

Economical and legal basics of Open Source software

Legal Heidelberg OSADL Talks, September 29, 2020, Online Session 1

Open Source in industry:

What is Open Innovation, what is Open Source?

Basics of international copyright law

Principle of a software license

What is copyleft, what is a derivative work?

What is “Open Innovation”?

Joint Research and Development

- Goal of participating companies and institutions:
Joint added value
- Underlying concept:
Open knowledge economics
- Type of organization:
Collaborative environments

What are the possible advantages of Open Innovation?

User and provider:

- Standardization of hardware and software interfaces
- Larger base of knowledge, expertise and experience
- Avoiding unnecessary parallel development
- Reduction of development cycles

User:

- Participation at the innovation process

Provider:

- Satisfied users

What are the possible advantages of Open Innovation?

User and provider:

- Standardization of hardware and software interfaces
- Larger base of knowledge, expertise and experience
- Avoiding unnecessary parallel development
- Reduction of development cycles

User:

- Participation at the innovation process

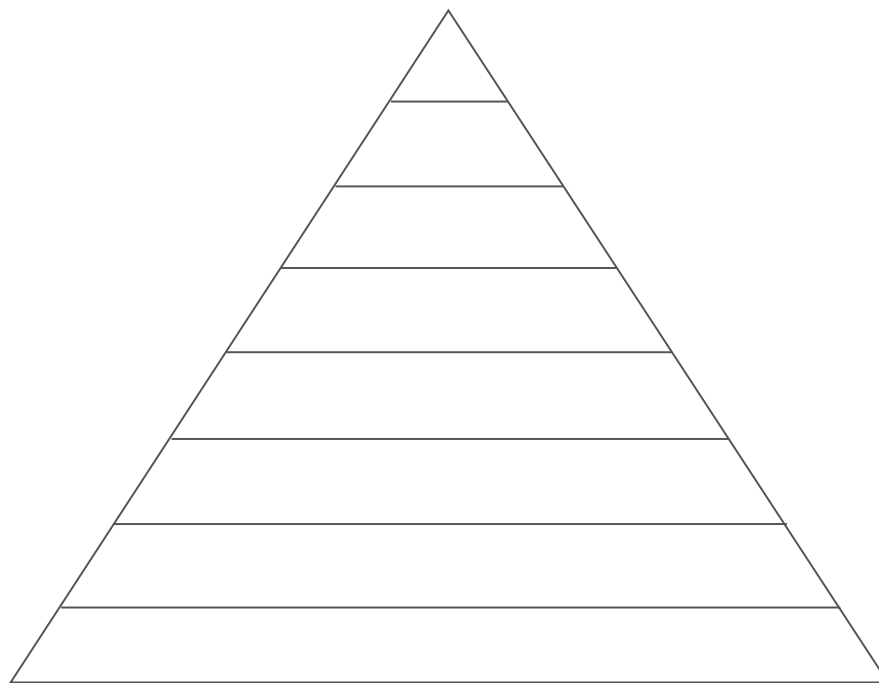
Provider:

- Satisfied users

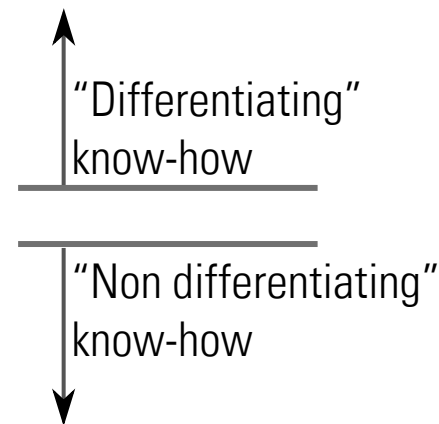
Cost reduction



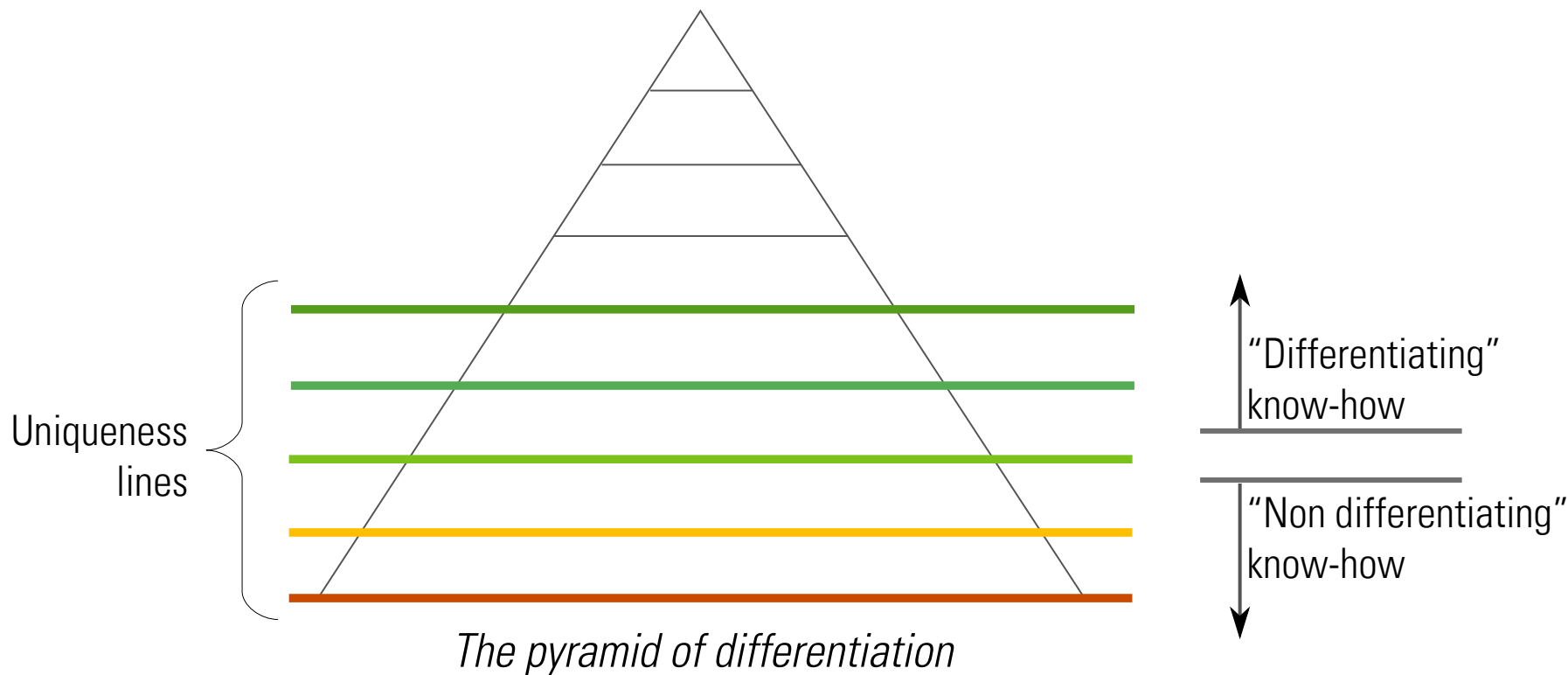
Is Open Innovation applicable to a given project?



