BE-CSE-2 VII Semester- FOSS
Unit-1

Dr. M Swamy Das
Professor, CSE Dept., CBIT, Hyderabad

Unit-I

- Introduction to open-source software
- Need and principles of OSS
- Requirements of open-source success
- Free software examples
- Licensing, free software Vs. proprietary software
- Public domain software
- History of free software
- Software licensing model
- Uses of open-source software

Introduction to Open Source

- Initially proprietary software providers like IBM, Oracle, Microsoft and other dominated technology
- **Open source**
  - Has transformed how software is built and implemented
  - Open Source is almost synonymous to Software Development
  - Most collaborative method of software development and increasingly popular over the last 2 decades
- In 2019, Open Source services exceed $17B and expected to reach $33B by 2022 (CB Insights Marketing tool)

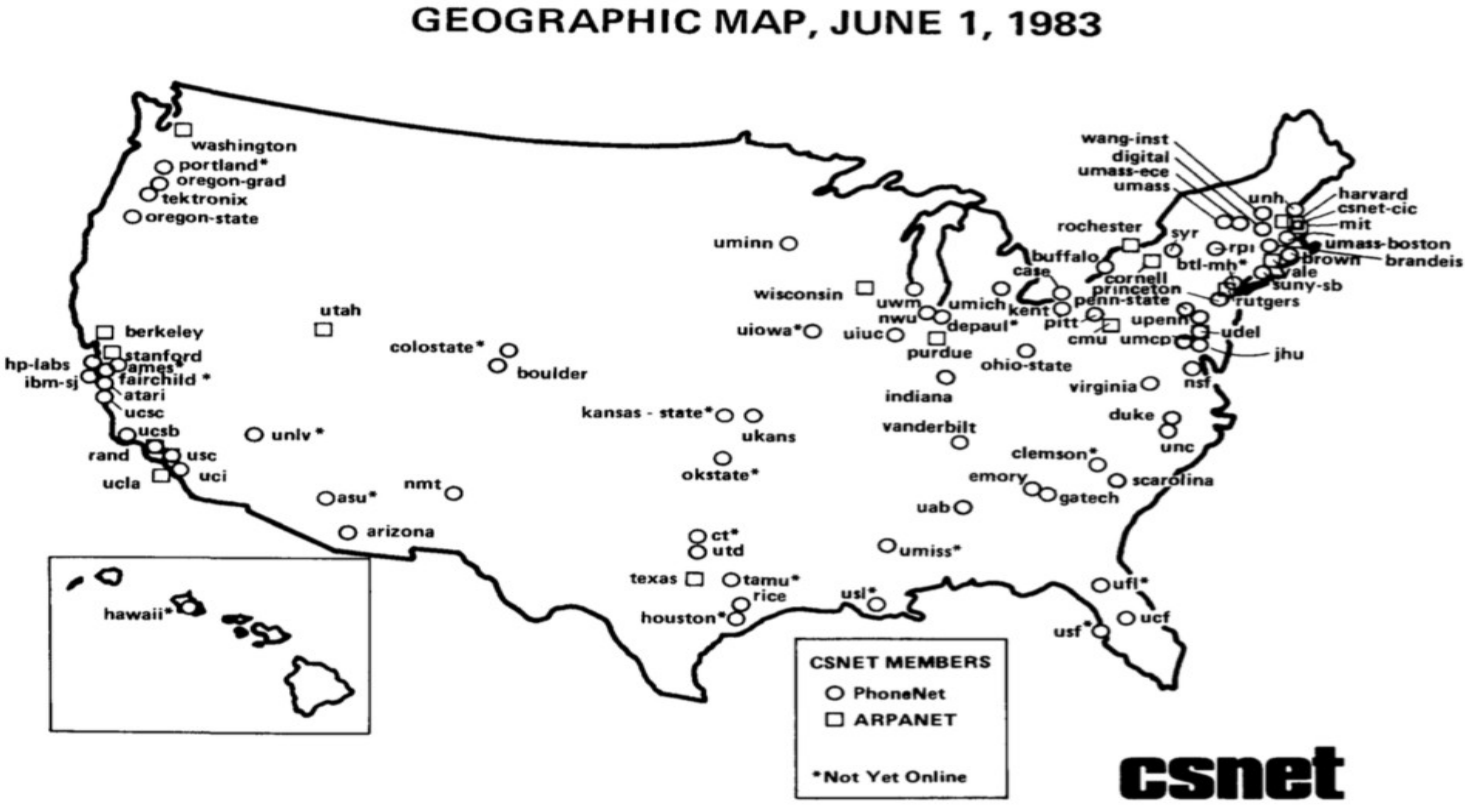
Introduction to Open Source(cont.)

