# Open Source Software

**Prof. Dr. Dirk Riehle**

**Friedrich-Alexander University Erlangen-Nürnberg**

## COSS C01

# Agenda

1. Legal definition (open source software)
2. A (very) short history
3. Traditional open source
4. Challenges to traditional open source
5. Open source foundations
6. Open source control mechanisms
7. Community vs. commercial open source

# 1. What is Open Source Software?

# Free and Open Source Software Definition

- Software is **free software** [1] if
    - The user is granted rights to
        - Use, study, modify, and distribute the software
        - Free of charge and other restrictions

- Managed by the Free Software Foundation



- Software is **open source software** [2] if
    - The user granted rights to
        - Use, modify, and distribute the software
        - Free of charge and other restrictions

- Managed by the Open Source Initiative
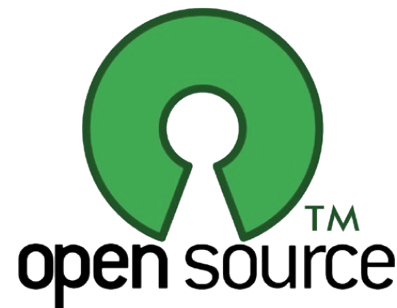


- For all practical purposes, free and open source software are the same

[1]  See https://www.gnu.org/philosophy/free-sw.html.en
[2]  See https://opensource.org/osd

# Anatomy of Open Source Licenses

1. **Copyright notice**
   - The name of the owner and when this work was created and updated

2. **Rights grant**
   - The rights granted to a user if they fulfill obligations matching the use-case

3. **Obligations to fulfill**
   - A set of obligations (requirements) before the rights grant becomes valid

4. **Prohibitions (none in the MIT license)**
   - A set of things the user is prohibited from

5. **Disclaimer**
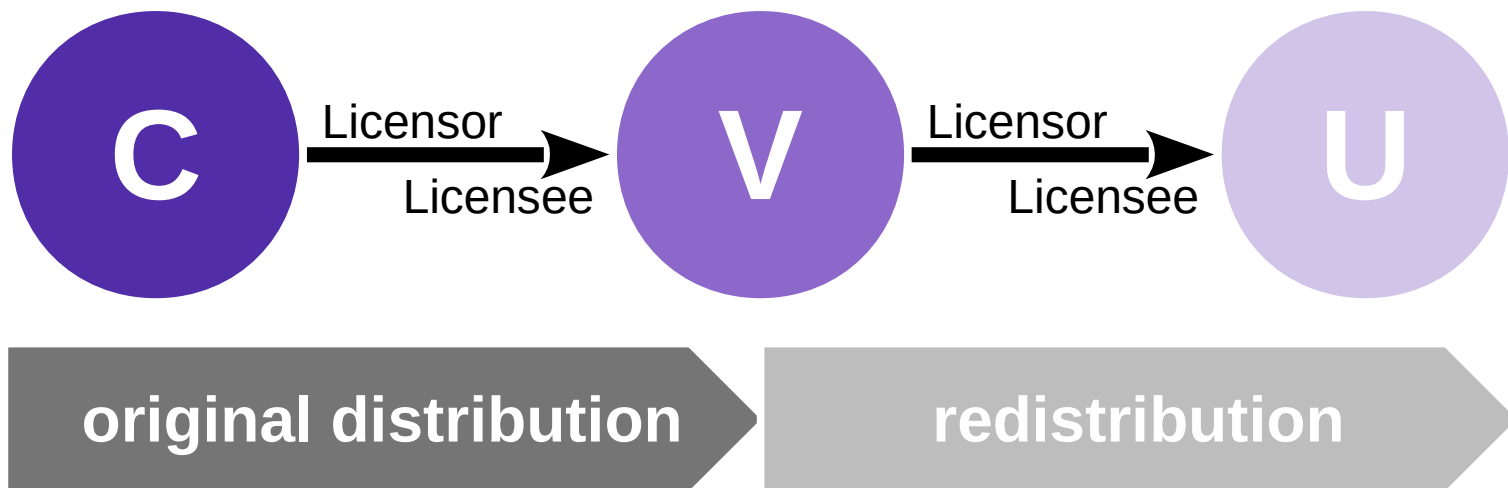   - The usual disclaimer of warranties, guarantees, etc.

