CHAPTER-2
OPEN-SOURCE PRINCIPLES AND METHODOLOGY

- Open-Source History, Open Source Initiatives,
- Open Standards Principles, Methodologies,
- Philosophy, Software freedom, Open-Source Software Development, Licenses,
- Copyright vs. Copy left, Patents,
- Zero marginal cost,
- Income-generation Opportunities,
- Internationalization.
- Licensing: What Is A License, How to create your own Licenses,
- Important FOSS Licenses (Apache, BSD, PL, LGPL),
- copyrights and copy lefts, Patent

Open-Source History, Open Source Initiatives

The 1950s – the A-2 system

In 1953, the A-2 system (an equivalent of today's compilers) was released together with its source code, and customers were asked to send any improvements to UNIVAC (the Universal Automatic Computer).

The 1980s – GNU

In 1983, however, Richard Stallman started to work on the GNU project, which was made up of rewrites of closed software he frequently used.

GNU stands for "GNU's Not Unix" and is pronounced as one syllable with a hard g.

The 1990s

In the 1990s, Linus Torvalds pushed OSS even further by creating the Linux kernel. He then released it to the public in 1991, along with its source code.

Principles for selecting open standards

- The government adopts open standards to use in government IT using the open standards for government data and technology process.
- Anyone can suggest an open standard for adoption and a central open standards secretariat maintains the open standards process.

There are 7 principles for selecting open standards for use in government.

1.Open standards must meet user needs.
2.Open standards must give suppliers equal access to government contracts.
3.Open standards must support flexibility and change.
4.Open standards must support sustainable cost.
5.Select open standards using well-informed decisions.
6.Select open standards using fair and transparent processes.
7.Specify and implement open standards using fair and transparent processes.

Copyright vs. Patent: What's the Difference?

- **Copyrights** cover artistic and intellectual works like books, songs, plays, and even computer software. **Patents** protect inventions and the way an item is used (utility patent) or how it looks (design patent).

- According to the U.S. Patent and Trade Office (USPTO), a patent grants an inventor the right to exclude others from making, using, offering for sale, or selling an invention. A copyright protects literary, musical, and other artistic works, whether it's published or not.

- Typically, copyright protection is filed by individuals or artists, but there are certainly business cases for copyright protection -- especially for companies looking to protect their business model or marketing ideas.

- Both patents and copyrights essentially prove that you are the creator of the item or idea and declare that no one can take it from you.

Why Are Copyrights and Patents Important?

- Copyrights and patents provide legal grounds for ownership and the right to pursue legal recourse if someone infringes on your idea. Otherwise, people can go around stealing ideas and creations and selling them as their own.

- It's hard to prove that one person had an idea over another, which is why most artists and businesses apply for protection the second they create something new.

Overview of copyleft

- Copyleft is a subset of open source. Contrary to what the term might imply, copyleft is not the opposite of **copyright**.
- In fact, copyleft is grounded in the concept of copyright, without which copyleft couldn't exist.
- Before someone can license software under a copyleft license, they must first **own the copyright** to that software, thus giving them the right to distribute it.
- Both open source and copyleft allow for source code to be modified and distributed. However, the difference is that with copyleft, the modified product must be distributed with the same copyleft license attached to the original software.
- This allows for creation of derivative work based on the source code covered by the copyleft license while protecting the original creator's interests.

For example,,,

Bob owns the copyright to original software, which he decides to distribute under a copyleft license. Gemma downloads Bob's software, modifies it, and then distributes her modified version.

Gemma must distribute her modified version under the same copyleft license that Bob used. Anyone who then modifies Gemma's version must also distribute it using Bob's same copyleft license.

Economics of Free and Open Source Software

Zero Marginal Cost

- At the core of the financial aspects of Free and Open Source is the zero negligible expense of merchandise in an environment that is digital.
- Right now, the rise of Free and Open Source speaks to an affirmation of old-style microeconomic value hypothesis - that a marginal cost in an ideal market approximates the minimal expense.
- From this point of view, Free and Open Source can be comprehended as a pioneer in arriving at what can be comprehended as a developmentally steady powerful Nash balance in a genuinely free market.
- Marginal cost is the term utilized in the study of financial aspects and business to allude to the increment in the actual development cost coming about because of delivering one extra unit of the product.

What is Internationalization?

- Internationalization is a corporate strategy that involves making products and services as adaptable as possible, so they can easily enter different national markets.

- This often requires the assistance of subject matter experts.

- Internationalization is sometimes shortened to "i18n", where 18 represents the number of characters in the word.