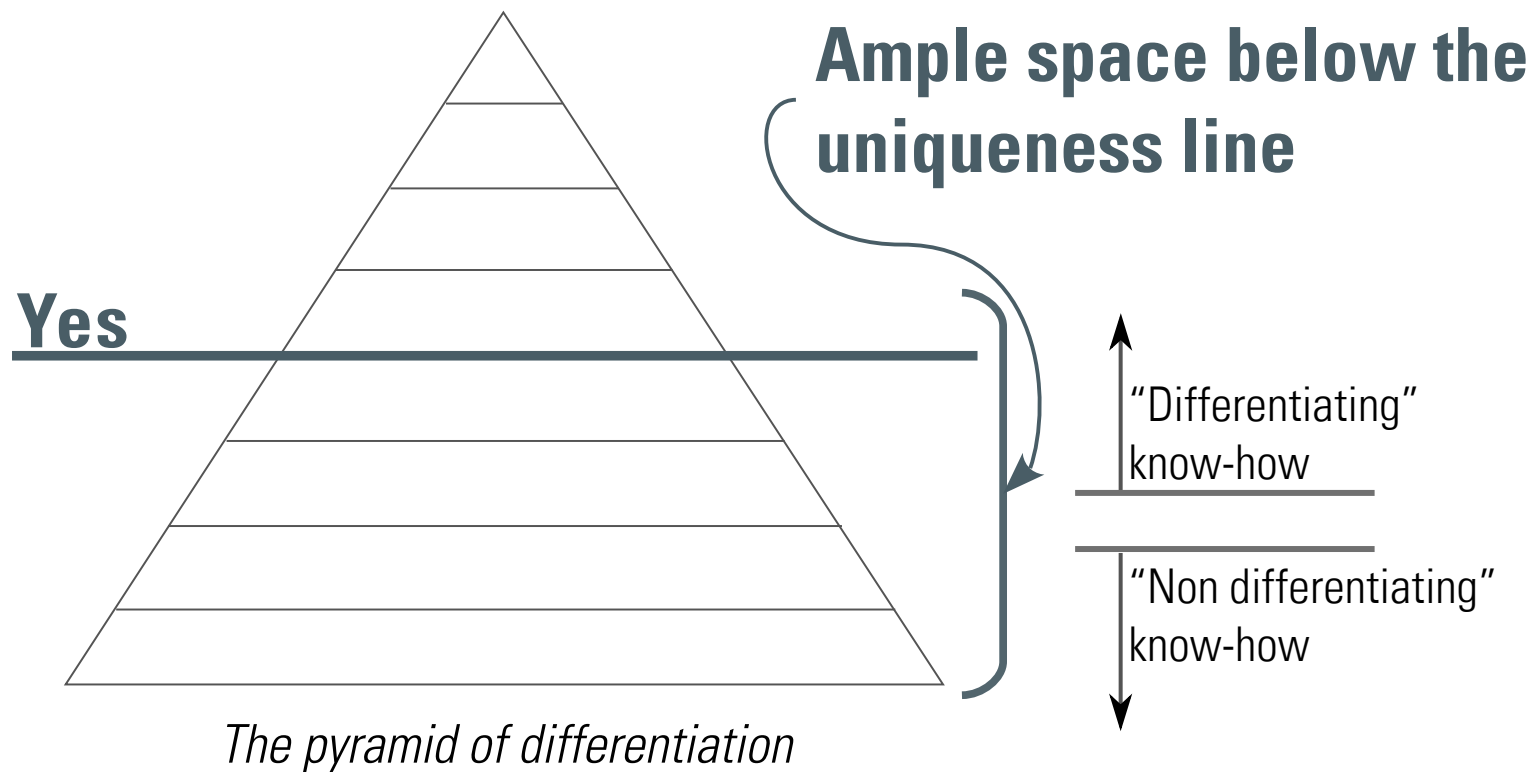
The pyramid of differentiation



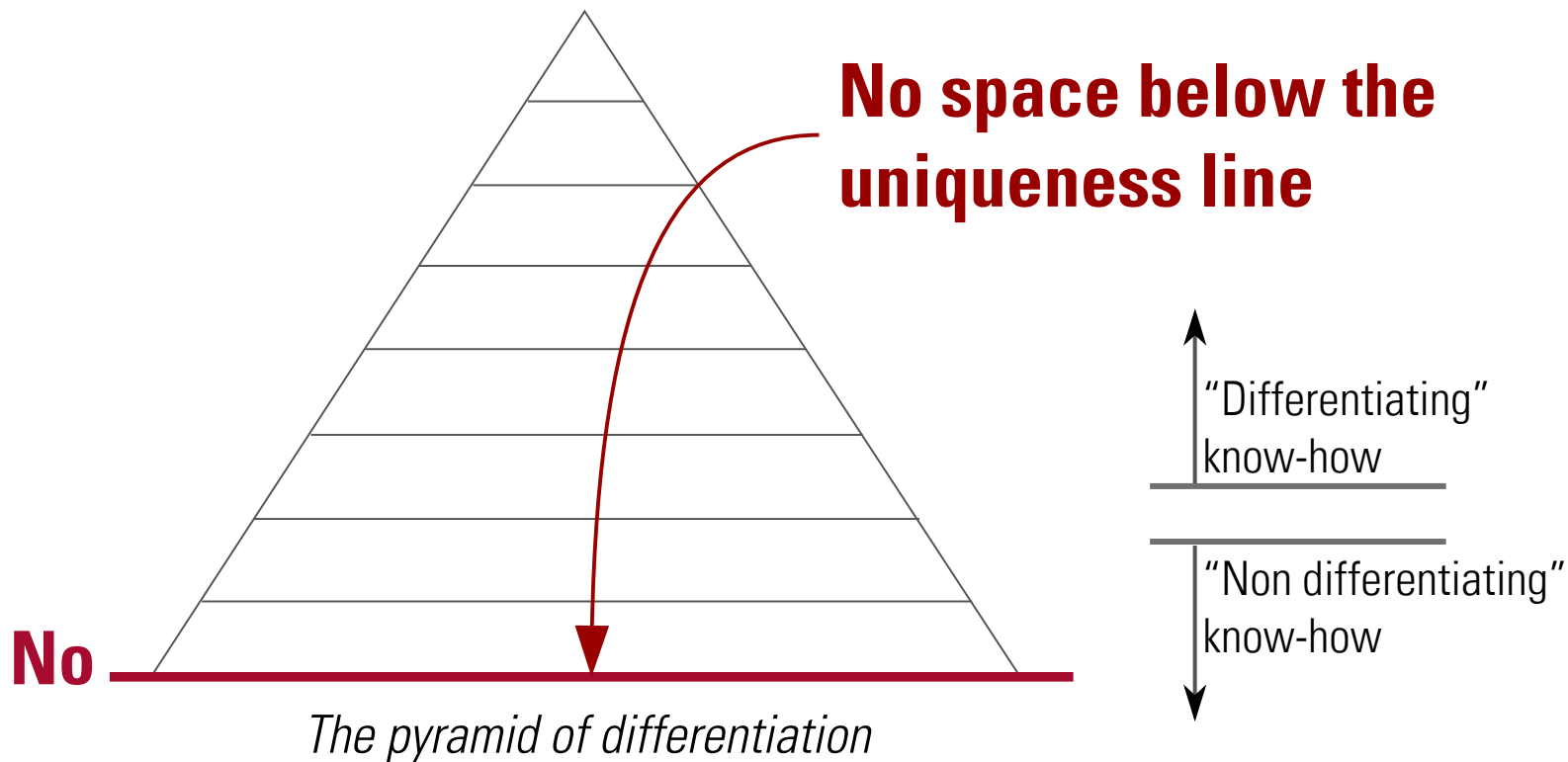
Is Open Innovation applicable to a given project?



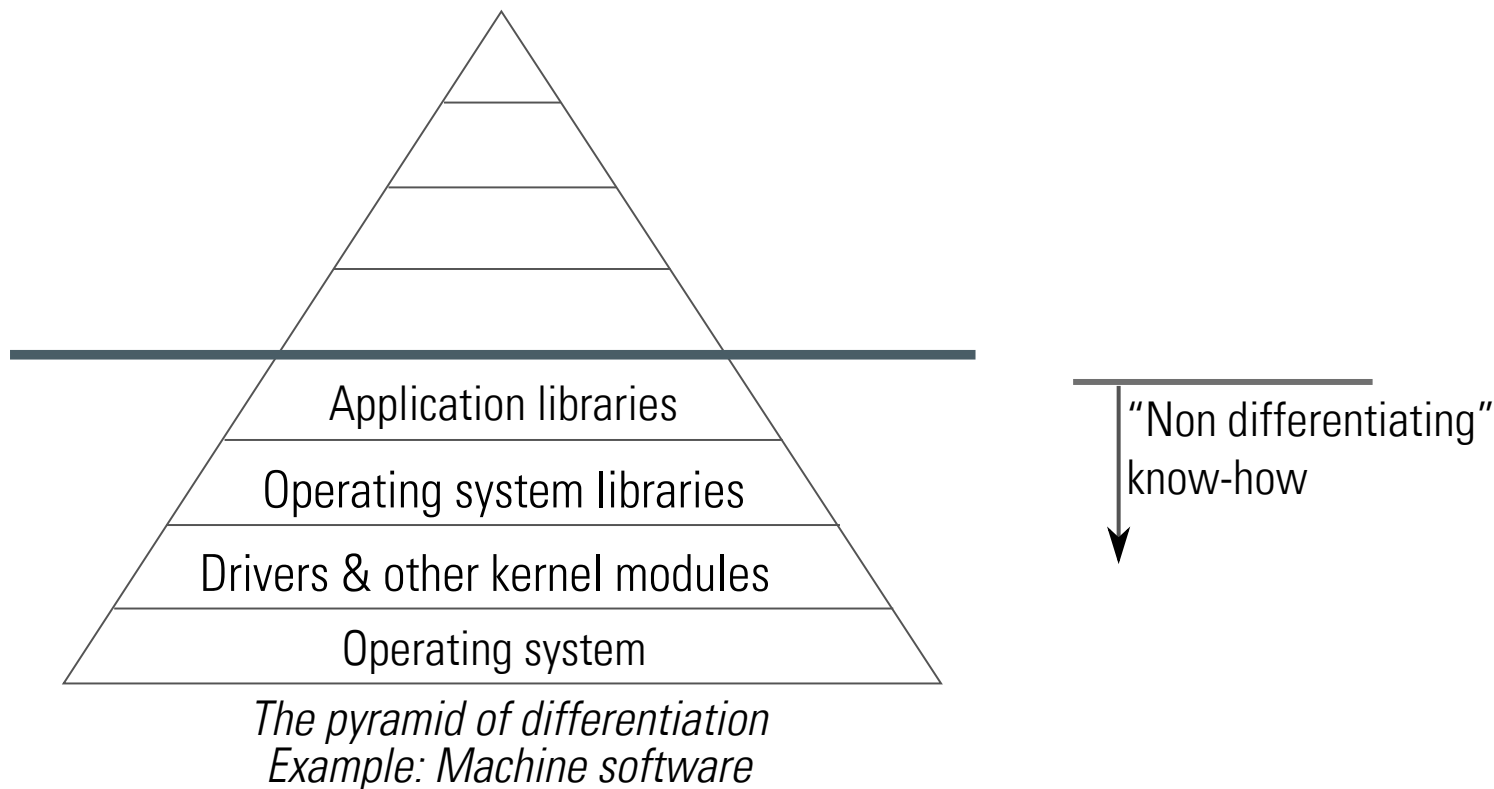
Is Open Innovation applicable to a given project?



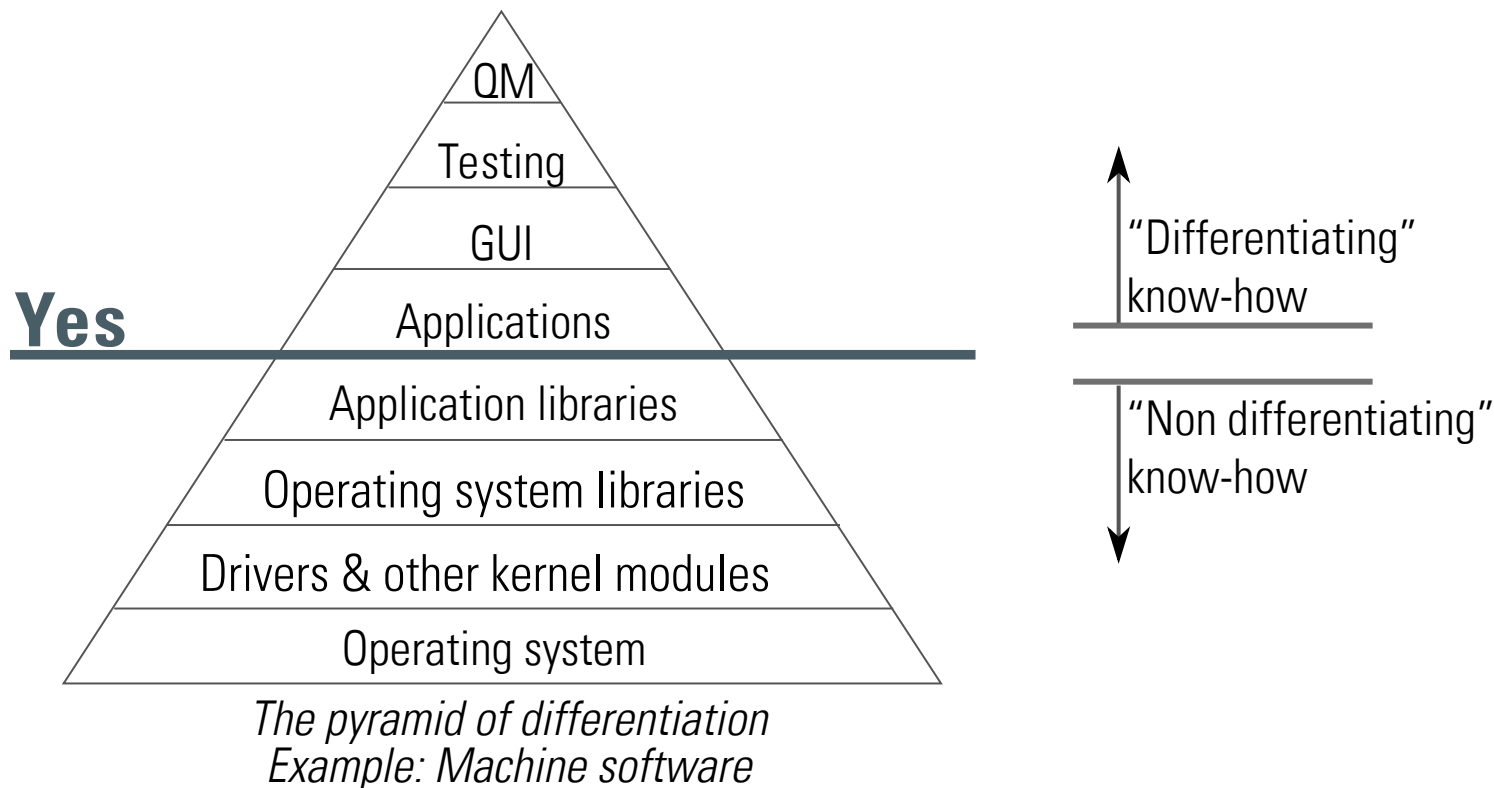
Is Open Innovation applicable to a given project?



