instructions for creating Notice

- ❑ The report also contains a suggested template for creating a notice that includes the Copyright statements and the license to be used
- ❑ Examples are given for different cases in which single or multiple developers and institutes are involved in the project