# The MIT License (Template)

**1** Copyright <YEAR> <COPYRIGHT HOLDER>

**2** 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:

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

**5** THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# The Most Common Obligations

- Legal notices
  - Provide attribution
  - Provide license text
  - Provide disclaimers
  - Provide additional notices

- Copyleft

# Distribution and Rights Propagation
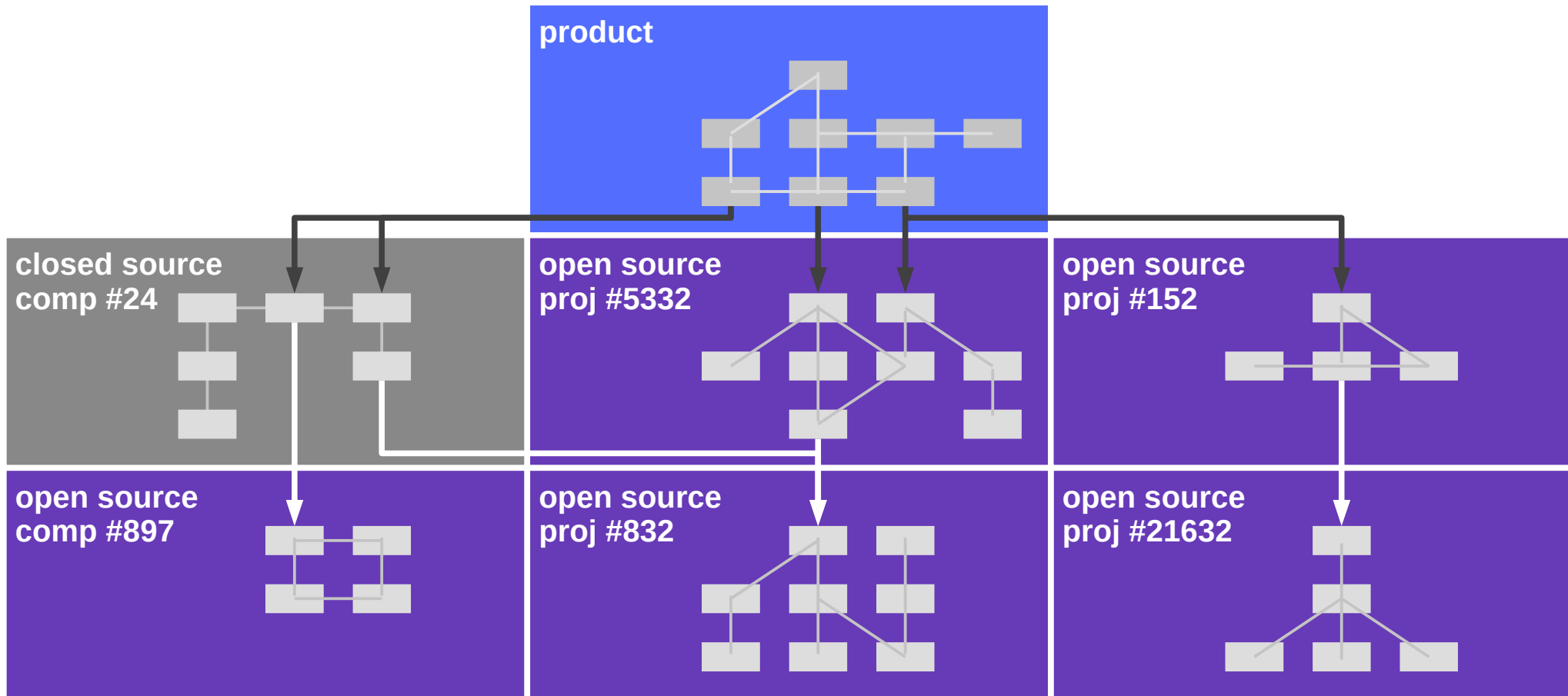
# Example Open Source Software

# The Open Source Software Supply Chain



**product**

**closed source comp #24**

**open source proj #5332**

**open source proj #152**

**open source comp #897**

**open source proj #832**

**open source proj #21632**
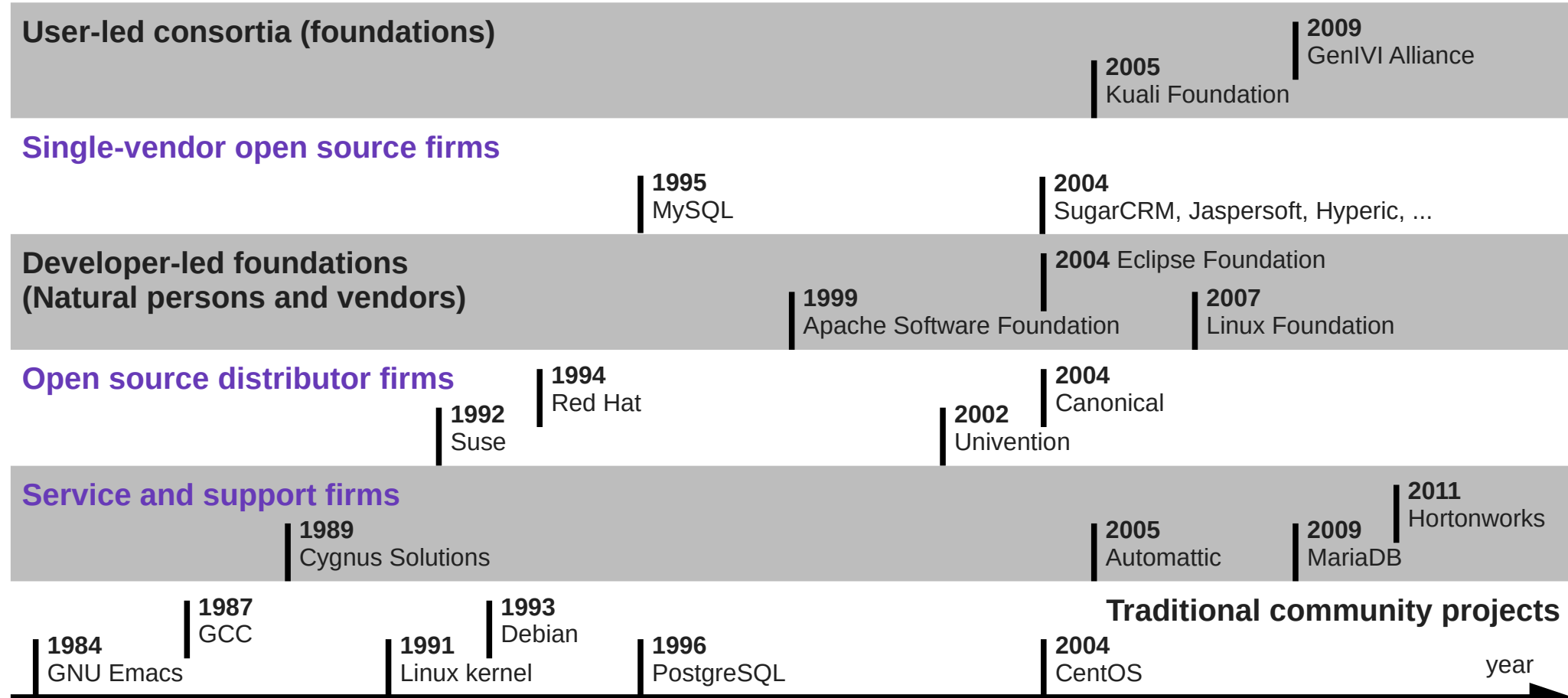
# 2. A (Very) Short History

# Short History of Open Source

- 1960-1979: Not-born-yet (the first era) [LT02]
  - Little or no recognition of software as intellectual property
  - Free sharing of source code, allowing for rapid diffusion and innovation

- 1980-1989: Philosophy (the second era)
  - Founding of the Free Software Foundation by Richard Stallman in 1985
  - Invention of GNU public license for "freeing software"

- 1990-1999: Pragmatism (the third era)
  - Founding of Open Source Initiative in 1998, increased pragmatism
  - Start of growth in number of projects as well as open source licenses

- 2000-2009: Professionalization (the fourth era)
  - Professionalization of open source, away from pure volunteerism
  - Increased focus on commercialization

- 2010-today: Mainstream (the current era)
  - Continued strong growth, simplified access, improved tooling
  - Open source as an on-ramp to the cloud

# Sustainable Open Source Projects

- Non-profit open source organizations
  - Confederated community projects
  - Open source **vendor-led foundations**
  - Open source **user-led foundations**

- For-profit open source firms
  - **Single-vendor** open source **firms**
  - Open source **distributor firms**
  - **Service and support firms**

