

Legal Aspects Open Source Players Should Know About

Caren Kresse

Open Source Automation Development Lab (OSADL) eG



Legal Aspects Open Source Players Should Know About
HeiChips Summer School, 3 August 2026
Caren Kresse, OSADL eG



Legal Aspects Open Source Players* Should Know About

*** at least, if they want to distribute or publish
Open Source in industry or academia**

Caren Kresse

Open Source Automation Development Lab (OSADL) eG



REGULAR MEMBERS *GOLD*

A4nXT		CARIAD Audi Research	Everlence	
	HE HOMAG	komax	PHOENIX CONTACT	TRUMPF
WAGO				

REGULAR MEMBERS *BRONZE*

					bachmann	BALLUFF	
BELIMO		BIZERBA	B&R	BWI	Christ	CODESYS	Calsonik
	Cornelsen		DEMICON	DEUTZ-WEISSE	Guogon	F.T.N	Elektrobit
Endress+Hauser	ETAS	FESTO	RECHER	Fraunhofer	Franklin	gantner	GINZINGER
HEIDELBERG	hilacher	HIRSCHMANN	Husqvarna Group	ifm	ikerlan	IMMS	interflex
	KEB	KEBA	Kellermann	KOENIG & BAUER	kontron	KOSTAL	KUKA
Leica	Leuze	LIEBHERR	LINTECHNIK	Mahr	MEERWEG	MicroSys	Miele
	Pangutronix	PEPPERLAWSON	PHYTRIC	PILZ	Politec	PROTEC	protonic
RATIONAL	REVOLUTION	rexroth	ROCKWELL	SABO	SAUTER	Schneider	
SENSORPART	SEW	SICK	Sirona	strattec	STW	System	
TEXAS INSTRUMENTS	TOPCON	TECLOGIC	TRIA	IQ	VELANT GROUP	VEGA	VIESSMANN
VOITH	VOLLMEIER	WÄCHNER	Wendlinger	wengler	WESTERMO	WIBU SYSTEMS	WIKAL
YASKAWA							

OSADL is the community for Open Source in industry

OSADL uses the principle

“develop once for the use of many”

for services that are needed to successfully use Open Source software in industrial products:

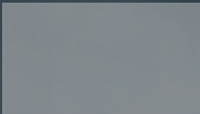
- Shared development of Open Source software for the machine, machine tool, and automation industry.
- Legal support, legal assessment, best practices, audits
- Quality assessment and assurance, technical support
- CRA support
- Safety certification
- Networking, training courses, seminars

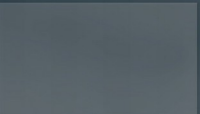
Associate and academic OSADL members



ASSOCIATE MEMBERS *GOLD*







ASSOCIATE MEMBERS *BRONZE*

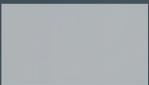




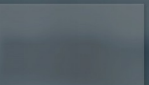












ASSOCIATE MEMBERS *BASE LEVEL*





















ASSOCIATE MEMBERS *ORGANIZATION*









AS OF MAY 2020



ACADEMIC MEMBERS



























































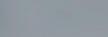






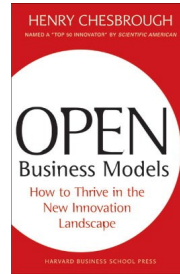






AS OF JUNE 2020

Open Innovation: The economic background of Open Source



Henry Chesbrough

- Standardization of hardware and software interfaces
- Larger base of knowledge, expertise and experience
- Avoiding unnecessary parallel development
- Reduction of development cycles
- Participation of users at the innovation process

**Combined value is
created through
open knowledge
economics**

Open Innovation: The economic background of Open Source

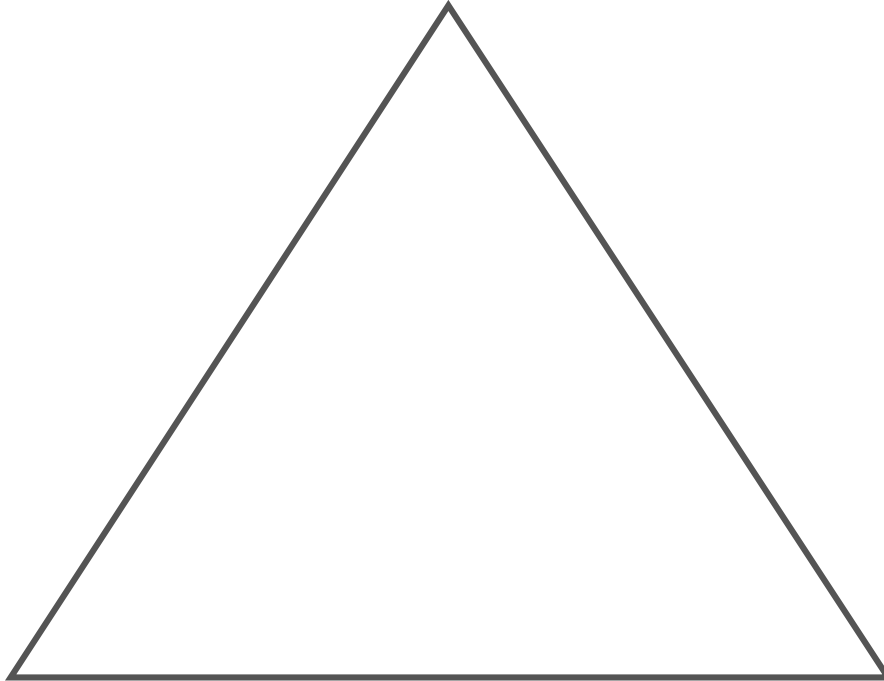


Cost reduction by avoiding parallel development

- Standardization of hardware and software interfaces
- Larger base of knowledge, expertise and experience
- Avoiding unnecessary parallel development
- Reduction of development cycles
- Participation of users at the innovation process