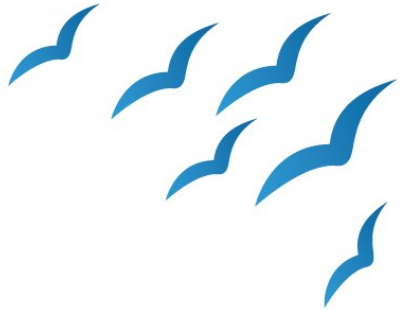
- Characteristics of Open-source Software
  - Publicly accessible, allows developers to exchange code & ideas in a transparent and collaborative manner, flexible to solve wide range of business problems
- Today, over **20 million** developers contribute to community-based platforms like **GitHub**.
- Market is expected hundreds of millions based on the recent acquisitions
  - Red Hat (acquired by IBM for $34B)
  - Git Hub (acquired by Microsoft for $7.4B), MongoDB($7.9B) and Elastic($7.3B)

What is Open-source Software?

- Open-source is software development model, allows startups by facilitating collaborative development
- Allows users to copy, inspect, modify, enhance and redistribute source code
- **History**
  - 1950s to early 1970s, computer users have software freedoms associated with the free software
  - Software was available in public domain, commonly shared by the users and hardware manufacturers
  - Used like '**SHARE**' to exchange software, written by BASIC, distributed and shared as printed source code- magazines, books, electronic communication etc (Creative Computing, Softside, Compute !, Byte etc.)

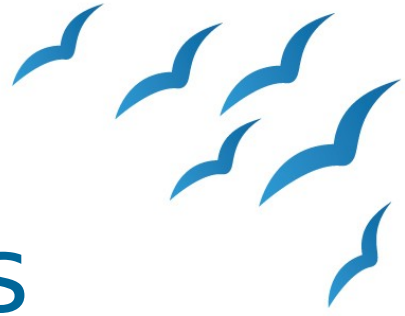
- Early adopters of computer SW&HW, Universities began to share bug fixes, even software enhancements, with other Universities
- Later Universities shared their own , free-to-use software with other public. Communities, Books
- ARPANET and CSNET used to share electronic form





History of Free Software

- Late 1970s and early 1980s, software companies (Microsoft, IBM, Oracle, SAP, SAS(Statistical Analysis System), Xerox etc. started producing software for sale, distributing in binary, preventing the users to study or adapt the software
- In 1980, copyright law extended to computer programs. NDA, restricted access
- 1983, **RMS**, author of Emacs editor, member of MIT AI hacker community announced GNU project -**GNU is Not Unix**
- 1985, FSF, developed definition for copyleft



Free Software and freedoms

1. Freedom to run the program as you wish, for any purpose (**freedom-0**)
2. Freedom to study how the program works, and change it so it does your computing as you wish (**freedom 1**). Access to the source code is a precondition
3. Freedom to redistribute copies so you can help your neighbor (**freedom 2**).
4. Freedom to distribute copies of your modified versions to others (**freedom 3**). Whole community a chance to benefit from your changes. Access to the source code is a precondition

Some of the notable Open-source Projects

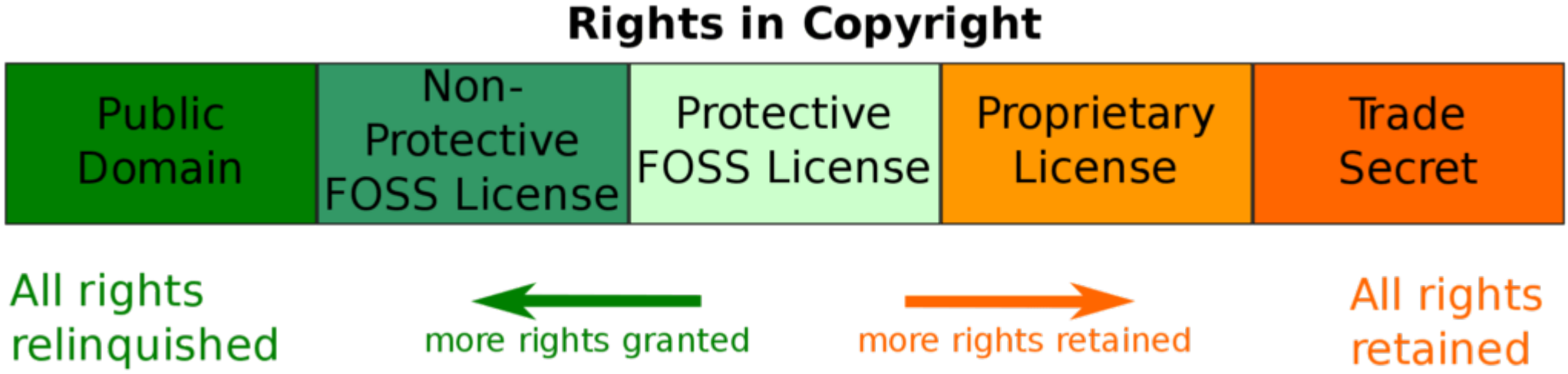
- **Mozilla Firefox** (open-source internet browser at Netscape) in 2002
- **Git** (a source code version-control system created by Linus Torvalds) in 2005,
- Recent open-source database like **Radis**
- Infrastructure as service- **OpenStack**
- Machine earning Library - **TensorFlow**