# Open Source Project Strata and History

**User-led consortia (foundations)**

**2009**
GenIVI Alliance

**2005**
Kuali Foundation

**Single-vendor open source firms**

**1995**
MySQL

**2004**
SugarCRM, Jaspersoft, Hyperic, ...

**Developer-led foundations
(Natural persons and vendors)**

**2004** Eclipse Foundation

**1999**
Apache Software Foundation

**2007**
Linux Foundation

**Open source distributor firms**

**1994**
Red Hat

**1992**
Suse

**2002**
Univention

**2004**
Canonical

**Service and support firms**

**2011**
Hortonworks

**1989**
Cygnus Solutions

**2005**
Automattic

**2009**
MariaDB

**1987**
GCC

**1993**
Debian

**Traditional community projects**

**1984**
GNU Emacs

**1991**
Linux kernel

**1996**
PostgreSQL

**2004**
CentOS

year

Not a complete history: Events have been chosen for illustration purposes

# 3. Traditional Open Source Software

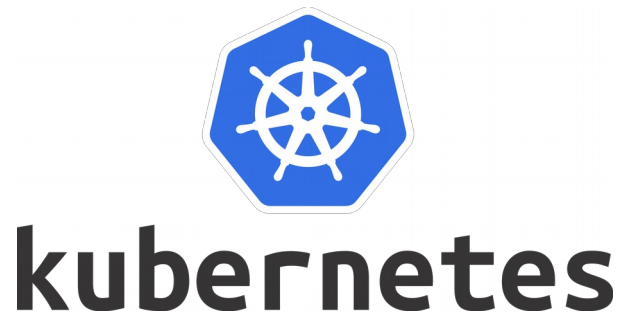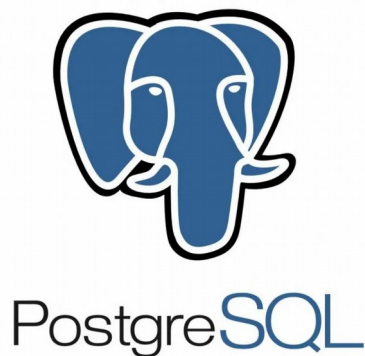# Aspects of an Open Source Project [1]

- Name
- License
- Place
- Rights

[1]   Bad naming: Neither open source software nror their communities are projects

# The Name of an Open Source Project

- Uniquely identifies project
- Ideally supported by place / domain
- Should be trademarked
- Often visually supported

# The License of an Open Source Project

## 2009

| # | Name | Market Share |
|---|------|--------------|
| 1 | GNU General Public License (GPL) 2.0 | 52.20% |
| 2 | GNU Lesser General Public License (LGPL) 2.1 | 9.84% |
| 3 | Artistic License (Perl) | 9.01% |
| 4 | BSD License 2.0 | 6.27% |
| 5 | GNU General Public License (GPL) 3.0 | 4.15% |
| 6 | Code Project Open 1.02 License | 3.59% |
| 7 | Apache License 2.0 | 3.58% |
| 8 | MIT License | 3.32% |
| 9 | Mozilla Public License (MPL) 1.1 | 1.25% |
| 10 | Common Public License (CPL) | 0.64% |
| 11 | zlib/libpng License | 0.51% |
| 12 | Academic Free License | 0.43% |
| 13 | Eclipse Public License (EPL) | 0.40% |
| 14 | Open Software License (OSL) | 0.37% |
| 15 | GNU Lesser General Public License (LGPL) 3.0 | 0.37% |
| 16 | Mozilla Public License (MPL) 1.0 | 0.30% |
| 17 | PHP License Version 3.0 | 0.28% |
| 18 | Ruby License | 0.26% |
| 19 | Sun Berkeley License (BSD 2+) | 0.18% |
| 20 | Common Development and Distribution License | 0.16% |