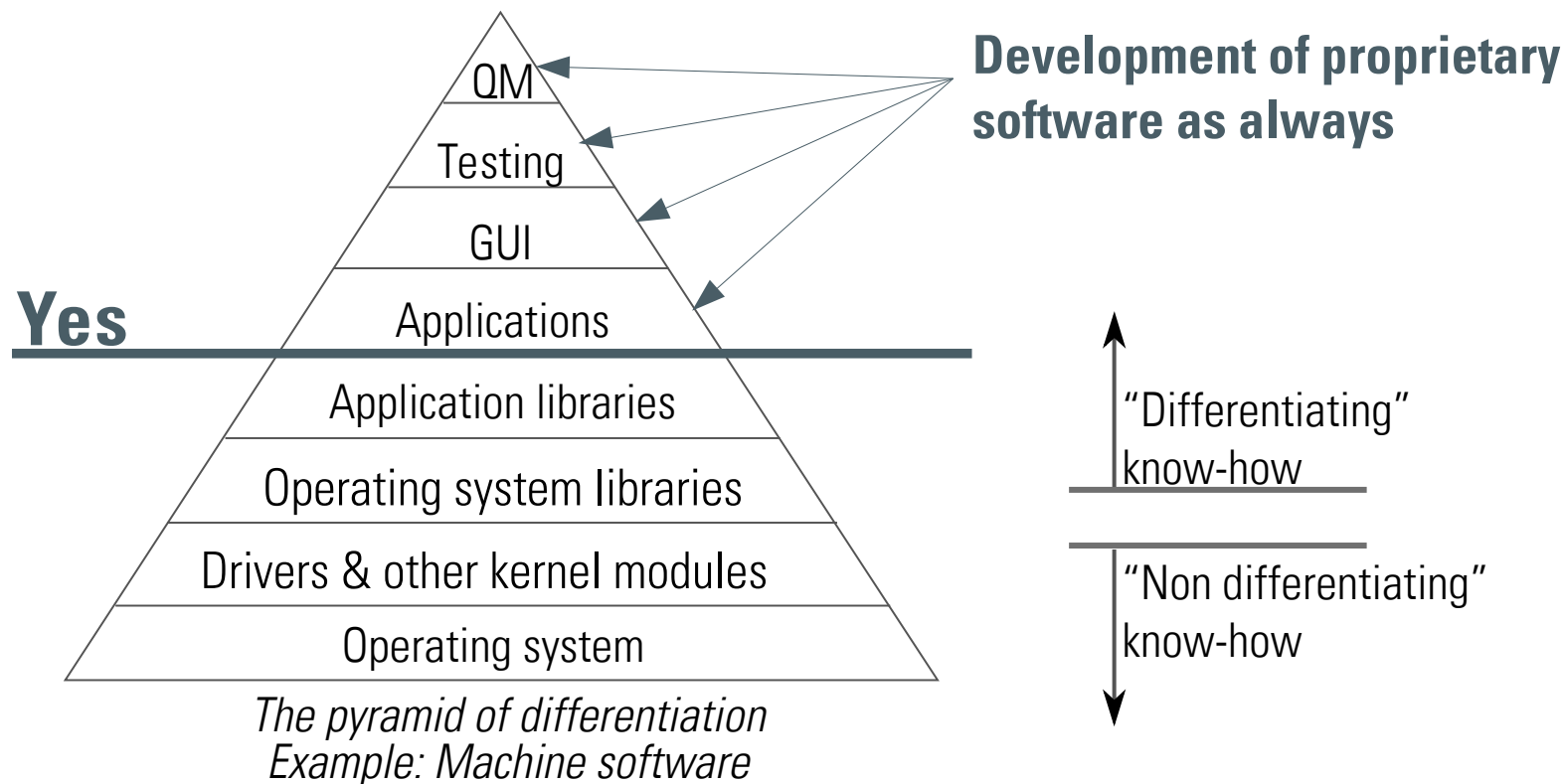
Is Open Innovation applicable to a given project?



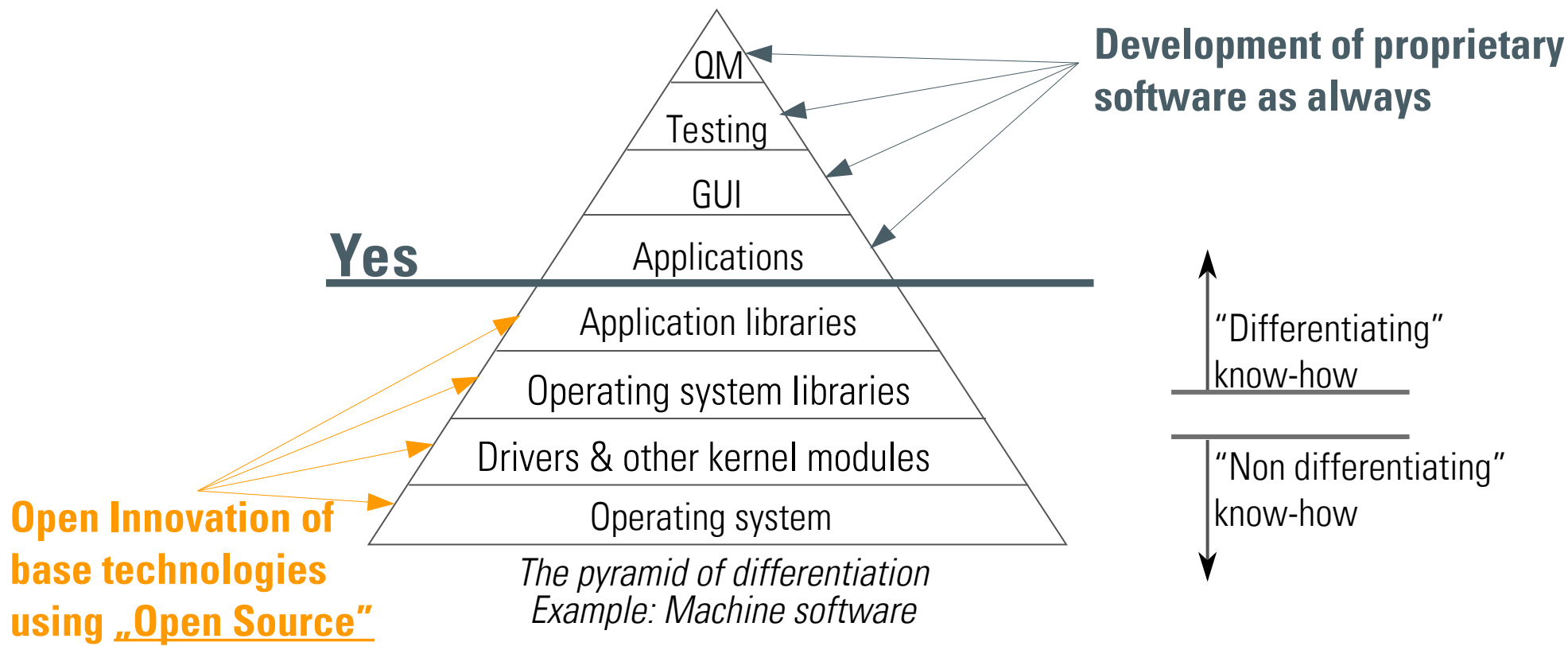
Is Open Innovation applicable to a given project?



Is Open Innovation applicable to a given project?



Is Open Innovation applicable to a given project?



What is Open Source software?

- **“Open Source”** is not an English phrase, but a **technical term**.
 - Cannot be translated
 - The original form “Open Source” must always be used – irrespective in which context, country or language.
- Open Source software must fulfill certain **requirements**, for example given in the Open Source Definition (OSD) released by the Open Source Initiative (OSI, <https://opensource.org/OSD>).

What is Open Source software? (2)

Requirements for Open Source software

- A software is Open Source if the holders of rights allow everyone
 - ✓ to **unrestrictedly and unconditionally use, analyze** and **modify** the software
 - ✓ to **copy and distribute** the software under very liberal **conditions**.
- Otherwise it may not be called “Open Source”.
- In particular: Being available as **source code** does **not** automatically make a software **“Open Source”**.

Proprietary vs. Open Source software

Action	Proprietary software		Open Source software	
	What is needed?	By whom/why?	What is needed?	By whom/why?
Run the software		Prohibited by the authors		Permitted by unilateral declaration of will of the authors
Analyze the software				
Modify the software				

Proprietary vs. Open Source software

Action	Proprietary software		Open Source software	
	What is needed?	By whom/why?	What is needed?	By whom/why?
Run the software	End User License Agreement (EULA)	Permitted by EULA	Access to the software	Permitted by unilateral declaration of will of the authors
Analyze the software		Prohibited by the authors	Access to the software	
Modify the software			Access to the software	

Proprietary vs. Open Source software

Action	Proprietary software		Open Source software	
	What is needed?	By whom/why?	What is needed?	By whom/why?
Run the software	End User License Agreement (EULA)	Permitted by EULA	<i>Access to the software</i>	Permitted by unilateral declaration of will of the authors
Analyze the software		Prohibited by the authors	<i>Access to the software</i>	
Modify the software			<i>Access to the software</i>	
Copy and distribute unmodified software		Prohibited by copyright law		Prohibited by copyright law
Copy and distribute modified software				

Proprietary vs. Open Source software

Action	Proprietary software		Open Source software	
	What is needed?	By whom/why?	What is needed?	By whom/why?
Run the software	End User License Agreement (EULA)	Permitted by EULA	<i>Access to the software</i>	Permitted by unilateral declaration of will of the authors
Analyze the software		Prohibited by the authors	<i>Access to the software</i>	
Modify the software			<i>Access to the software</i>	
Copy and distribute unmodified software	License	Permitted by license	License	Permitted by license
Copy and distribute modified software		Prohibited by copyright law	License	

Proprietary vs. Open Source software

Action	Proprietary software		Open Source software	
	What is needed?	By whom/why?	What is needed?	By whom/why?
Run the software	End User License Agreement (EULA)	Permitted by EULA	Access to the software	Permitted by unilateral declaration of will of the authors
Analyze the software	Tough control rules must be accepted	Prohibited by the authors	Liberal license obligations must be fulfilled	
Modify the software			Access to the software	
Copy and distribute unmodified software	License	Permitted by license	License	Permitted by license
Copy and distribute modified software		Prohibited by copyright law	License	Permitted by license

What is protected by copyright law?

Copyright law protects “works”.

What is a work?

What is a work?

Question #1: Who is the **creator** of the work?

- A work protected by copyright must be created by a human being, that is, by a **natural person**.
- Things created by animals – including primates – are not considered works in terms of copyright. Same principle applies to things created by a computer by itself.
- Plants, trees or even animal products, e.g. skins or mussels with exceptional designs are not protected by copyright.
- Natural objects, like stones, for instance, are not protected by copyright.

What is a work? (2)

Question #2: Is the work **perceptible** by human beings?

- The work must be able to be perceived by the senses of a human being.
- The work does not have to be embodied. It can also be a sequence of tones (music) or movements (theater, dance).

What is a work? (3)

Question #3: Is the work a result of **individual creativity**?

- The work created by an author has to be new or different in order to be protected by copyright.
- The work must reflect a recognizable individuality of the creator.
- In case two very similar or even practically identical works are created accidentally and independently by two different people, both works are protected by copyright – even if in an individual case it might be difficult to correctly determine who is the author of which work.

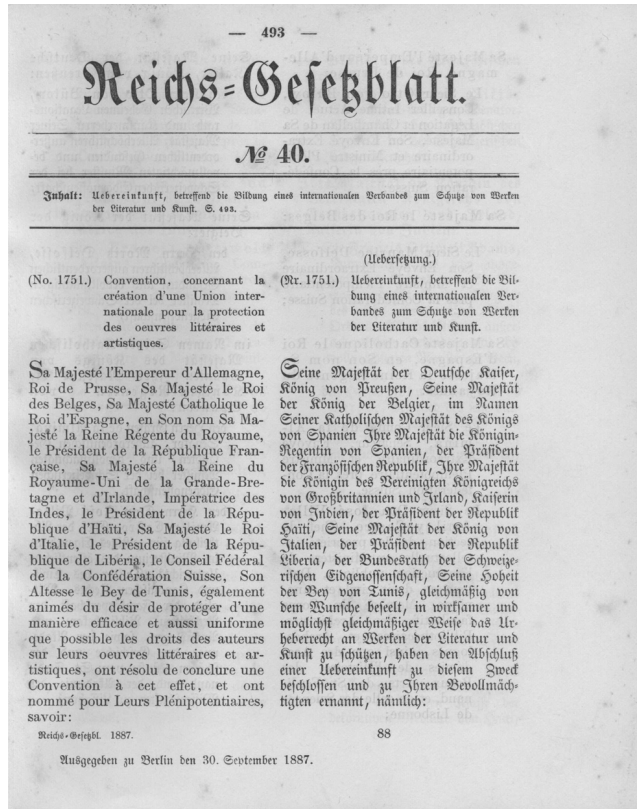
What has all of this to do with software?

Copyright law protects **works of art and literature**

- Somebody who writes something is a writer.
- The work of a writer is called literature.
- A person who is programming, writes (software). As a result, **software** belongs to literature and is **protected by copyright law**.

International copyright law

Berne Convention for the Protection of Literary and Artistic Works



Date	States
December 5, 1887	Belgium, Germany, France, Italy, Switzerland, Spain, Tunisia and United Kingdom
March 1, 1989	USA
As of February 2018	A total of 176 states

Article 5 of the Convention of Berne

- 1) **Authors shall enjoy**, in respect of works for which they are protected under this Convention, **in countries of the Union** other than the country of origin, **the rights which their respective laws do now or may hereafter grant to their nationals**, as well as the rights specially granted by this Convention.
- 2) **The enjoyment and the exercise of these rights shall not be subject to any formality**; such enjoyment and such exercise shall be independent of the existence of protection in the country of origin of the work.