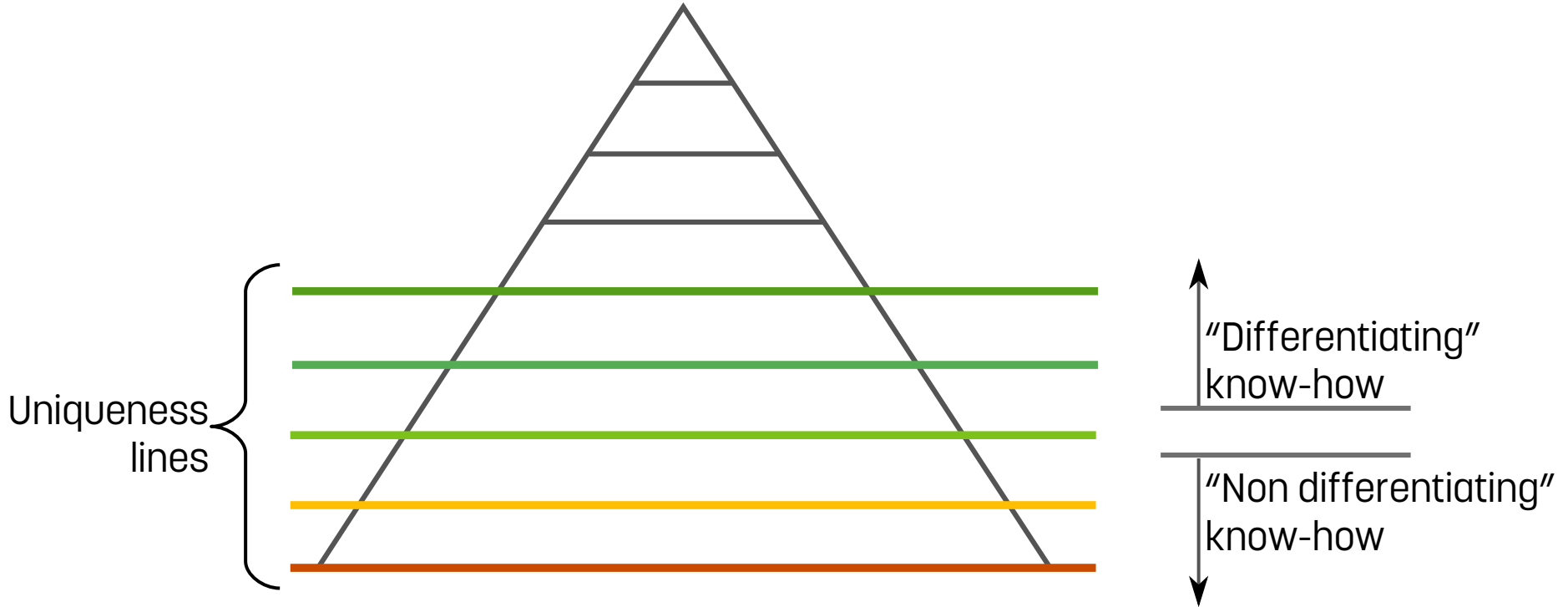
Combined value is created through open knowledge economics

Open Source the world?



Open Source the world?

The pyramid of differentiation



Open Source the world?