## 2016

| # | Name | Market Share |
|---|------|--------------|
| 1 | GNU General Public License (GPL) 2.0 | 25% |
| 2 | MIT License | 19% |
| 3 | Apache License 2.0 | 16% |
| 4 | GNU General Public License (GPL) 3.0 | 10% |
| 5 | BSD License 2.0 (3-clause, New or Revised) | 7% |
| 6 | Artistic License (Perl) | 5% |
| 7 | GNU Lesser General Public License (LGPL) 2.1 | 5% |
| 8 | GNU Lesser General Public License (LGPL) 3.0 | 2% |
| 9 | Microsoft Public License | 2% |
| 10 | Eclipse Public License (EPL) | 2% |
| 11 | Code Project Open License 1.02 | 1% |
| 12 | Mozilla Public License (MPL) 1.1 | < 1% |
| 13 | Simplified BSD LIcense (BSD) | < 1% |
| 14 | Common Development and Distribution License | < 1% |
| 15 | GNU Affero General Public License v3 or later | < 1% |
| 16 | Microsoft Reciprocal License | < 1% |
| 17 | Sun GPL with Classpath Exception v2.0 | < 1% |
| 18 | CDDL-1.1 | < 1% |
| 19 | zlib/libpng License | < 1% |
| 20 | Common Public License (CPL) | < 1% |

# The Place of an Open Source Project

- Wants to be found
- Wants to explain itself
- Needs to market itself
- Where work gets done
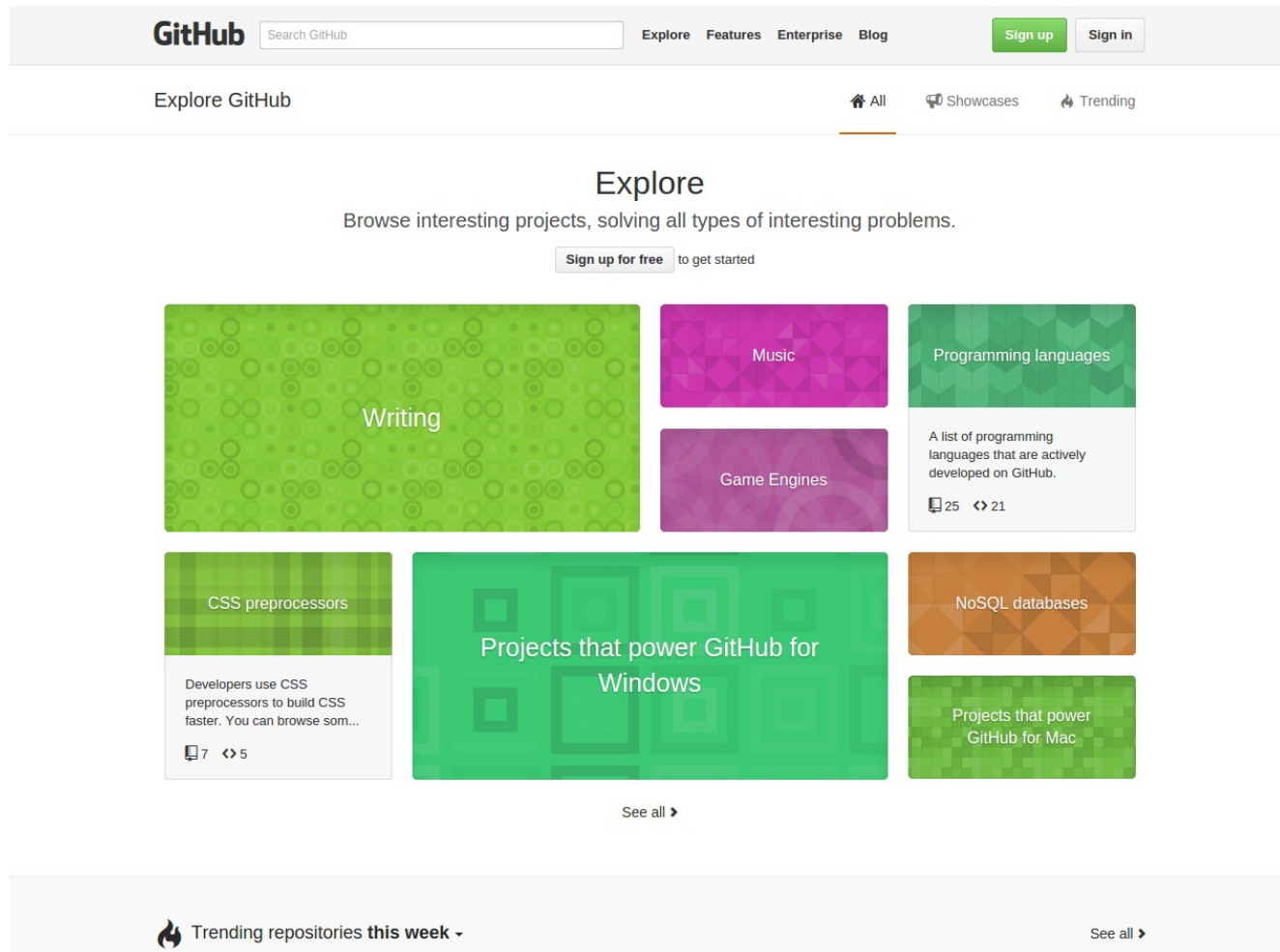
# The Rights to an Open Source Project