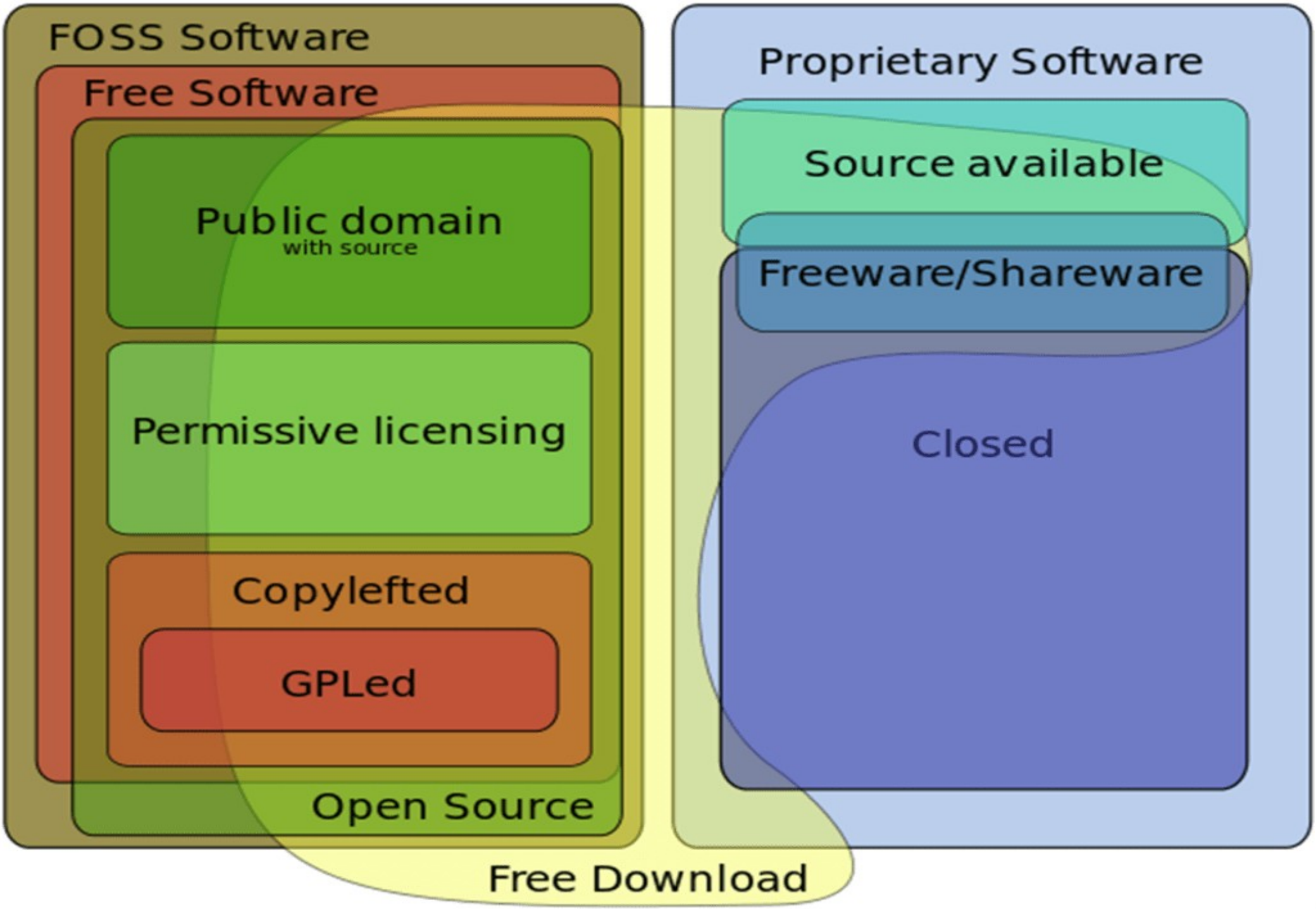
Open-source software Definition

- Open-source software (OSS) is any type of program or application while developers can inspect, copy, modify and redistribute
- Also referred to as free, open-source software (FOSS)
- Requirements to consider it as open-source,
  - The source code of the program – the underling code that makes up the design, functionality,and defining the attributes of an application must be publicly accessible