The creator of a work has two different rights

Exclusive rights of use

- The author of a work has exclusive rights of use. The author may grant rights of use to third parties.

Moral rights of authorship

- Even after the transfer of the rights of use, the author has the right to claim authorship of the work and may object to any modification of the authorship information. In addition, the author may object to any modification of the work that would negatively affect the author's reputation. In Europe and in many other countries, the right of authorship cannot be sold or modified in any way.

Rights of use and right of authorship reflected in the copyright notice

Initially, an author has both the right of use and the right of authorship. Thus, the copyright notice looks like:

Copyright © 2019 John Doe

After the right of use is transferred, e.g. to the employer of John Doe (which happens automatically), the copyright notice may look like:

Copyright © 2019 Employer LLC, author John Doe

John Doe's employer may then sell the software, and the buyer may modify the copyright notice to:

Copyright © 2019 Buyer LLC, author John Doe

Terminology of the copyright notice

EN

Copyright © 2019 Employer LLC, author John Doe

Copyright notice

Author attribution

Copyright notice

DE

Copyright © 2019 Employer LLC, author John Doe

Rechteinhabervermerk

Urhebervermerk

Urhebervermerk

Which types of permissions exist?

Copyright law forbids copying and distributing of protected works.

If a (natural or legal) person wants to copy and distribute software, a permission is needed:

1. **Legal permissions**

- a) Backup copies
- b) Copies made during proper use

2. **Licenses** from holders of exclusive rights of use

- a) Proprietary run-time licenses
- b) Open Source licenses

License rules of thumb

- First of all, a license always is a "good" thing, because something is allowed that normally is not permitted.
- Even a license that only grants minimal rights of use is better than no license.
- Without license, the „default“ state occurs, this means copyright law applies and any copying and distributing is prohibited.

What does a license look like?

- The licensor grants **rights** to the licensee.
- The licensee is imposed **obligations** by the licensor.
- Non-fulfillment of obligations will be **sanctioned**; in that event and as a general rule, the rights granted will be withdrawn.

Example of a license with rights and obligations

GPLv2, sect. 1:

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

Example of a license with rights and obligations

GPLv2, sect. 1:

1. **You may** copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, **provided that** you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

The license usually becomes void, if license obligations are not fulfilled

GPLv2, sect. 4:

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License.

Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.

The license usually becomes void, if license obligations are not fulfilled

GPLv2, sect. 4:

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License.

Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and **will automatically terminate your rights under this License.**

What are the rights of the licensor in case of copyright violations?

- The copyright law regulates the **claims** a right holder may assert against a licensee in the event of non-fulfillment:
 - ♦ Injunctive relief, e.g. to no longer copy and distribute
 - ♦ Obligation to inform all customers to whom the unauthorized copies have been conveyed
 - ♦ Provision of complete customer lists
 - ♦ Physical destruction of unauthorized copies
- Penalties may be imposed for **infringement of copyright law**.

GPLv2, sect. 1 and 4 for programmers ...

```
/* Default by copyright law */
copyright = distributionright = 0;

if (add_copyright_notices_to_manual() &&
    add_disclaimer_of_warranty_to_manual() &&
    add_license_text_to_manual()) {
    copyright = distributionright = 1;
}

if ((!copyright || !distributionright) && copy() && distribute()) {
    bank_balance[holder_of_rights] += huge_amount;
    bank_balance[you] -= huge_amount;
    if (negligence >= gross && judge >= strong)
        goto jail;
}
```

Please do not forget:

- Software is protected by copyright law.
- Whoever wants to copy software needs a permission (license).
- Copying software without license violates copyright law and may entail serious consequences.

What is Free Software?

- „Free Software“ is – at least legally – identical to „Open Source“ software.
- Therefore, it is sometimes proposed to use the term “Free and Open Source Software” (FOSS).
- However, the two terms originate from two very different political, social and economical movements.

What is “Copyleft”?



Closed Source

Run-time license
No modifications



What is “Copyleft”?



Closed Source

Run-time license
No modifications

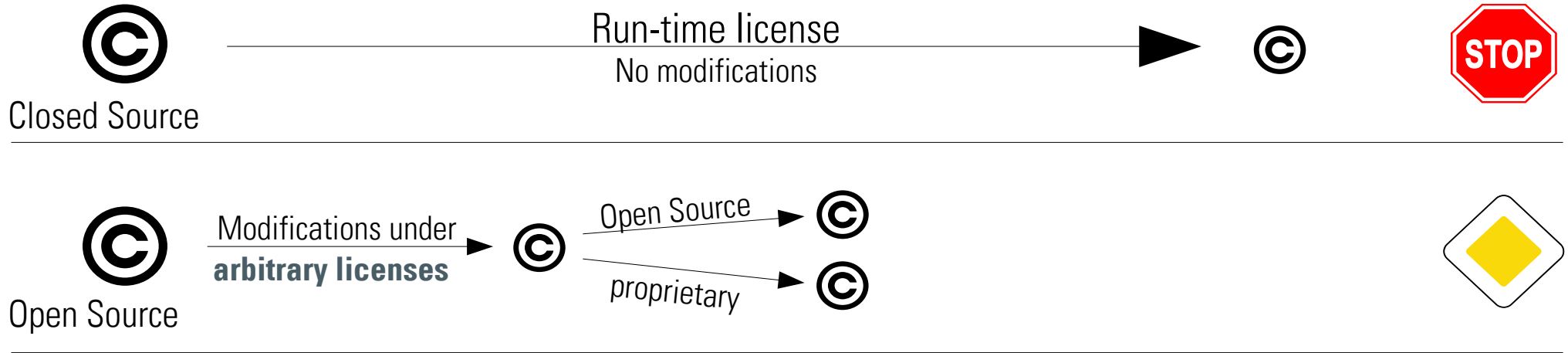


Open Source

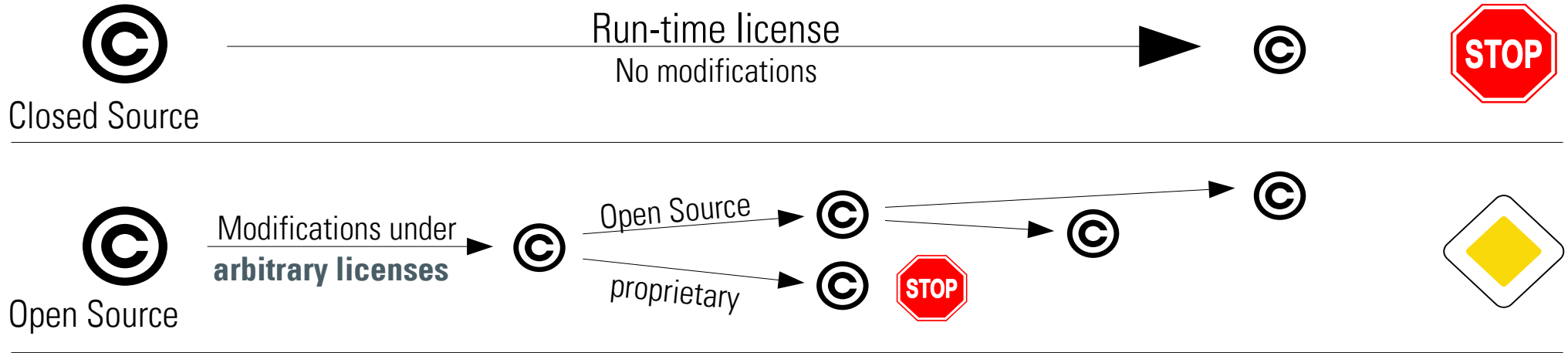
Modifications under
arbitrary licenses



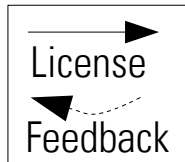
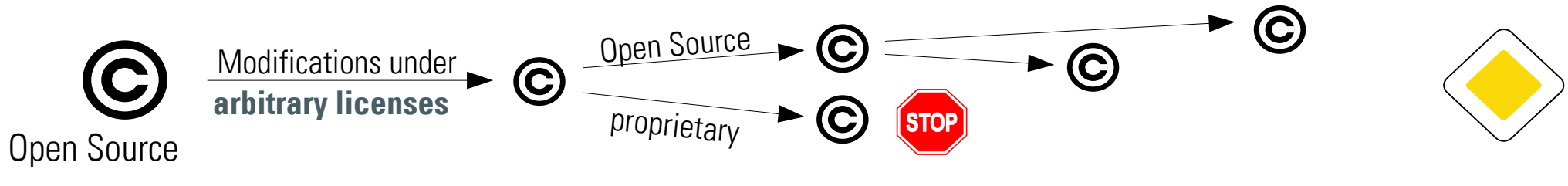
What is “Copyleft”?



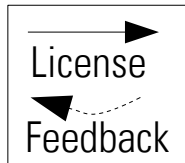
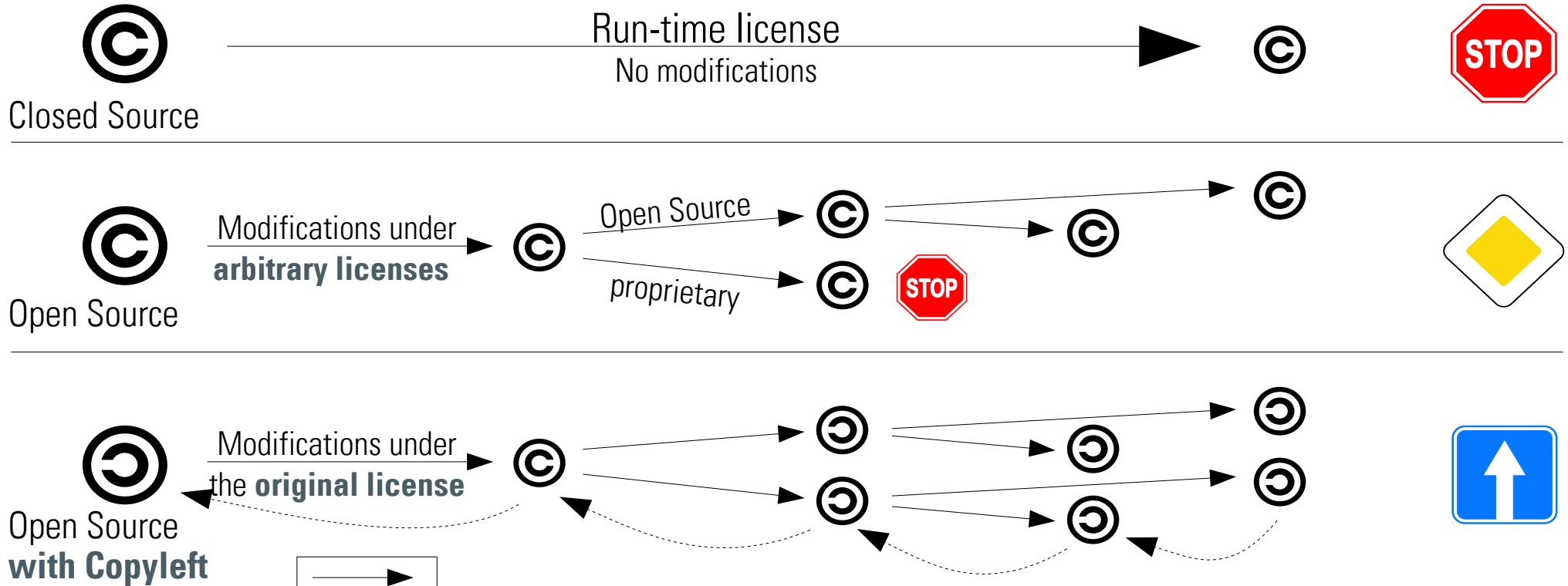
What is “Copyleft”?