The pyramid of differentiation

Ample space below the uniqueness line

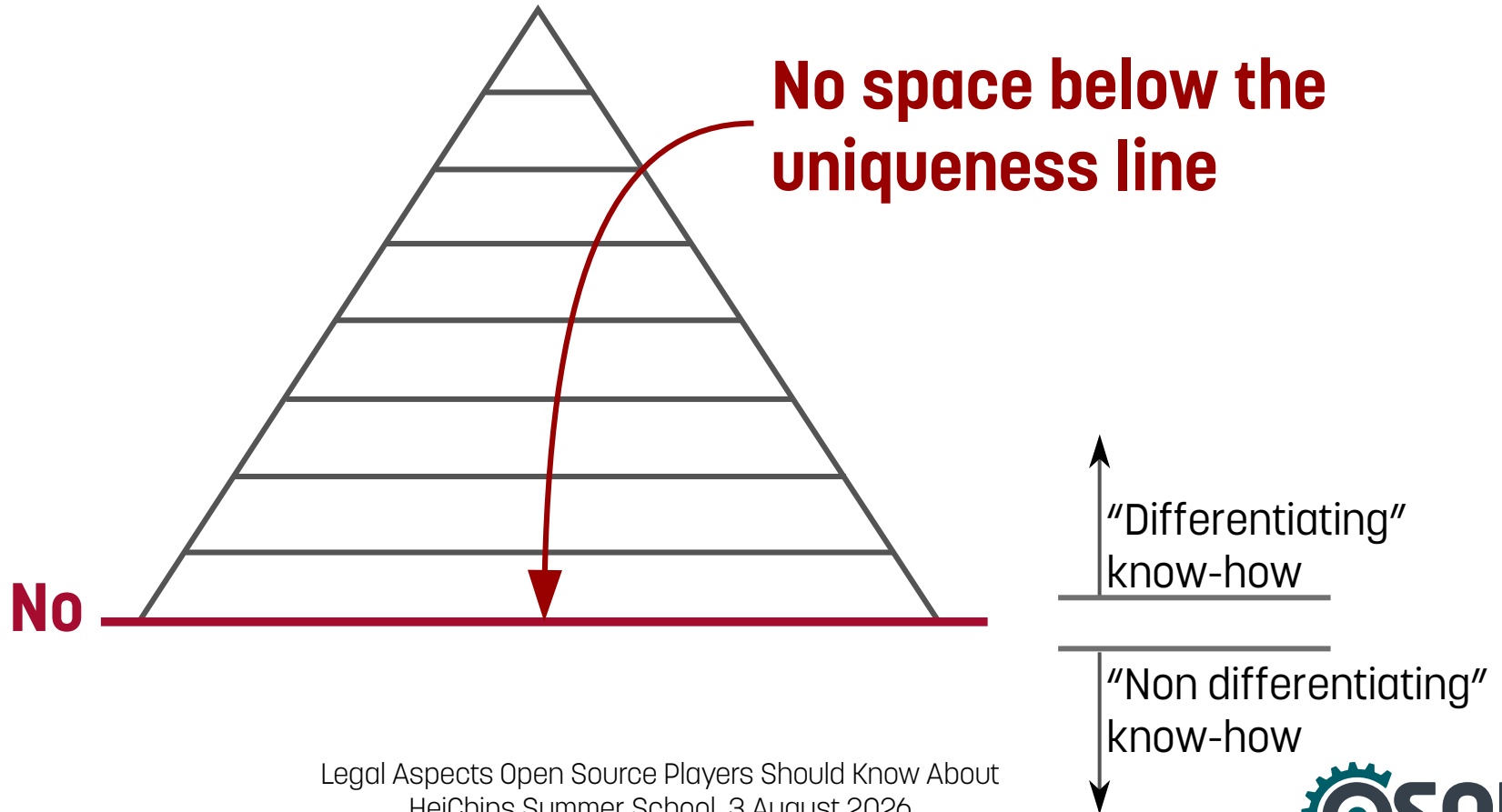
Yes

“Differentiating”
know-how

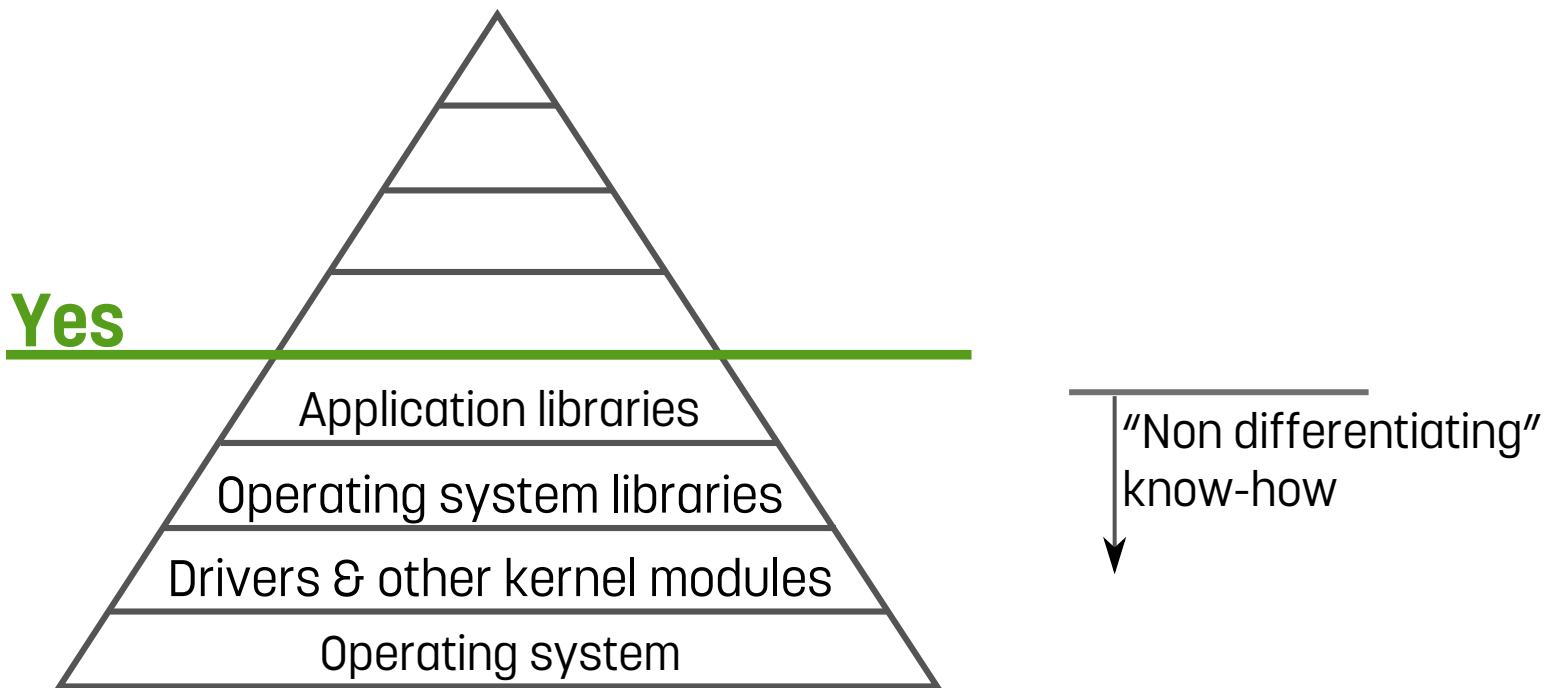
“Non differentiating”
know-how

Open Source the world?

The pyramid of differentiation



Open Source in industry: An example

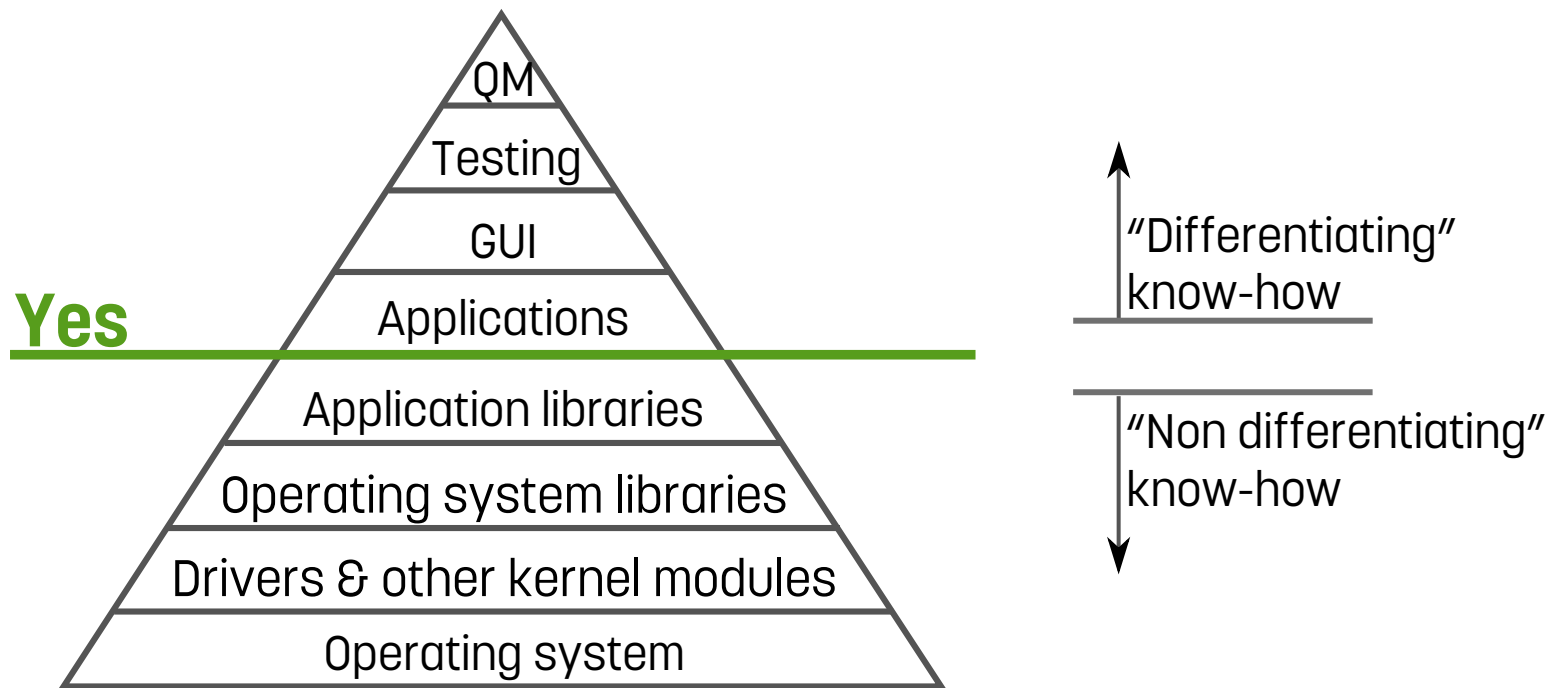


The pyramid of differentiation

Example: Machine software

Legal Aspects Open Source Players Should Know About
HeiChips Summer School, 3 August 2026
Caren Kresse, OSADL eG