- With the access, study it, “fork”(copy) it, change it, and share the modified version with others

Software Licensing

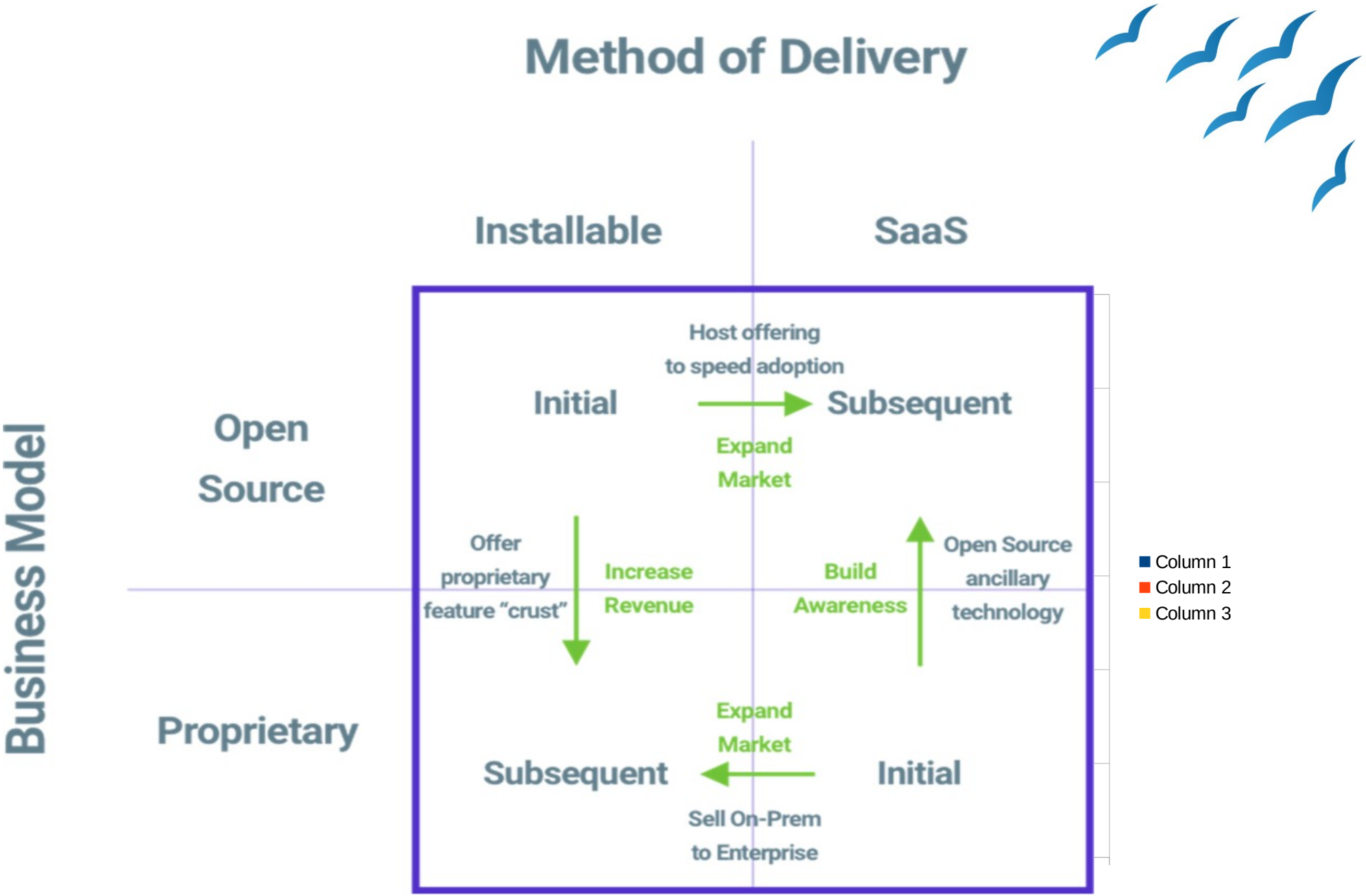
- A software license is a document that provides legally binding guidelines for the use and distribution of software.
- Software licenses typically provide end users with the right to one or more copies of the software without violating copyrights.