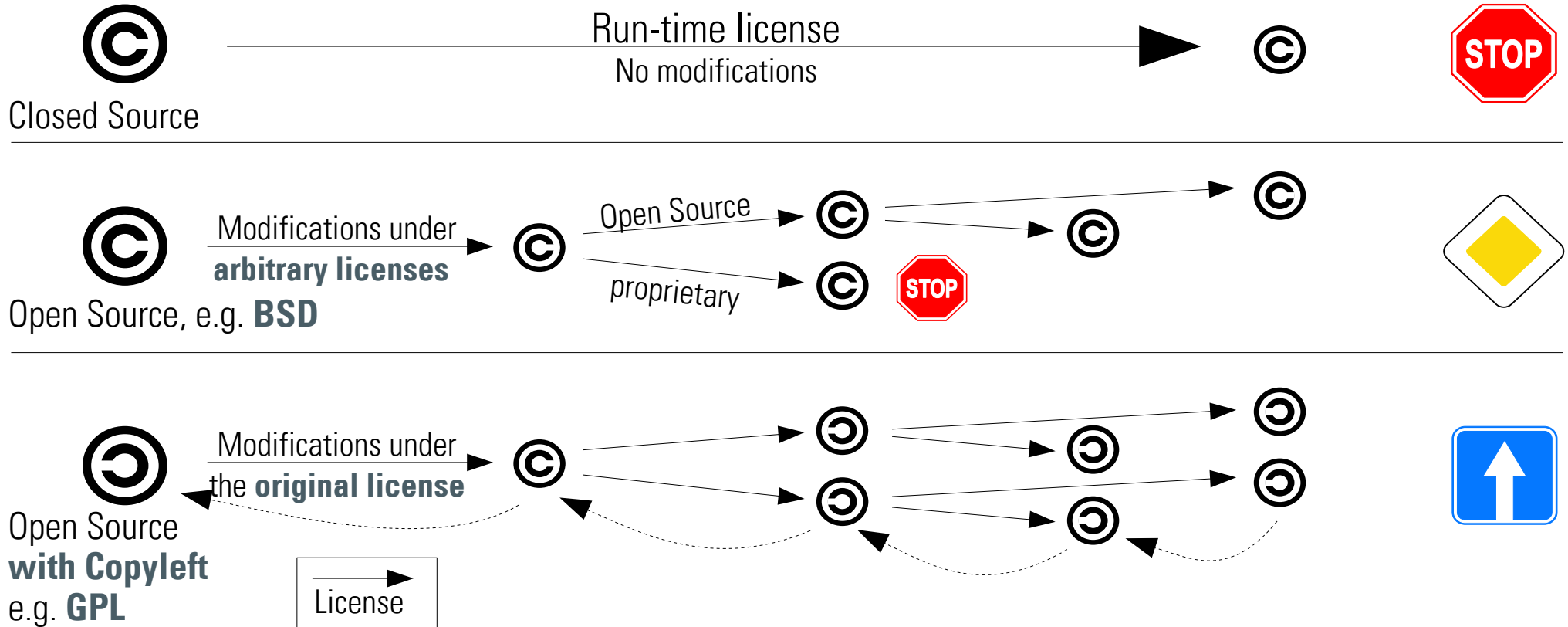
What is “Copyleft”?



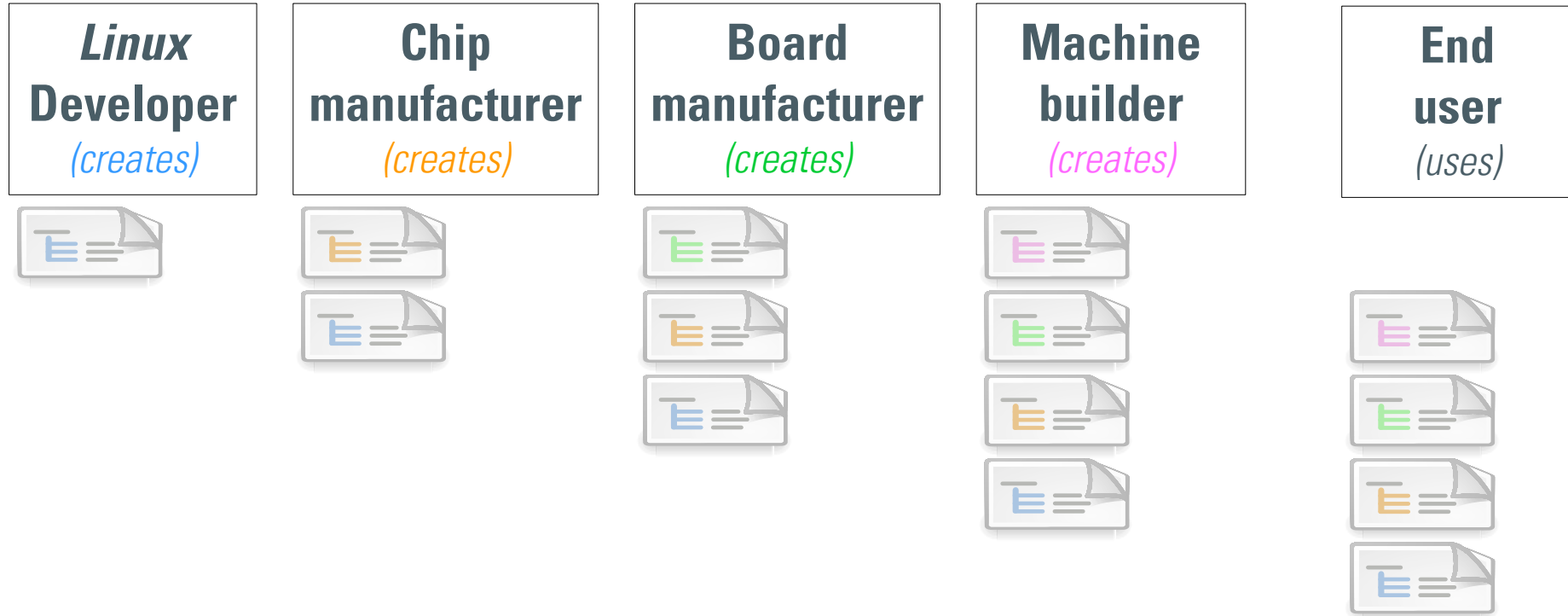
What is “Copyleft”?



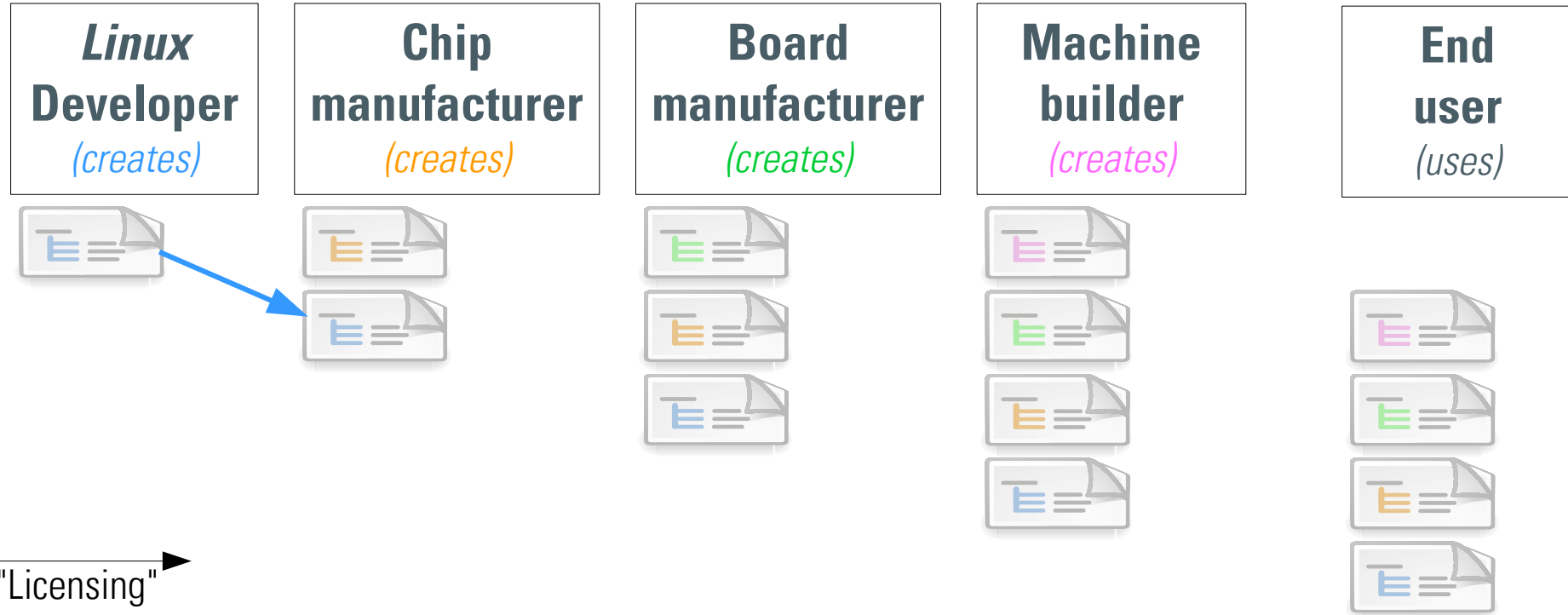
What is “Copyleft”?