Open Source in industry: An example

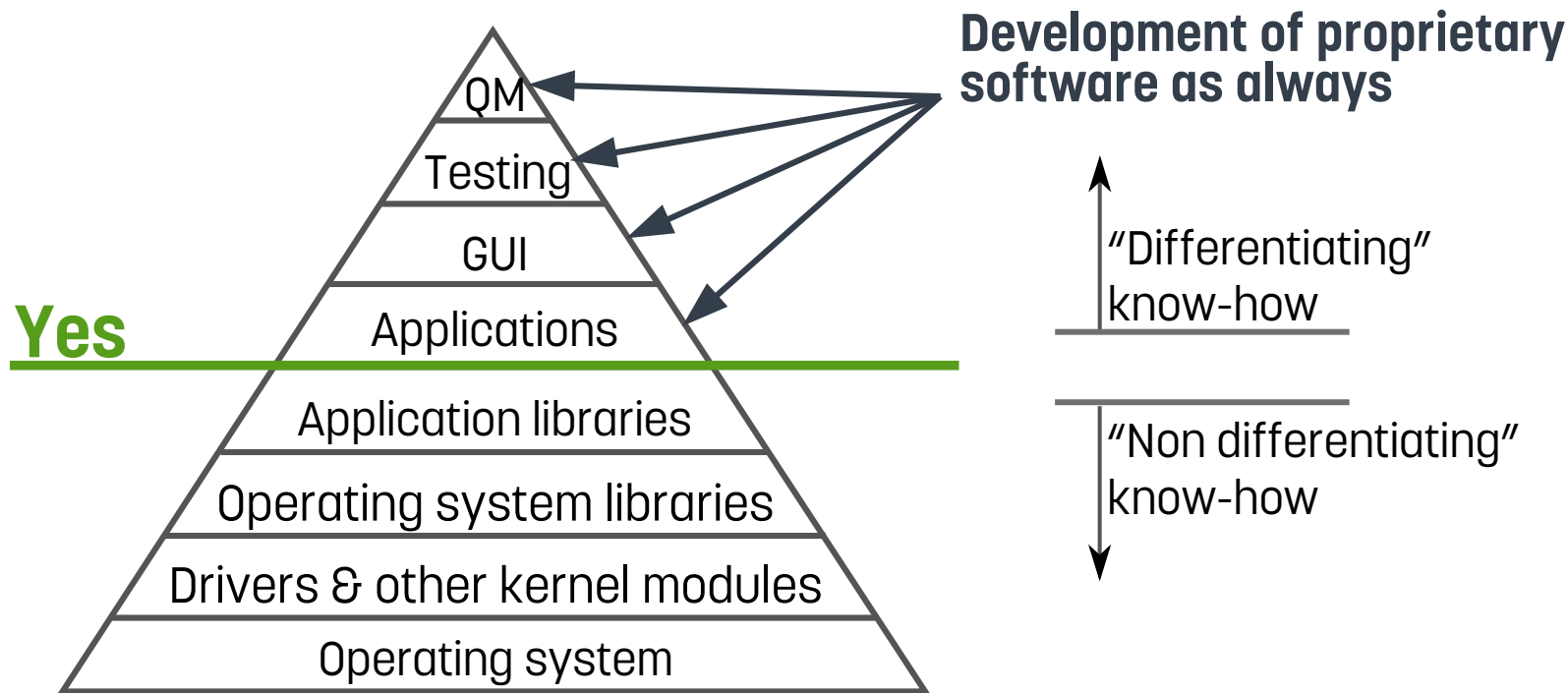


The pyramid of differentiation

Example: Machine software

Legal Aspects Open Source Players Should Know About
HeiChips Summer School, 3 August 2026
Caren Kresse, OSADL eG

Open Source in industry: An example

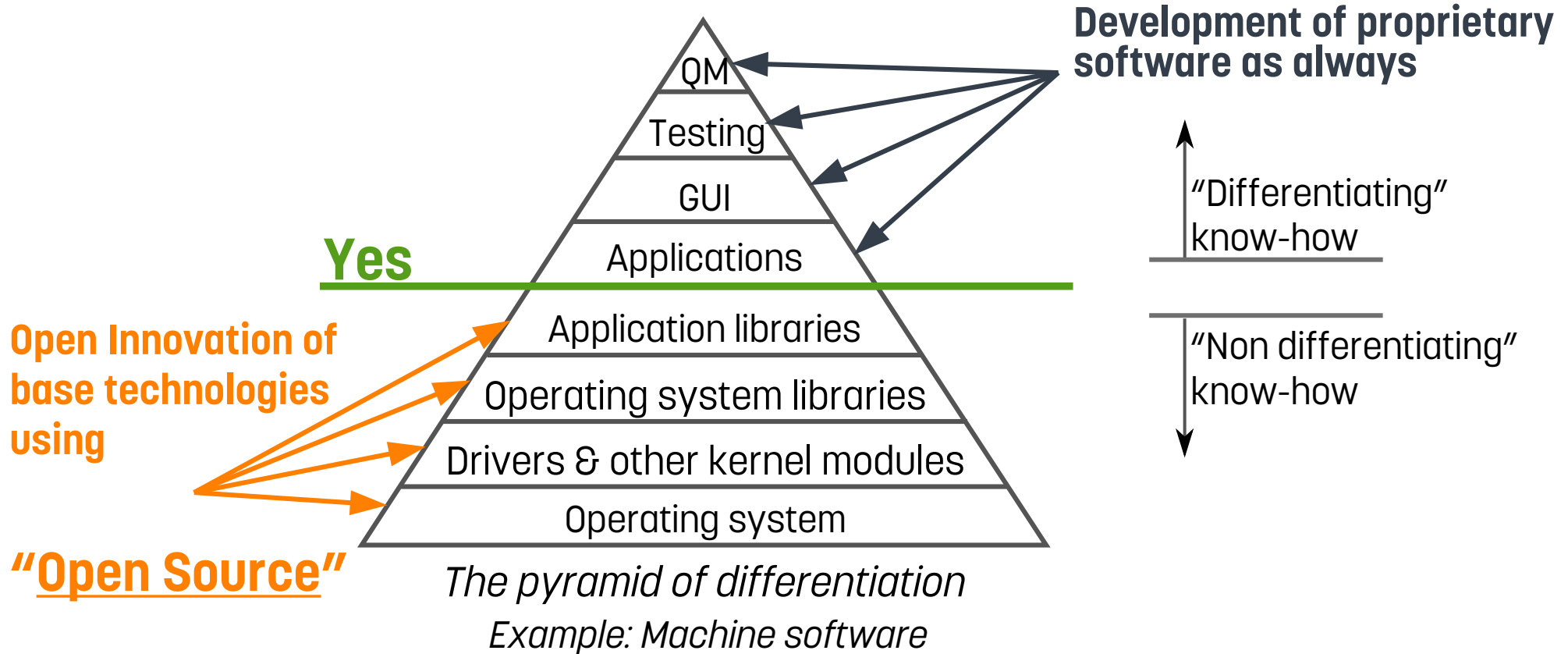


The pyramid of differentiation

Example: Machine software

Legal Aspects Open Source Players Should Know About
HeiChips Summer School, 3 August 2026
Caren Kresse, OSADL eG

Open Source in industry: An example



What is 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"**.

What is Open Source software? (2)

- **“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.
- A software is “Open Source” if it is licensed under an **Open Source license**.
- A software license may only be called “Open Source” if it fulfills certain **requirements**, for example given in the Open Source Definition (OSD, <https://opensource.org/OSD>) released by the Open Source Initiative (OSI).