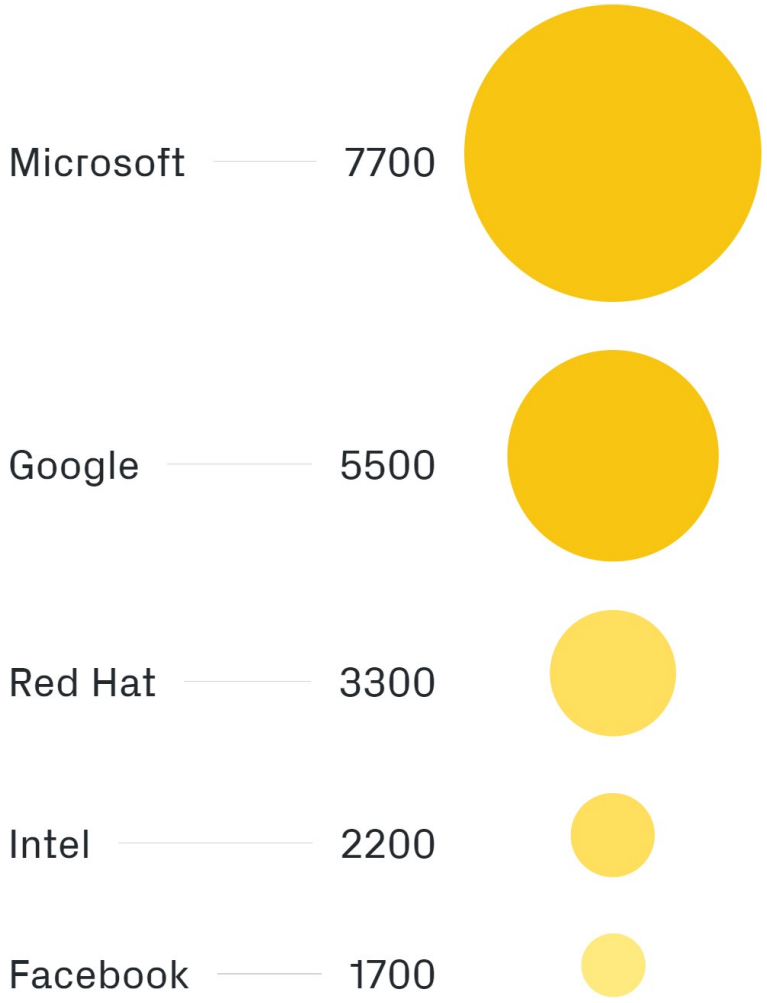

FOSS Software
Free Software
Public domain
with source
Permissive licensing
Copylefted
GPLed
Open Source
Proprietary Software
Source available
Freeware/Shareware
Closed
Free Download

How to make money with OSS

- Most OSS projects don’t start with the goal of monetization, instead look to provide solutions to problems
- Some times those problems are small and experienced by few developers, other times much larger
- When a project provides solution to a big problem, demand for that project grows, as scales, revenues grow.
- Revenue generation business started by the founders of the project, to provide enterprise support to large organizations
- Enterprise wants assurance-security flaws to be fixed, dedicated assistance, software longivity, companies provide
- Commercial support – established method of monetization
- Monitization strategies varies from company to company and change as an organization matures


Method of Delivery
Installable
SaaS
Business Model
Open Source
Proprietary
Host offering to speed adoption
Initial
Subsequent
Expand Market
Offer proprietary feature "crust"
Increase Revenue
Build Awareness
Open Source ancillary technology
Expand Market
Subsequent
Initial
Sell On-Prem to Enterprise
Column 1
Column 2
Column 3

Who are the developers of Open-source Software



- Many of the open-source projects are built by individual developers from the ground up, many of the newest and most successful projects are by the world’s largest tech companies
- Microsoft, Google, Intel, and Facebook(all are proprietary) contributing various projects on GitHub
- Most of the Microsoft developers are GiHub profile

Google’s own open-source projects

Google offers open-source, multi-cloud tools

July 2015

Kubernetes v1.0

Kubernetes manages containerized applications across multiple hosts, providing basic mechanisms for deployment, maintenance, and scaling of applications.