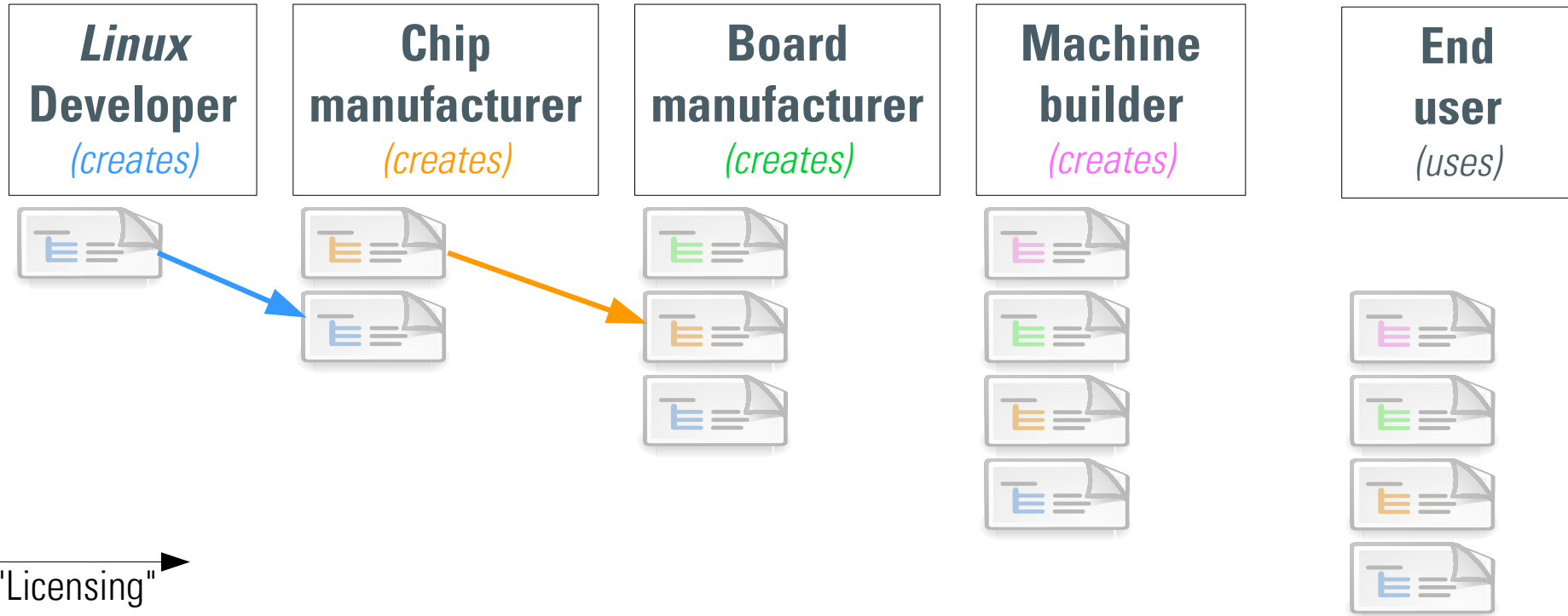
GPL software trade chain



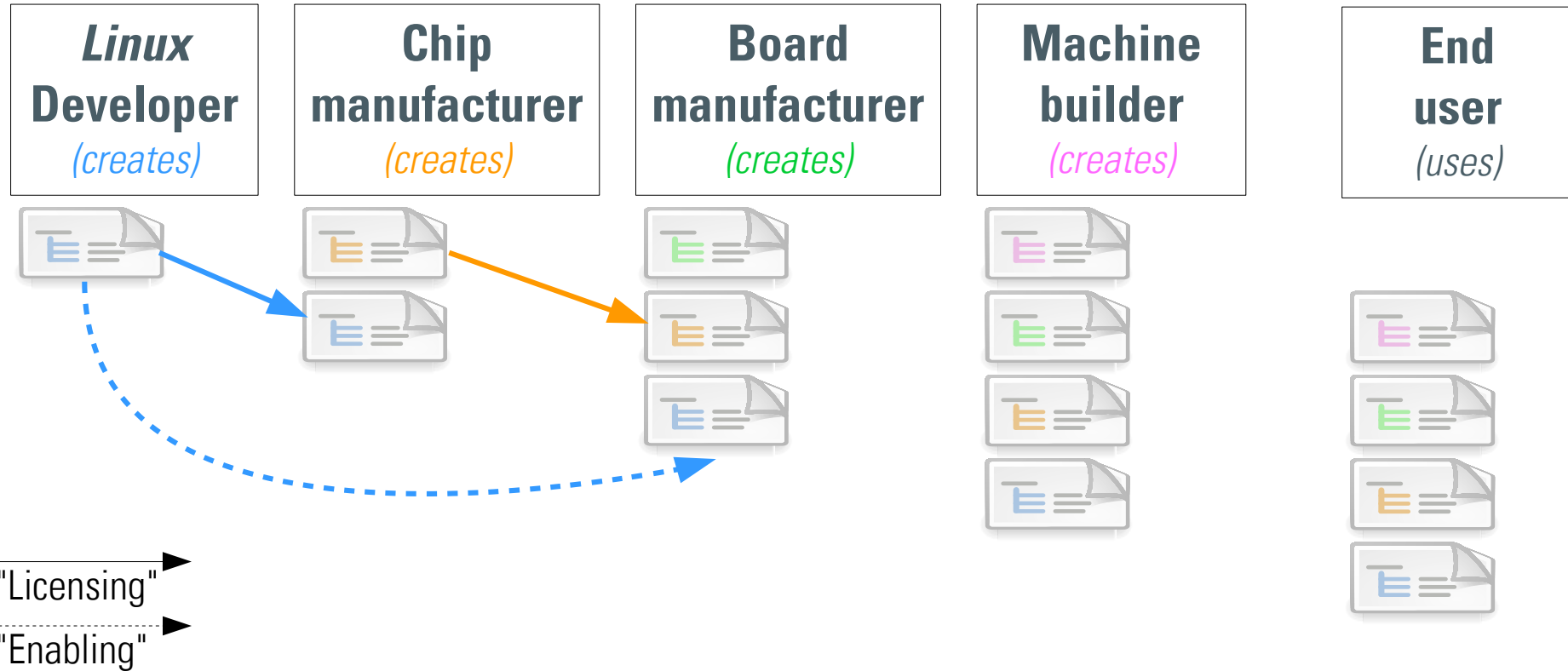
GPL software trade chain (1st step)



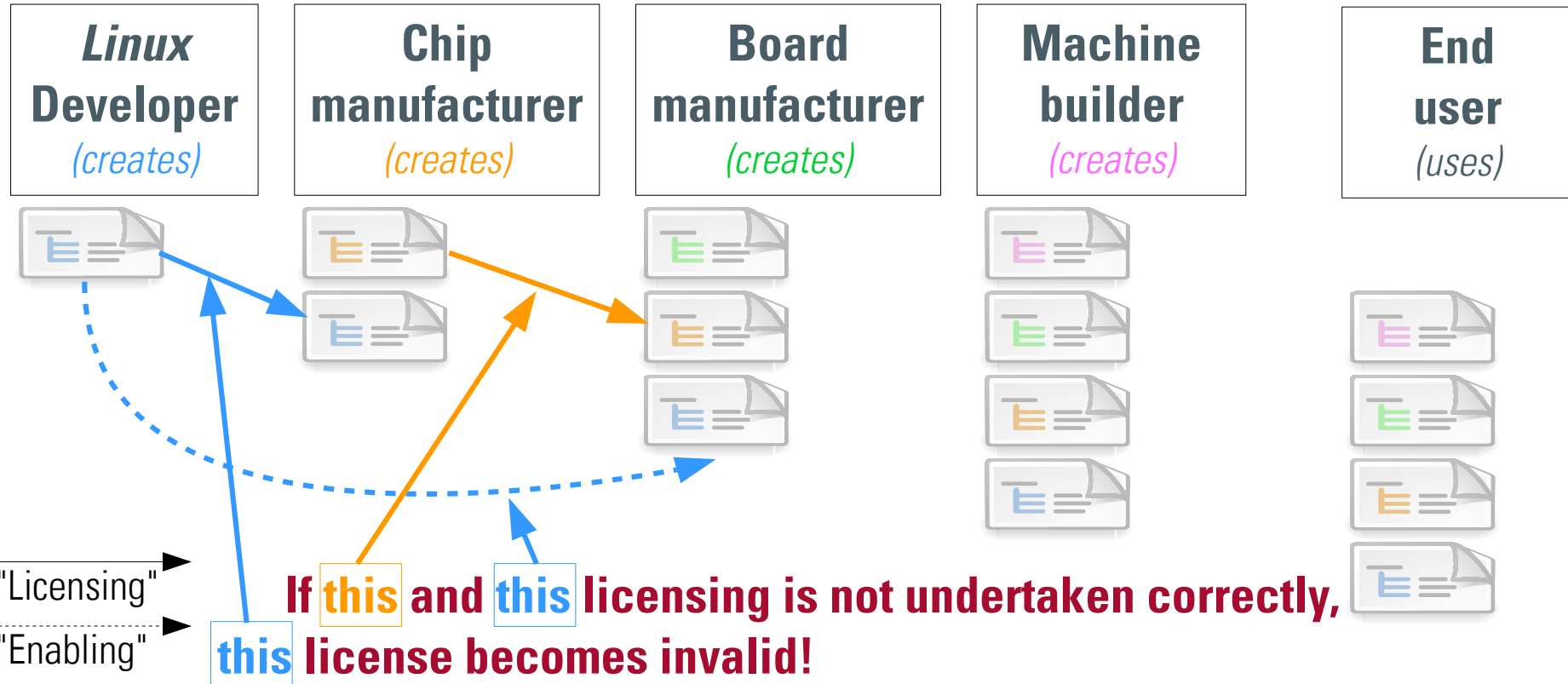
GPL software trade chain (2nd step)



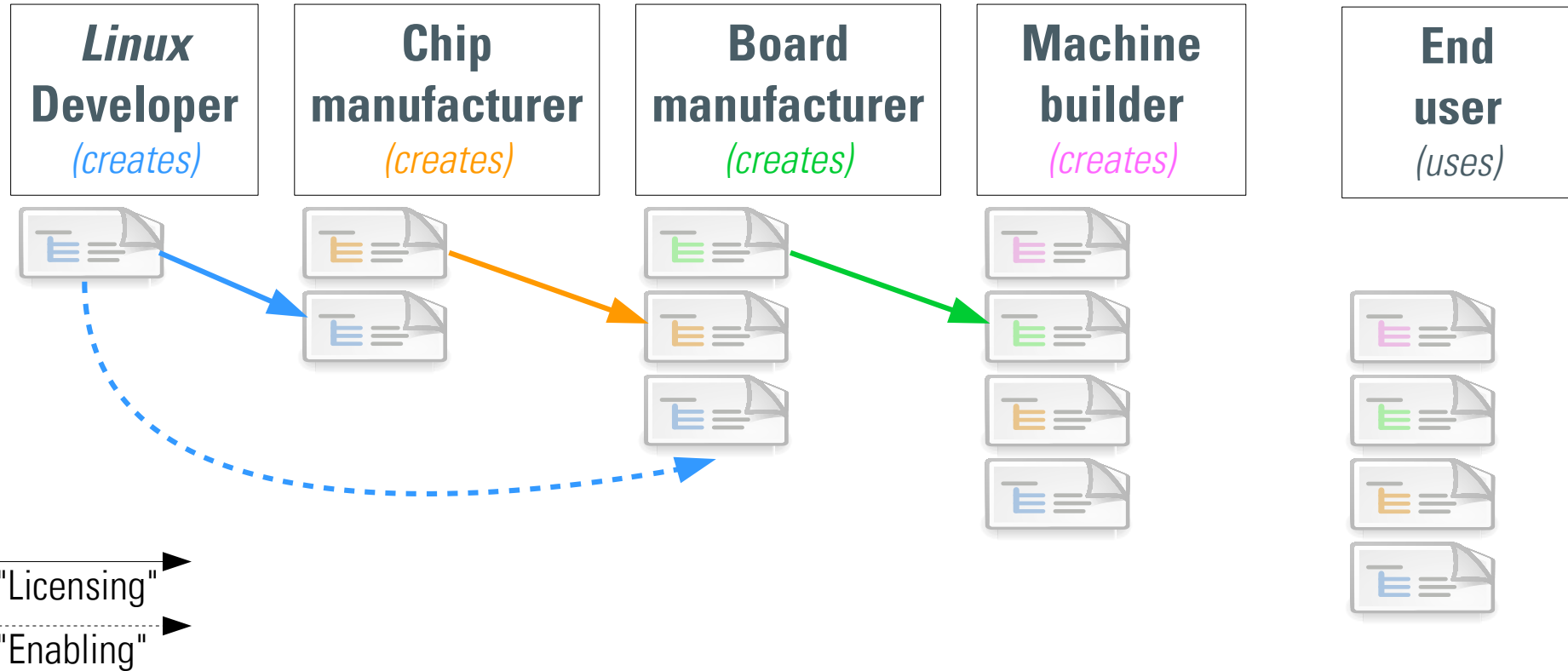
GPL software trade chain (2nd step)



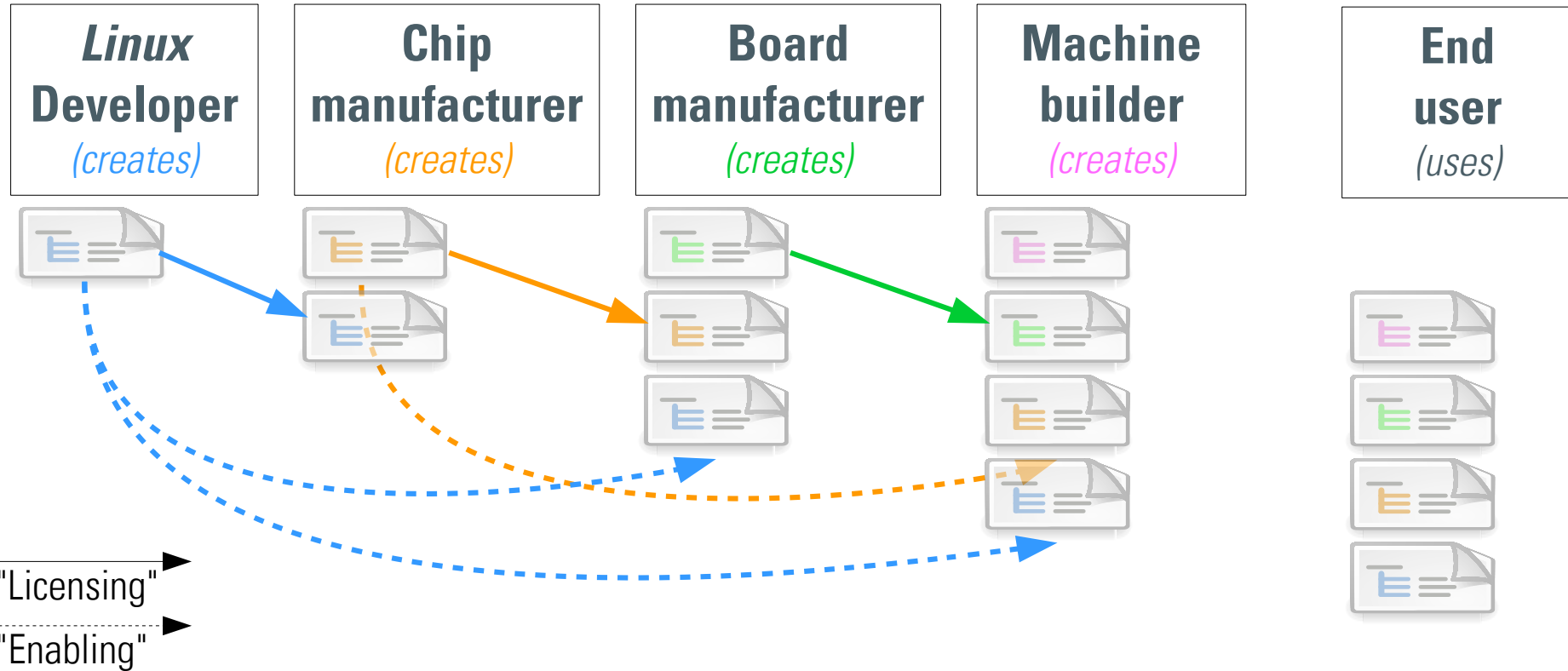
GPL software trade chain (2nd step)