Example: MIT license

Unrestricted and unconditional rights of use

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Copying and distribution under liberal conditions

Proprietary vs. Open Source software

Proprietary software

Open Source software

Proprietary vs. Open Source software

Action:

Proprietary software

Open Source software

Run software:

**Analyze and
modify software:**

Action:	Proprietary software	Open Source software
Run software: Analyze and modify software:		

Proprietary vs. Open Source software

Action:	Proprietary software	Open Source software
Run software:		
Analyze and modify software:		
Copy and distribute unmodified software:		
Copy and distribute modified software:		

Proprietary vs. Open Source software

Action:	Proprietary software	Open Source software
Run software:	Default: Prohibited	Default: Permitted
Analyze and modify software:		
Copy and distribute unmodified software:		
Copy and distribute modified software:		

Proprietary vs. Open Source software

Action:

Proprietary software

Open Source software

Run software:

Permitted

by contract (EULA, ToU, ...)

Default: Permitted

by unilateral declaration of will of the authors through **access to the software**

Analyze and
modify software:

Default: Prohibited

Copy and distri-
bute unmodified
software:

Copy and distri-
bute modified
software:

Proprietary vs. Open Source software

Action:

Proprietary software

Open Source software

Run software:

Permitted

by contract (EULA, ToU, ...)

Default: Permitted

by unilateral declaration of will of the authors through **access to the software**

Analyze and
modify software:

Default: Prohibited

Copy and distri-
bute unmodified
software:

Default: Prohibited

by copyright law

Default: Prohibited

by copyright law

Copy and distri-
bute modified
software:

Proprietary vs. Open Source software

Action:	Proprietary software	Open Source software
Run software:	Permitted by contract (EULA, ToU, ...)	Default: Permitted by unilateral declaration of will of the authors through access to the software
Analyze and modify software:	Default: Prohibited	
Copy and distribute unmodified software:	Permitted by license	Permitted by license
Copy and distribute modified software:	Default: Prohibited by copyright law	

Proprietary vs. Open Source software

Action:	Proprietary software	Open Source software
Run software:	Permitted by contract (EULA, ToU, ...)	Default: Permitted by unilateral declaration of will of the authors through access to the software
Analyze and modify software:	Default: Prohibited	
Copy and distribute unmodified software:	Permitted by license	Permitted by license
Copy and distribute modified software:	Default: Prohibited by copyright law	

e.g. runtime license

Open Source license

Copyright Law



Legal Aspects Open Source Players Should Know About
HeiChips Summer School, 3 August 2026
Caren Kresse, OSADL eG



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?

- Must be created by a **human being**, *i.e.* a natural person.
- Animals, machines, computers, **AI** cannot create protected works by themselves.



Protected



"Monkey selfie"

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

#define ALPHABET 26

int main(void) {
    int *freq = calloc(ALPHABET, sizeof(int));
    if (!freq) {
        perror("Memory allocation failed");
        return 1;
    }

    printf("Enter text (Ctrl+D to finish):\n");