May 2017

Istio v0.1

Istio is an open-source framework for connecting, managing, and securing microservices across heterogeneous clouds platforms. It is a service mesh similar to that offered by HashiCorp and Buoyant.

July 2018

Knative v0.1

Knative (pronounced kay–native) is an open-source, Kubernetes-based platform that helps developers build, deploy, and manage modern serverless workloads.

OSS Projects and contributions

- Recent Google’s open-source project is Machine Learning Library- **TensorFlow**
- Its widespread use has created a large, engaged community, resulting in contributions from many independent developers
- Microsoft Visual Studio Project has over 19,000 contributors, while MS-7700 a minor
- 8 out of 10 most popular GitHub projects of big tech companies Microsoft, Facebook, Google and IBM(Ansible) – only a fraction of project contributions come from their respective employees
- At the time of Microsoft’s acquisition, GitHub counted 28M users and 85M repositories.
- And at the end of 2018, GitHub had 31M+ users and 96M+ repositories.

Top open source projects

VS Code, React, and Tensorflow once again top our list of open source projects by contributor count. New to the list are projects that manage containerized applications, share Azure documentation, and consolidate TypeScript type definitions: Kubernetes, Azure Docs, and DefinitelyTyped.

		Contributors
1	Microsoft/vscode	19K
2	facebook/react-native	10K
3	tensorflow/tensorflow	9.3K
4	angular/angular-cli	8.8K
5	MicrosoftDocs/azure-docs	7.8K
6	angular/angular	7.6K
7	ansible/ansible	7.5K
8	kubernetes/kubernetes	6.5K
9	npm/npm	6.1K
10	DefinitelyTyped/DefinitelyTyped	6.0K

Reasons to use Open-source Strategy

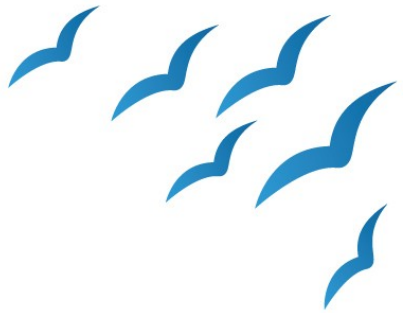
- Reduce dependency on closed source vendors
- Annual budget will not increase with software maintenance costs (18-25%)
- More access to tools.
  - variety of development and testing tools, project and portfolio management tools, network monitoring, security, content management, etc
- Try before use
- 24x7 online support- huge community
- Access to source code and ability to customize
- More features as open source is driven by community
- Great negotiating power to deal with closed source vendors
- More secured than the closed source vendors
- Faster bug fixes

Principles of Open-source Software

- Licenses are free to
  1. Use software for any purpose
  2. Make copies of OSS and distribute them without payment of royalties to the licensor
  3. Create derivative works of OSS and distribute them without payment of royalties
  4. Access source code and use
  5. Combine OSS and other software
- Transparency, participation and collaboration

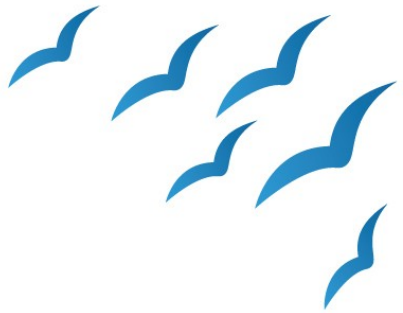
Open Source Standards Requirement

- **Requirement**: An “Open Standard” must not prohibit conforming implementation in Open Source Software
- Criteria to comply with Open Source Standard
  - **No intentional secrets**
    - Must not withhold any detail necessary for interoparable operation
  - **Availability**
    - Standard must be freely and publicly available
  - **Patents**
    - Should be royalty-free covered by non-assertion
  - **Agreements**
    - Must not be any license agreement
  - **No OSR-incomplete Dependencies**
    - Must not require any other technology



Open Source Success

- There are many open source projects
  - sourceforge.org, a leading site for open source development. Lists nearly **80,000**
  - Good number of open source projects are highly successful
  - Provide widely used alternative to traditional or “closed source”
  - Many more projects are used by communities of experts
- The most successful projects to be infrastructure software, not applications



Open Source Success(cont.)