Licensing: What Is A License, How to create your own Licenses

- In most instances, software products require activating licenses or agreeing to "terms and conditions" before programs can be downloaded, installed, or accessed.

- **What Is a Software License?**

- A software license is a contract between the entity that created and supplied an application, underlying source code, or related product and its end user.

- The license is a text document designed to protect the intellectual property of the software developer and to limit any claims against them that may arise from its use.

- Most software falls under one of two categories that have distinct differences in how they are viewed under copyright law:

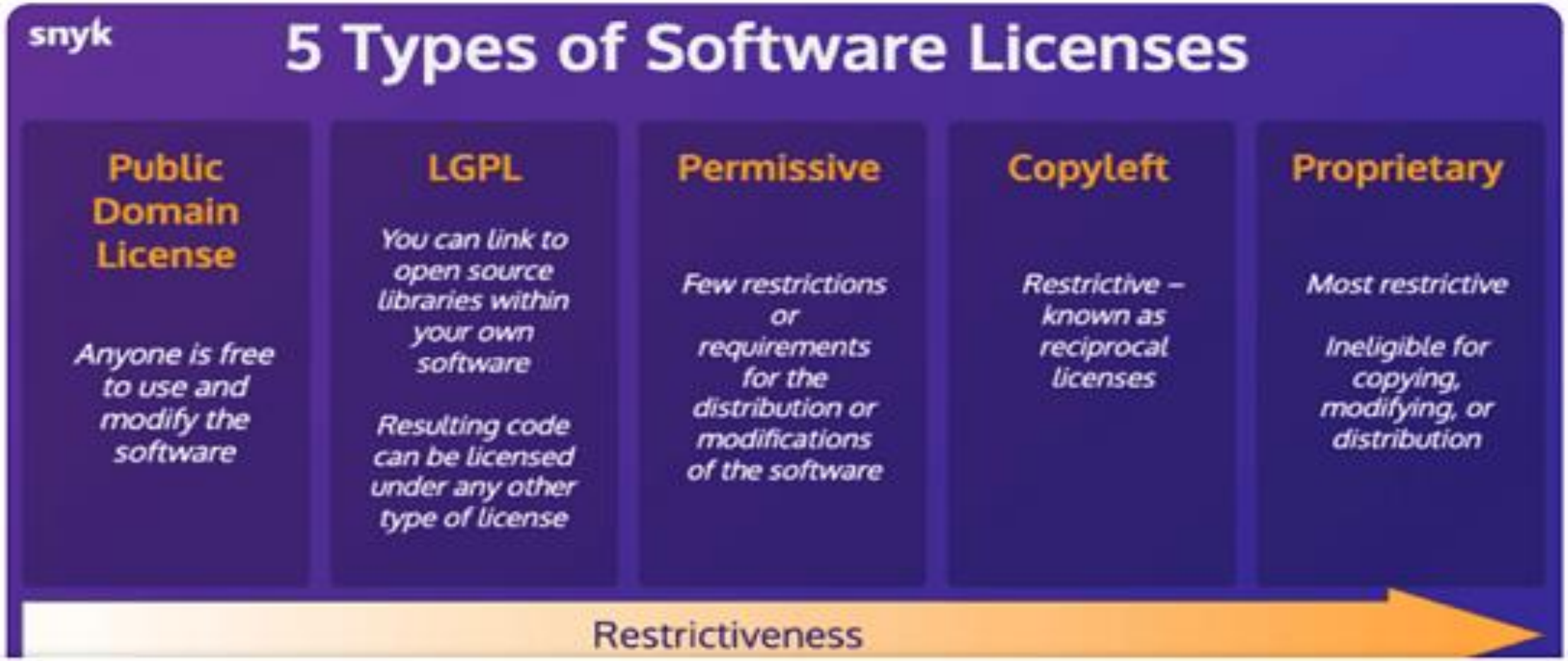
- **Proprietary**– also referred to as “closed source”

- **Free and open-source software (FOSS)** – referred to as “open source”

Important FOSS Licenses

- FOSS software licenses - give rights to the customer that include modification and reuse of the software code, providing the actual source code with the software product(s).
- This open-source type of licensing affords the user authority to modify the software functions and freedom to inspect the software code.
- **Proprietary software licenses** - provide no such authority for code modification or reuse and normally provide software with operational code only, and no source code.
- A proprietary software license often includes terms that prohibit “reverse engineering” of the object code with the intention of obtaining source code by the licensee.

5 Types of Software Licenses You Need to Know About



….THANK YOU….

Online Linux Terminal - unixterm (tutorialspoint.com)