    int ch;
    while ((ch = getchar()) != EOF) {
        if (isalpha(ch)) {
            freq[tolower(ch) - 'a']++;
        }
    }
}
```

AI code

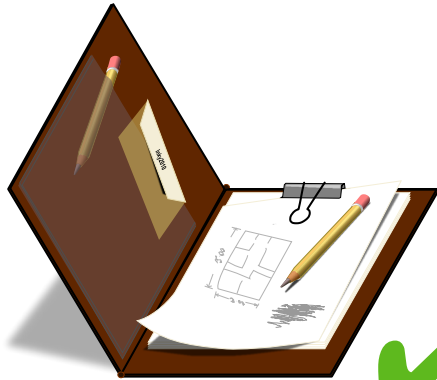


Not protected

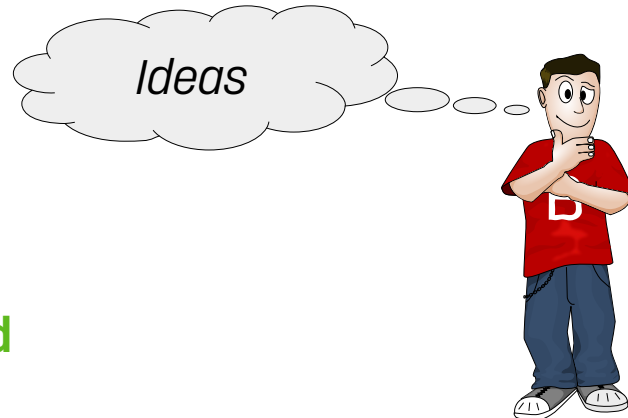
What is a work? (2)

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).



Protected



Not protected

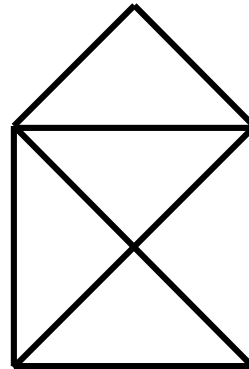
What is a work? (3)

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

- Has to be new or different.
- Must reflect a recognizable individuality of the creator.
- In case of accidental parallel creation of identical works, both are protected by copyright (can be difficult to prove in practice).



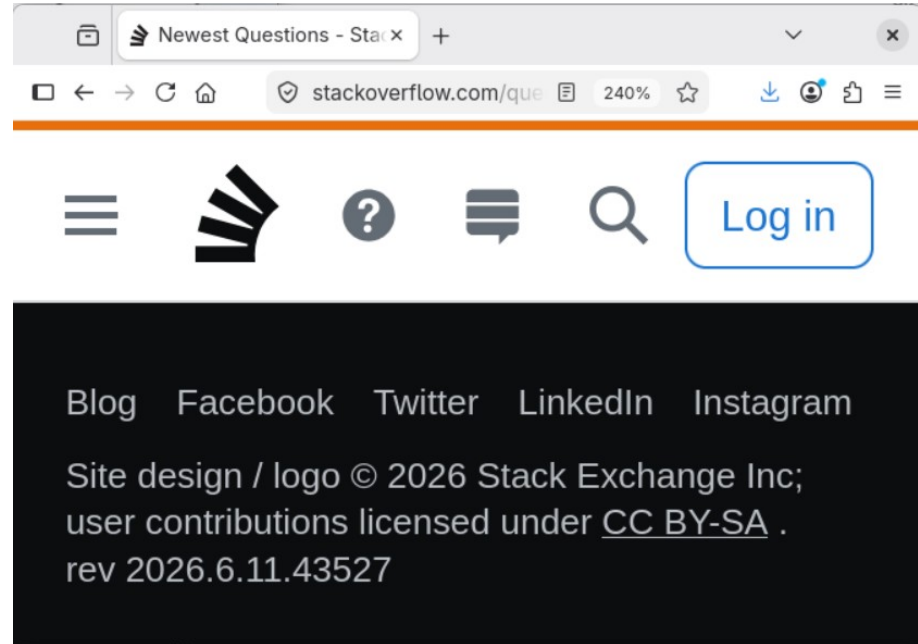
Protected



Not protected

Threshold of originality

- Copyright protection principle of the “small coin”: When in doubt, a code snippet is protected.
- E.g. Stackoverflow:



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**.

The creator of a work has two different rights

Moral rights of authorship

- The author has **the right to claim authorship** of their work.
- 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 transferred** in any way.

Exclusive rights of use

- The author of a work has exclusive rights of use.
- Rights of use can be **granted to third parties** (exclusively or limited).
- When a work is created in the course of employment, the rights of use **automatically transfer to the employer**.

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

Author is a private person:

Copyright © 2026 Jane Doe

Author is an employee:

Copyright © 2026 Employer LLC, author Jane Doe

Software is sold:

Copyright © 2026 Buyer LLC, author Jane Doe

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

Author is a private person:

Copyright © 2026 Jane Doe

Author is an employee:

Copyright © 2026 Employer LLC, author Jane Doe

Software is sold:

Copyright © 2026 Jane Doe

**What about academic forms of
"employment"?**

Copyright law prohibits ...

... among others **copying, distributing and publicly performing** protected works.

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

- **Legal permissions**, *e.g.* copies made during proper use, copies made for text and data mining, backup copies, ...
- **Licenses** from holders of exclusive rights of use, *e.g.* proprietary run-time licenses or Open Source licenses.

License rules of thumb

“License” from licere (“to be allowed, be allowable”)

- 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.

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
 - ♦ If applicable, damages (e.g. dual licensing)
 - ♦ Obligation to inform all customers to whom the unauthorized copies have been conveyed
 - ♦ Provision of complete customer lists who received unauthorized copies