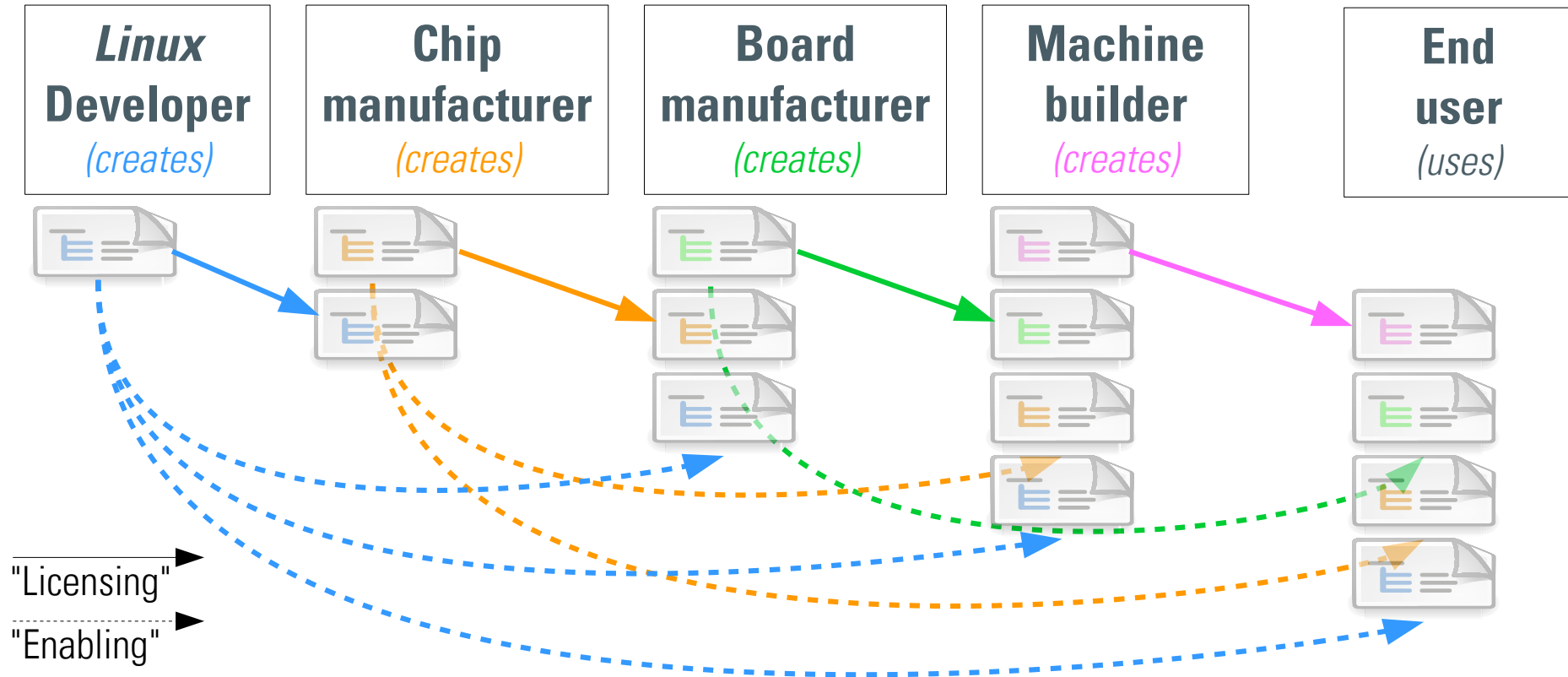
GPL software trade chain (3rd step)



GPL software trade chain (3rd step)



GPL software trade chain (all)



Who wrote the GPL?

„GPL“ or „GNU GPL“ is the abbreviation of “GNU General Public License”. The GPL was developed originally by Richard Stallman.

GPLv1: January 1989 by Richard Stallman

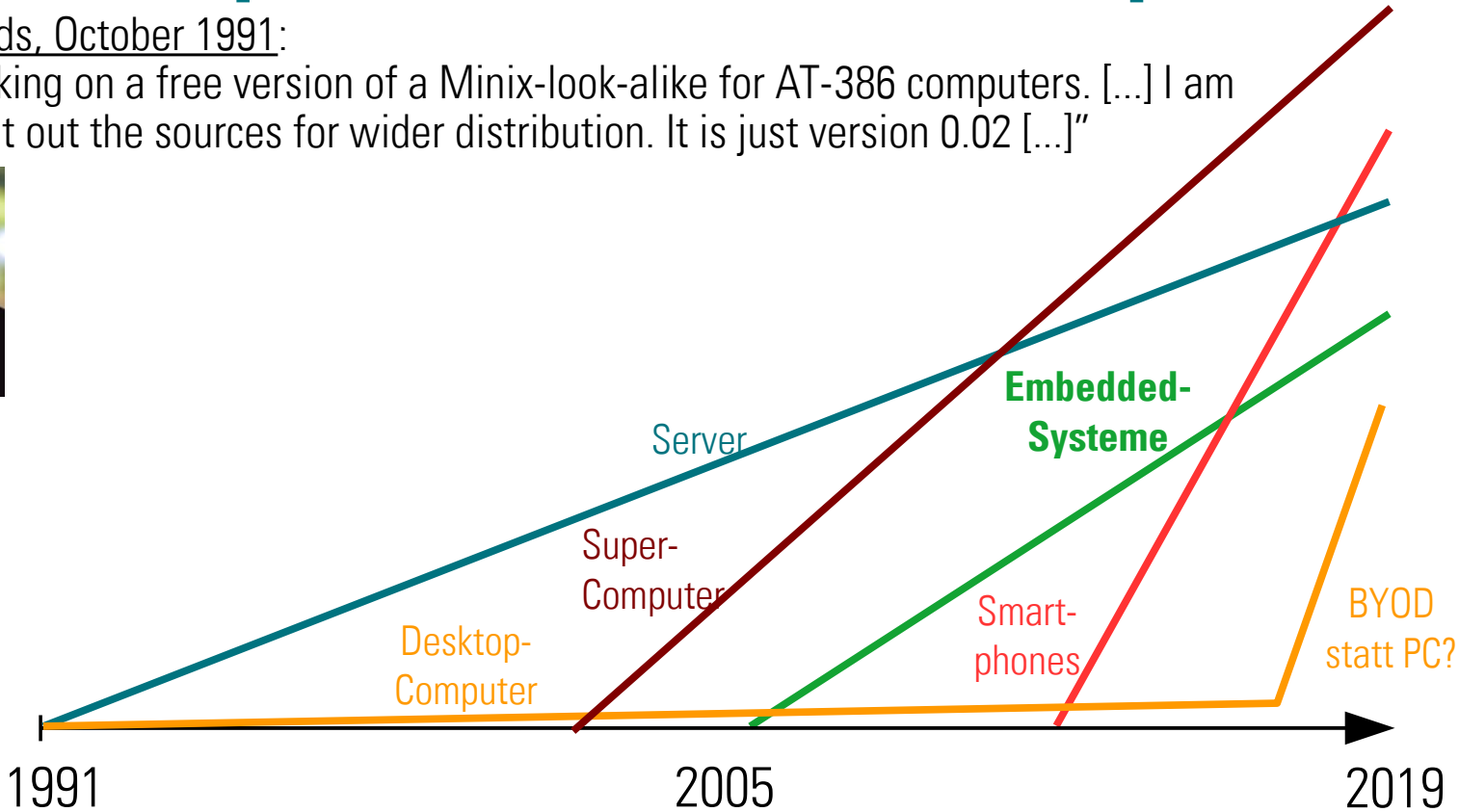
GPLv2: June 1991 by Richard Stallman

GPLv3: June 2007 by Richard Stallman, legal support from Prof. Eben Moglen (Columbia University, New York City) and the Software Freedom Law Center

Example of a successful GPL project (1)

Linus Torvalds, October 1991:

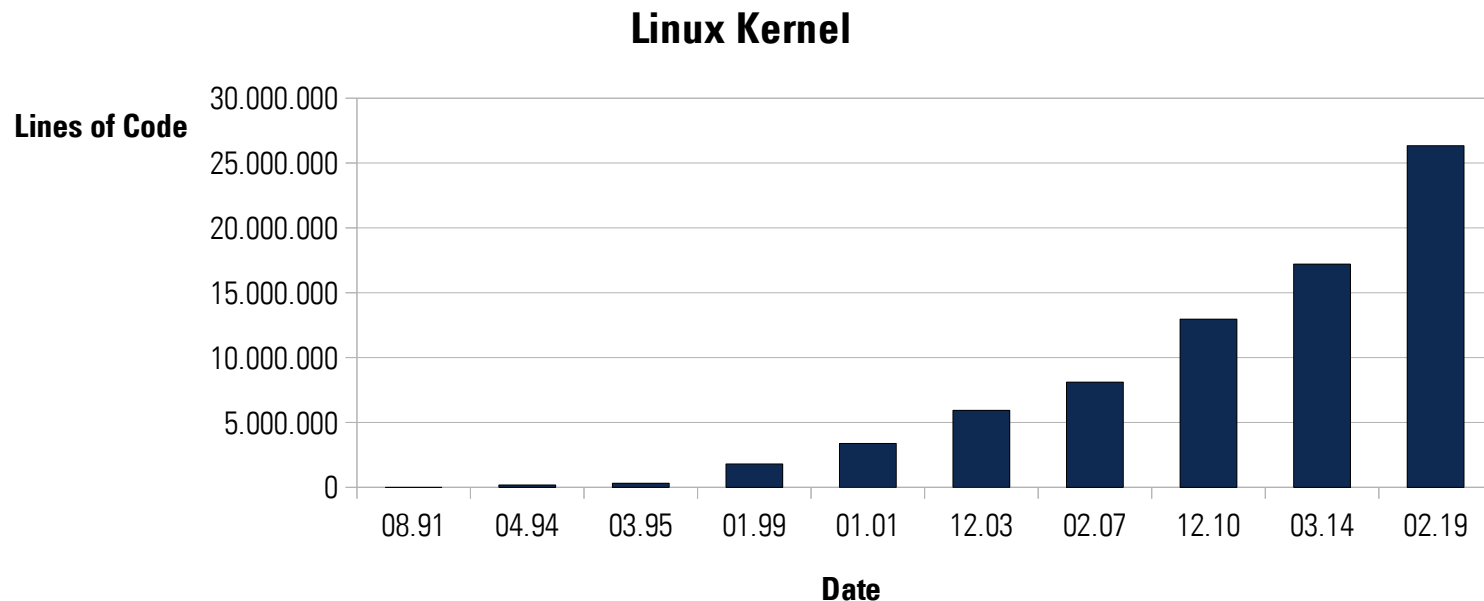
"[...] I'm working on a free version of a Minix-look-alike for AT-386 computers. [...] I am willing to put out the sources for wider distribution. It is just version 0.02 [...]"



Example of a successful GPL project (2)

Linus Torvalds, 1997:

"Making Linux GPL'd was definitely the best thing I ever did"



How Open Source may not work ...