- More on server than client side
  - More expert users of servers
  - Server operating systems
    - Linux
  - Certain server software
    - Web
    - Mail
    - DNS
  - So far less so with other server software
    - Database (MySQL rising, less easy to measure)
- Software for technical users

Related search

Open source software's

View 3+ more


OpenStack

GIMP

Linux kernel


Kubernetes

MySQL

Blender

Ubuntu

Related search

Open source organizations

View 3+ more

Free software movement

Open Source Initiative

Open Source Geospati...

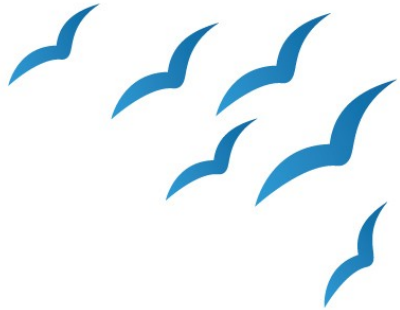
Linux Foundation

Red Hat Software

Open Source Ecology

Mozilla Foundation

Open Source Organizations



- Free Software Foundation (FSF)
- Open Source Initiative, GNOME Foundation
- Apache Foundation; Cloud Native computing Foundation (CNCF)
- Digital Freedom Foundation; Document Foundation
- Drupal Association; Eclipse Foundation
- KDE e.V.; Linux Foundation; Linux Professional Institute
- Mozilla Foundation; Open Bio-informatics Foundation
- Open Educational Consortium (OEC); Open Source Automation Development Lab
- Open Source Geo-spatial Foundation; Open Source for America
- Open Source Matters
- Open-stack Foundation;
- Oregon State University (OSU) Open Source Lab
- Software Freedom Conservancy; Software in the Public Interest
- Swecha, FSMI

Free software Examples

- Operating Systems: Linux, BSD, Darwin and Open Solaris, Android
- Compilers: GCC Compiler, GDB debugger and C/C++ libraries
- Server: BIND name server, Send mail transport, Apache web server and Samba file server
- Database systems: MySql, PostgreSQL, MariaDB, SQLite, MogoDB, CouchDB, Apache- Hadoop, Cassandra, Spark, Storm etc.
- Programming Languages: Java, Perl, PHP, Pythons, Lula, Ruby, Tcl/TK, Rust, Kotlin etc.
- GUI related: X Windows system, GNOME, KDE, Xface
- GRASS: Open source Geography Information System (GIS)
- OpenOffice.org suite, LibreOffice, Mozilla and Firefox web browser and GIMP graphics editor
- Typesetting: Free Software multimedia file type that can functions like mp3 or mp4,

Free Software Licenses

- GNU General Public License **GPL** (GPLv.1, GPL v.2, GPL v.3)
  - Most common – Linux kernel, GCC
- GNU Less General Public License (**LGPL**)
  - Comprises between the strict GPL and the more permissive licenses such as BSD and the MIT license; It does not carry the “inheritance” element of GPLSD licenses
- **BSD License** – more permissive, BSD kernel Unix like OS
- **Mozilla** licenses
  - Mozilla Foundation, weak copy-left, Firefox, Thunderbird etc.
- **MIT** licenses
  - Originated from Massachusetts Institute of Tech. Is a public license. Most permissive type of license. X-Windows
- **Apache** license
  - Incompatible with the GPL., allows modification to remain secret and said modifications can be sold. Apache Web server
- “**as-is**” release model
  - Released code is simply placed in public domain with no restrictions – dedicated libraries


CDDL
copyleft
BSD
GPL
Permissive
BUILT ON
eclipse
zlib
MIT

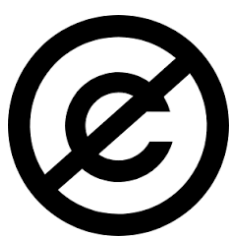
Proprietary Vs. Open Source Licensing Model

Proprietary / Closed source Software	Open Source Software
Licensor distributes object code only; source code is kept a trade secret	Licensor distributed source code
Modifications are prohibited	Modifications are permitted
All upgrades, support and development are done by licensor	Licensee may do its own development and support or hire any third party to do it
Fees are for the software license, Maintenance, and upgrades	Fee, if any, are for integration, packing, support and consulting
Sub-licensing is prohibited, or is a very limited right	Sub-licensing is permitted; licensee may have to distribute the source code to program and modification
Ex: Microsoft Windows, Acrobat etc.	Ex: GCC, Wikipedia, Firefox etc.