 - ♦ Physical destruction of unauthorized copies
- Penalties may be imposed for **infringement of copyright law**.

What else ...

- Free Software
- Open Source Hardware
- Types of Open licenses
- Derivative works and Copyleft
- AI and copyright
- Open Source and the CRA

What is Free Software?

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

What is Open (Source) Hardware?

- **Design specification** and **documentation** for hardware that is licensed in accordance with **Open Source principles**.
- Often, Open Source licenses or more generic Open licenses (Creative Commons) are used; but there are some specific Open Hardware licenses.

Types of Open licenses

- **Permissive** (e.g. MIT, BSD-[0234]-Clause, Apache)
- **Copyleft** (e.g. GPL, AGPL, EUPL)
- **Restricted Copyleft** (e.g. LGPL, MPL, EPL)
- **Open Data Licenses** (e.g. Odbi, ODC-By)
- **Open Hardware licenses** (e.g. CERN Open Hardware Licenses)
- **For other works** (e.g. CC-licenses)

Types of Open licenses

- **Permissive** (e.g. MIT, BSD-[0234]-Clause, Apache)
- **Copyleft** (e.g. GPL, AGPL, EUPL)
- **Restricted Copyleft** (e.g. LGPL, MPL, EPL)
- **Open Data Licenses** (e.g. Odbf, ODC-By)
- **Open Hardware licenses** (e.g. CERN Open Hardware Licenses)
- **For other works** (e.g. CC-licenses)

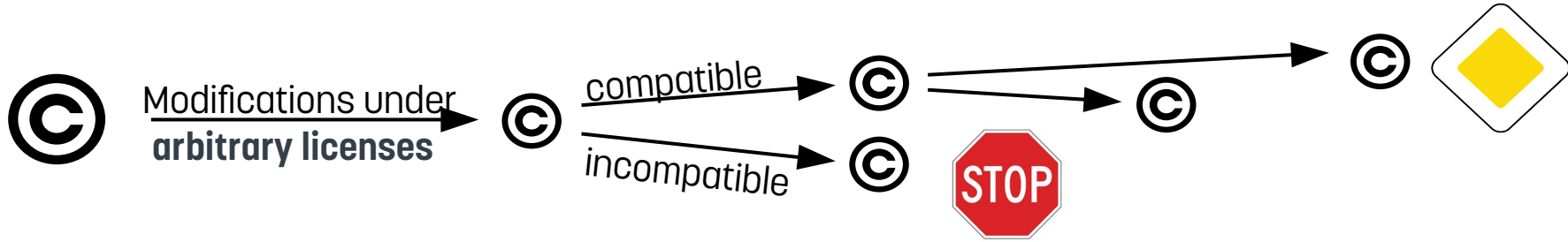
What is a derivative work?

- A work may contain significant **elements of another work**.
- If this is the case, the new work is called a **“derivative”** of the original work.
- The **copyright of both authors** must be recognized.
- Such a derivative work can only be copied and distributed if the license conditions of both authors are **compatible**.



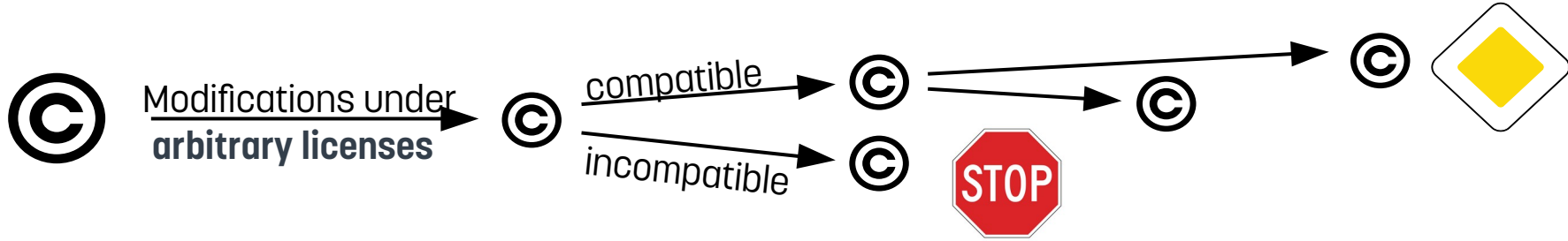
What is Copyleft?

Permissive licenses

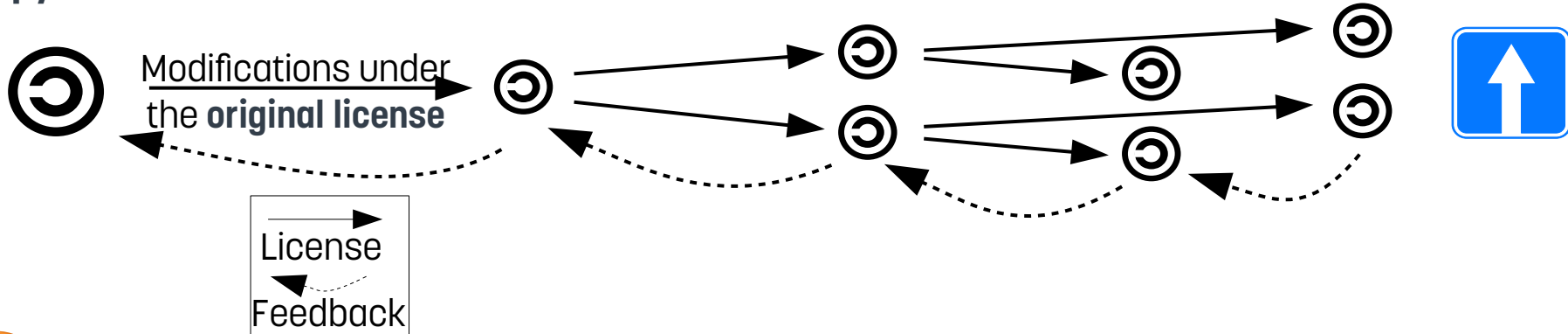


What is Copyleft?

Permissive licenses



Copyleft licenses



What is Copyleft?

Permissive licenses



Modifications under
arbitrary license

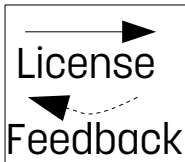
**Restricted (or weak) Copyleft
licenses make exceptions for certain
types of modifications, e.g. linking.**



Copyleft licenses

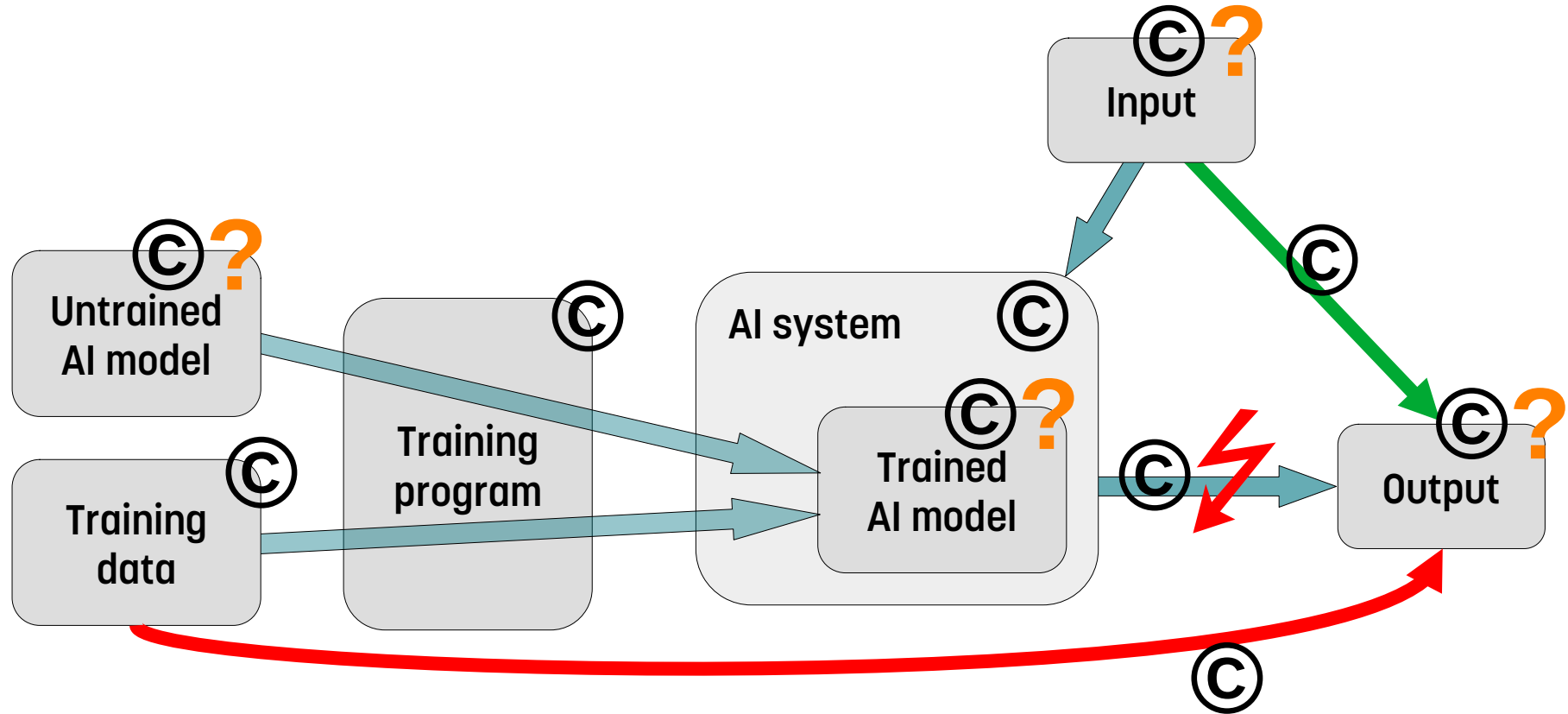


Modifications under
the **original license**



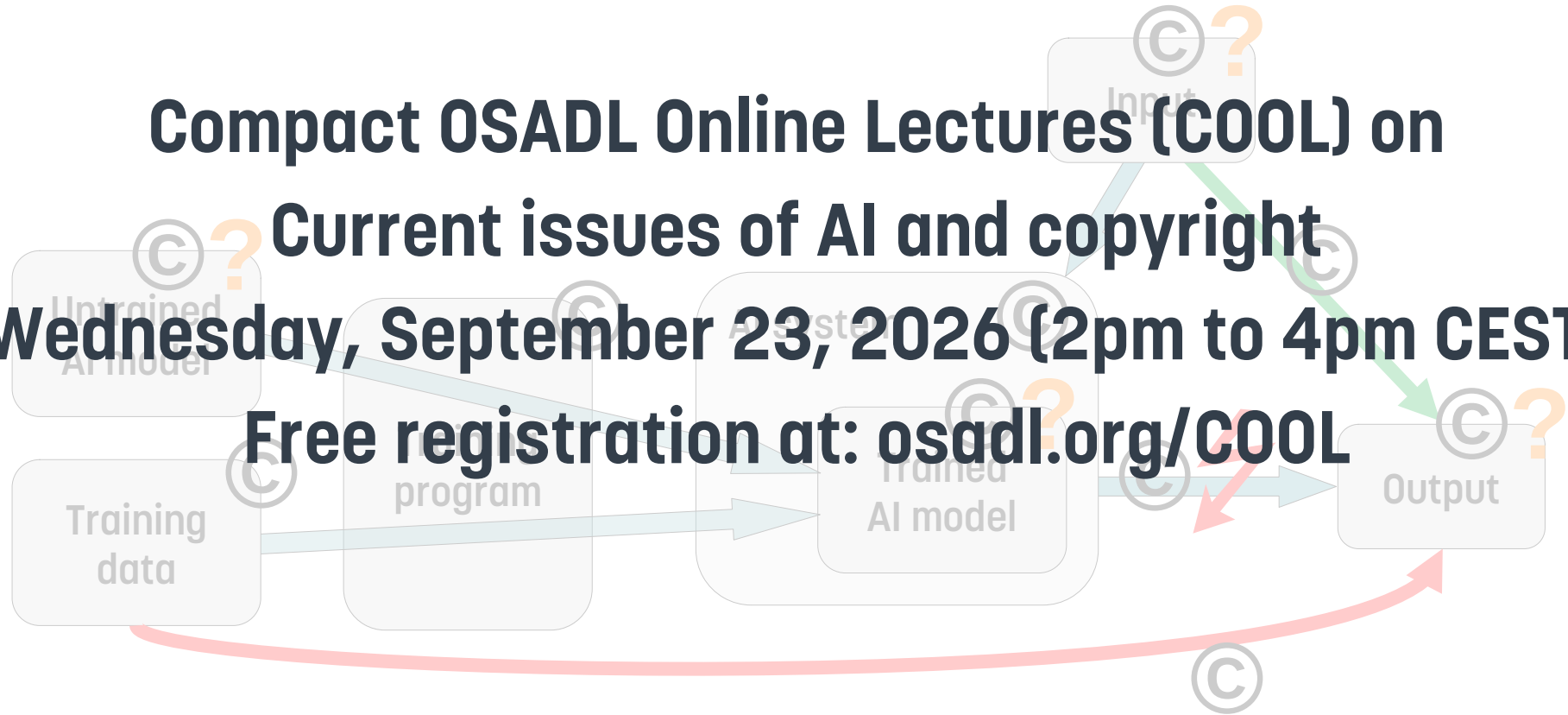
Legal Aspects Open Source Players Should Know About
HeiChips Summer School, 3 August 2026
Caren Kresse, OSADL eG

What about AI and copyright?



What about AI and copyright?

**Compact OSADL Online Lectures (COOL) on
Current issues of AI and copyright
Wednesday, September 23, 2026 (2pm to 4pm CEST)
Free registration at: osadl.org/COOL**



Open Source and the Cyber Resilience Act

In (very) short:

- Open Source software/hardware falls under the CRA if it is made available on the EU market in the course of a **commercial activity**.

Open Source and the Cyber Resilience Act