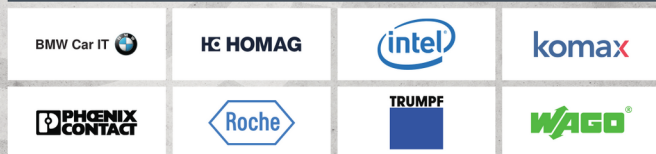
Project lead: "We need to get someone to develop a Linux driver for our new machine controller. Here's a purchase order for the work."

Purchase department: "Okay. Let me check the purchase order to make sure it follows our standard procedure: NDA to be executed, source going into company's safe ... "

Project lead: "Nope, we're going with Open Source - source gets published to the Internet."

Purchase department: "What? We're paying for the development but the source code goes out to the Internet? If **everybody** can use the source code, why are **we** the only ones paying for this? No way this can fly!"

GOLD MEMBERS



SILVER MEMBERS



BRONZE MEMBERS



AS OF FEBRUARY 2020

... but there is OSADL

OSADL initially was a purchase community of „Open Source“ Software.

OSADL promotes and coordinates the development of Open Source software for the machine, machine tool, and automation industry.

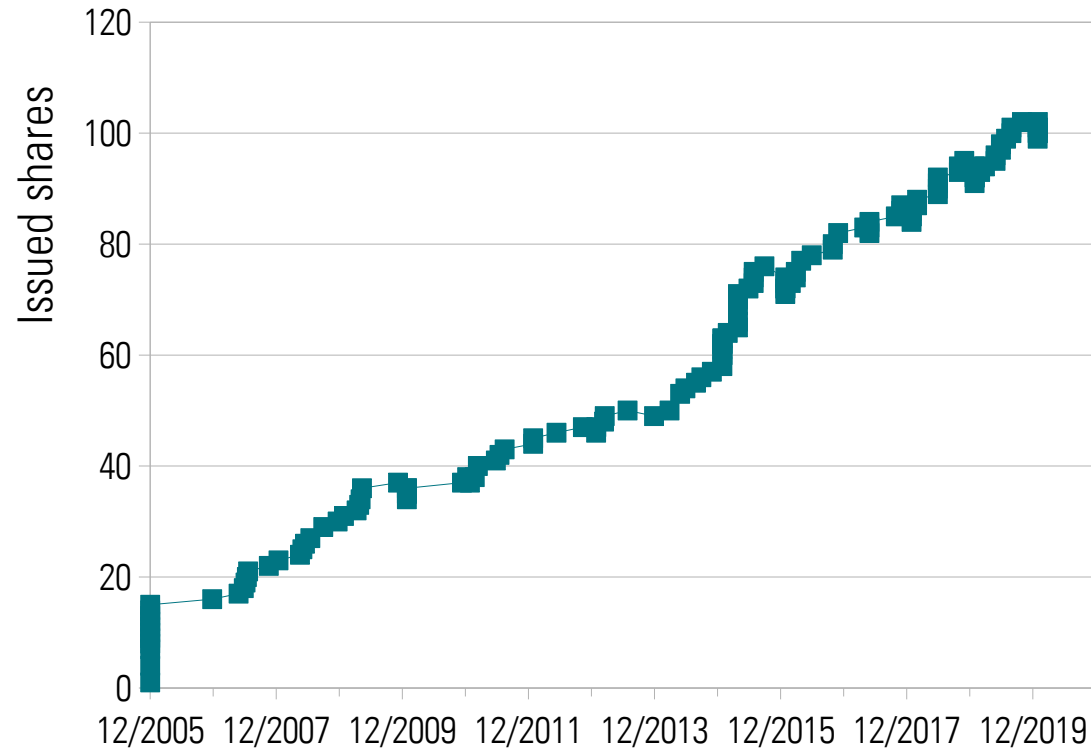
Today, OSADL uses the principle of “Open Innovation” for a lot more services:

- Legal support, legal assessment, best practices, audits
- Quality assessment and assurance of Open Source software, technical support
- Safety certification
- Networking, training courses, seminars
- Marketing

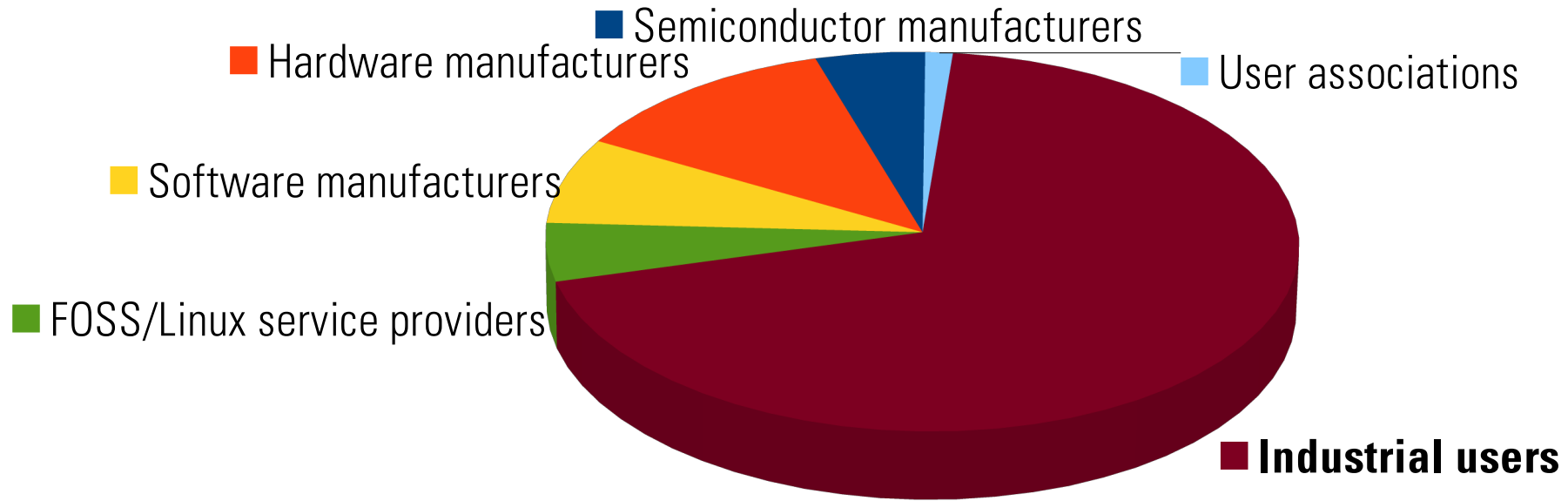
OSADL also has associate and academic members



OSADL members over time



OSADL Member Industries



Why Open Source?

The principle of Open Source is successful, since it is economically and technologically successful.

- Open Source is using the principle of Open Innovation for the development of **software base components**.
- However, before deciding to use a particular Open Source software under a particular license, an analysis must be undertaken to ensure that any involved proprietary company software may stay proprietary, if so desired.

You need more than just the source code

- To successfully deploy software, more than source code may be needed such as
 - ✓ Maintenance, tools
 - ✓ Quality assessment and assurance
 - ✓ Installation
 - ✓ Training, documentation
 - ✓ Legal advice
 - ✓ Safety certification
 - ✓ Patent clearance
- Open Source software is just the code, any **service** must be provided by the user, but this does not need to be done individually.
- Companies are well advised to join in a **community** to get access to everything that is needed to successfully deploy Open Source software.

Let's take a closer look at a license

A number of obligations must be fulfilled when copying and distributing a work that is licensed, for example, under the GNU General Public License v2, **GPL-2.0**:

- Information obligations
- Disclosure obligations
- Licensing obligations

GPLv2 and information obligations (distributing unmodified software)

1. **You may** copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, **provided that** you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

GPLv2 and information obligations (distributing modified software)

- a) **You must** cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) ...
- c) If the modified program normally reads commands interactively when run, **you must** cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

GPL and information obligations (Recommendations)

- **Non-interactive program**
 - License notice and copyright notices in the manual
 - GPL text in the appendix of the manual
- **Interactive program**
 - License notice and copyright notices in the manual
 - Reference in “About box”
 - GPL text in the appendix of the manual

GPLv2 and source code disclosure

3. **You may** copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above **provided that you** also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange [..]

GPL and source code disclosure (Recommendations)

- Option 1 **“Accompany with every delivered device”**
 - simple
 - safe
 - may technically not be feasible
 - recommended (whenever possible)
- Option 2 **“Make it available on request”**
 - for 3 years (GPLv2)
 - for 3 years + duration of device support (GPLv3)
 - relatively expensive
 - recommended, if option 1 not possible

GPL and derivative work (1)

What is “derivative work”?

The term “derivative work” relates to the fact that a work may contain significant **elements of another work**. If this is the case, the new work is called “derivative” of the other work, and the **copyright of the original author** must be recognized. Otherwise, the work is an original new work, and the author may independently decide under which condition, if any, the work is licensed.



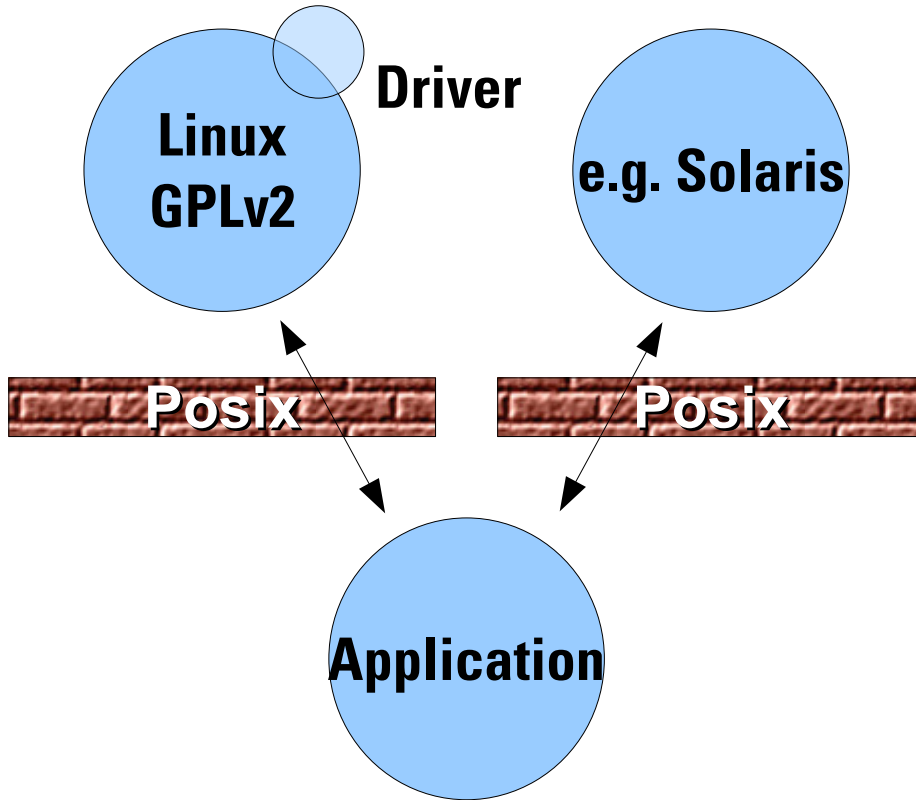
GPL and derivative work (2)

What is the significance of “derivative work” in the context of the GPL?

If a particular software project forms a derivative work of existing software that is licensed under a GPL, this project must be **licensed under the same or compatible** GPL, *i.e.* all information and disclosure obligations must be obeyed.

If this is not the case, the author may independently decide how the project is licensed.

Linux, GPLv2 and derivative work (3)



Linux driver:

Is considered derivative work, since it

- runs in the same memory space
- uses exclusive kernel interfaces

Application with POSIX interface:

Is not considered derivative work, since it

- runs in a different memory space
- is assigned as such by the author

GPL and derivative work (Recommendations)

Project	Requires	GPL of the project?
Application	Linux kernel	No
Driver	Linux kernel	Yes
Application	GPL library	Yes
Application	LGPL library	No
Application	GPL compiler	No
Web application	GPL web server	No

Conclusion

“Open Source software” is short for “software that is licensed under a listed or otherwise verified Open Source license”.

Whoever is **copying and distributing** Open Source software, must know the **license conditions** and obey these contractual obligations – this is not different from proprietary software.

The source code of Open Source software is developed in a **community** – everything else that is needed to successfully deploy Open Source software should also be developed in a community.