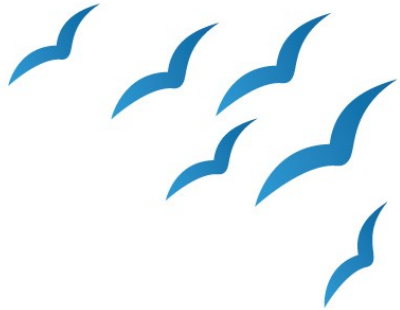
Open-source Vs Proprietary SW

Open Source	Proprietary Software
Developed & tested through open collaboration	Solely developed by the individual or the organiztion that developed it
Anyone with academic knowledge can access, inspect, modify and redistribute the source code	Only the owner or publisher who holds the legal property rights of the source can access it
Projects are maintained by an open source community of developers and programmers	Projects are managed by a closed group of individuals or team that developed
Not aimed at unskilled users outside of the programming community	Focused on a limited market of both skilled and unskilled end users
Provides better flexibility which means more freedom, encouragement to innovation	Limited scope of innovation with the restrictions and all
Ex: Android, Firefox, LibreOffice, Ubuntu, Drupal, GNOME etc.	Ex: Windows, MacOS, iTunes, Google Earth, Adobe Flash Player etc.

Pubic domain Software

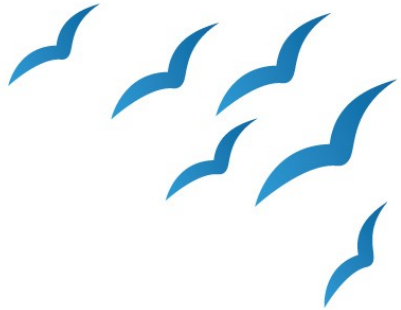


- 1950s to 80s software culture, was academic phenomena, “**Public domain**” software was popular
- Freeware, shareware, FOSS was created in academia and by hobbyists and hackers
- Software was written in an interpreted languages like BASIC, source was distributed as printed code through magazines(like Creative Computing, Softside, Compute!, Byte etc.) and books like best sellers BASIC computer Games.
- Early on, closed -source was uncommon until mid 1970s and 80s.
- 1974, US Commission on New Technological Uses (CONTU) decided that “Computer Programs, to the extent that they embody an author's original creation, but, software was not copyright-able, and there fore always in public domain. (Apple v. Franklin in 1983 for object code)
- Common way to share public-domain software was through user groups, company like PC-SIG of Sunnyvale, California, BBS networks.



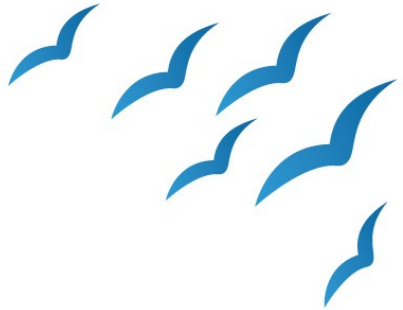
Public domain Software

- Public domain software was commercialized sometimes by a donation-ware model, asking the users for a money donation to be sent by mail. Later shareware, freeware models came
- Berne convention 1988, Act: before this act, released without a copyright notice, after the act, software was by default copyright-protected, and need explicit waiver statement or license of the author
- Cryptograhic algorithms are under public domain- CERThttpd 1993, serpent cppher 1999,Openwall project
- Examples include LINUX, Apache, Firefox, Thunderbird, OpenOffice, KOffice, and SquirrelMail.



Advantages of OSS

- Lesser hardware costs
  - Easily portable and compressed
- High-quality software
  - Well designed and efficiently used in coding
- No vendor lock-in
  - Freedom
- Integrated management
  - CIM and WBEM; servers, applications integrated
- Simple license management
  - GPL; installed several times
- Lower software costs
- Abundant support
  - Larger communities, support
- Scaling and consolidating
  - Easily scaled, options varies for clustering, load balancing and open source applications



Uses of Open-source

- Software Freedoms
- Monetary benefits
- Speed development
- Community support
- Collaboration
- Contribution
- Business opportunities
- Internationalization and Multi-lingual support

References

- https://en.wikipedia.org/wiki/Free_software#1980s:_Foundation_of_the_GNU_project
- https://www.cbinsights.com/research/report/future-open-source/#what
- https://en.wikipedia.org/wiki/Public-domain_software