In (very) short:

- Open Source software/hardware falls under the CRA if it is made available on the EU market in the course of a **commercial activity**.

“Commercial activity” is to be understood as providing goods, in return for payment or free of charge, in a business-related context, e.g.

- charging a price for technical support services
- intention to monetize
- processing of personal data (except for certain improvements)
- accepting donations exceeding costs.

Open Source and the Cyber Resilience Act

In (very) short:

- Open Source software/hardware falls under the CRA if it is made available on the EU market in the course of a **commercial activity**.
- CRA obligations when **integrating FOSS components** into a commercial product:
 - Due diligence & Documented risk assessment
 - Machine-readable SBOM
 - No exploitable CVEs at release & Vulnerability management during support period (e.g. security updates)
 - Reporting actively exploited vulnerabilities

Publishing FOSS: Case 1

Internally used tooling is published as a project under a commercial company's name on a public source code repository.

- Commercially driven software development is not covered by the CRA, only **commercial distribution**.
- Merely mentioning the manufacturer is not a commercial activity.
- As the offer is not commercial, **case 1 does not fall under the CRA**.
- Might be assessed differently if the company **charges for use** of the tool internally **within a corporation**.

Publishing FOSS: Case 2

Bug fixes or functionalities required for a company's own products are contributed to an external FOSS project.

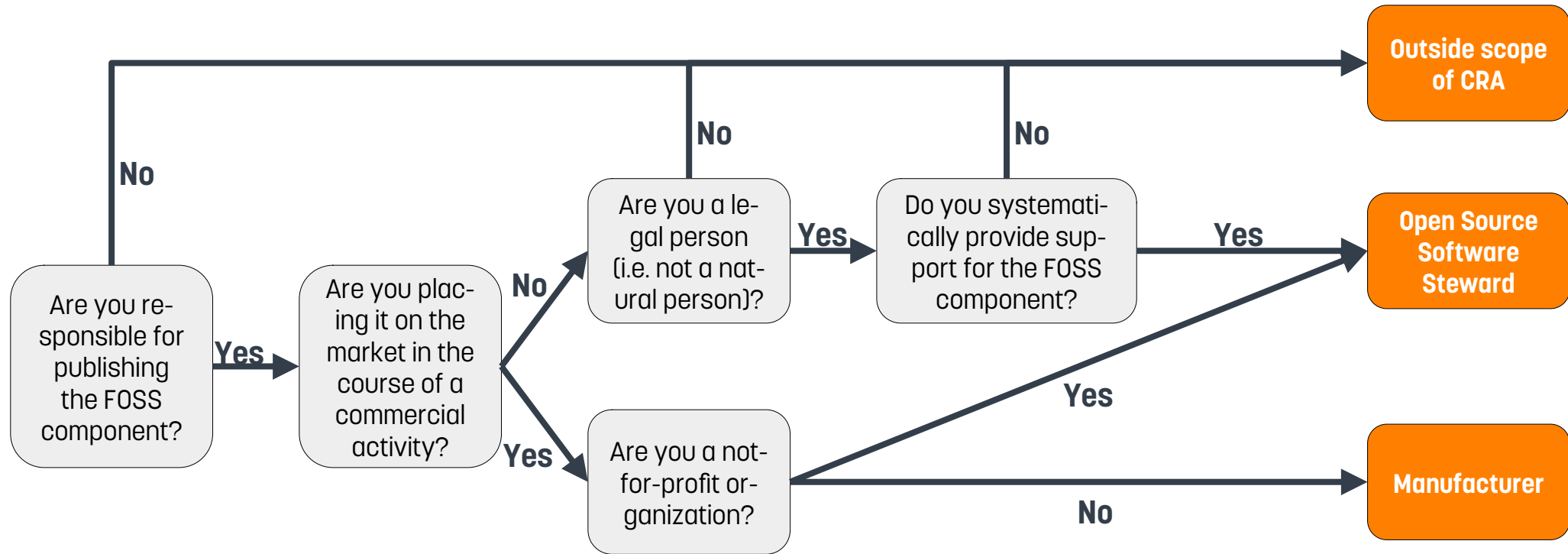
- A contribution to a FOSS project does not lead to responsibility under the CRA → to not inhibit Open Source innovation.
- In the context of making the product with the named functionalities available on the market, the CRA must be considered.

Publishing FOSS: Case 3

Software that can be used with the company's own products is published as a project on a public source code repository.

- Not a contribution to an external project but an entire component that is intended for commercial use.
- Company might be **Open Source Stewart** or – if the component is otherwise monetized – manufacturer.

Manufacturer, Open Source Stewart or out of scope?



Adapted from Comission Guidance on CRA, chapter 3

Conclusion

- Open Source is **Open Innovation** for software/ hardware/ data/ other works.
- Open Source is protected by **copyright law**.
- **Distributing or publishing** Open Source **requires compliance** with copyright (licenses, employment situations) and other regulations (if in commercial context: CRA).

Questions?

Get in touch at **C.Kresse@osadl.org**

Or via **LinkedIn (Caren Kresse)**

**OPEN SOURCE +
INDUSTRY = OSADL**