- Copyright
- Trademark rights
- Domain ownership
- Sysadmin access
- Commit rights

# 4. Challenges to Traditional Open Source

# Challenges to Traditional Open Source

- Challenges to using open source code
  - **Ensuring license compliance**
  - **Ensuring clean intellectual property**
  - **Managing security vulnerabilities**
  - **Managing the technical dependency**

- Challenges to contributing to project
  - Avoiding abuse of contributions
  - Maintaining your interests

- Challenges to building a business
  - **Ensuring access to trademarks**
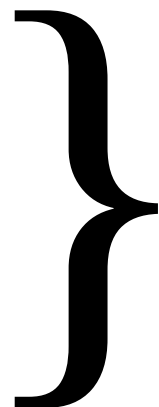
}  **Maintaining your interests**

# Ensuring License Compliance

## Legal information

Legal and safety

Regulatory information

RF information

Regulatory notices

Arbitration and opt-out

Motorola Terms and Conditions

Warranty

Open source

Google legal

System WebView licenses

Wallpapers
Satellite imagery providers:
©2014 CNES / Astrium, DigitalGlobe, Bluesky

## Third-party licenses

- /fake_packages/selinux_policy-timestamp
- /kernel
- /obj/include/qcril/qcril_features_def.h
- /recovery/root/nonplat_file_contexts
- /recovery/root/nonplat_property_contexts
- /recovery/root/plat_file_contexts
- /recovery/root/plat_property_contexts
- /recovery/root/sbin/recovery
- /recovery/root/sepolicy
- /root/init
- /root/sbin/adbd
- /system/app/CertInstaller/CertInstaller.apk
- /system/app/CompanionDeviceManager/Com
- /system/app/HTMLViewer/HTMLViewer.apk
- /system/app/LiveWallpapersPicker/LiveWallp
- /system/app/MotoPhotoEditor/MotoPhotoEd
- /system/app/PrintSpooler/PrintSpooler.apk
- /system/app/ProgramMenu/ProgramMenu.a
- /system/app/ProgramMenuSystem/Program
- /system/app/Stk/Stk.apk
- /system/app/UserDictionaryProvider/UserDic
- /system/app/ims/ims.apk
- /system/bin/am
- /system/bin/app_process32
- /system/bin/applypatch
- /system/bin/appops
- /system/bin/appwidget
- /system/bin/atrace
- /system/bin/bmgr
- /system/bin/bu
- /system/bin/bzip2

## Third-party licenses

Notices for file(s):
/system/bin/bzip2
/system/lib/libbz.a

------------------------------------------

This program, "bzip2", the associated library documentation, are copyright (C) 1996-2010 Ju rights reserved.

Redistribution and use in source and binary f modification, are permitted provided that the are met:

1. Redistributions of source code must retain notice, this list of conditions and the fo

2. The origin of this software must not be mi not claim that you wrote the original soft software in a product, an acknowledgment i documentation would be appreciated but is

3. Altered source versions must be plainly ma not be misrepresented as being the origina

4. The name of the author may not be used to products derived from this software withou permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIM WARRANTIES OF MERCHANTABILITY AND FITNESS FOR ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCU

## Third-party licenses

==========================================

The following files are from the open source project (git://w1.fi/srv/git/hostap.git)

wlantest_ctrl.h
wpa_ctrl.c
wpa_ctrl.h

These are redistributed using the BSD license

Copyright (c) 2003-2012, Jouni Malinen <j@w1. All Rights Reserved.

This software may be distributed, used, and m BSD license:

Redistribution and use in source and binary f modification, are permitted provided that the met:

1. Redistributions of source code must retain notice, this list of conditions and the fo

2. Redistributions in binary form must reprod notice, this list of conditions and the fo documentation and/or other materials provi

3. Neither the name(s) of the above-listed co names of its contributors may be used to e derived from this software without specifi

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HO "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, LIMITED TO, THE IMPLIED WARRANTIES OF MERCHAN A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EV OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIREC SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
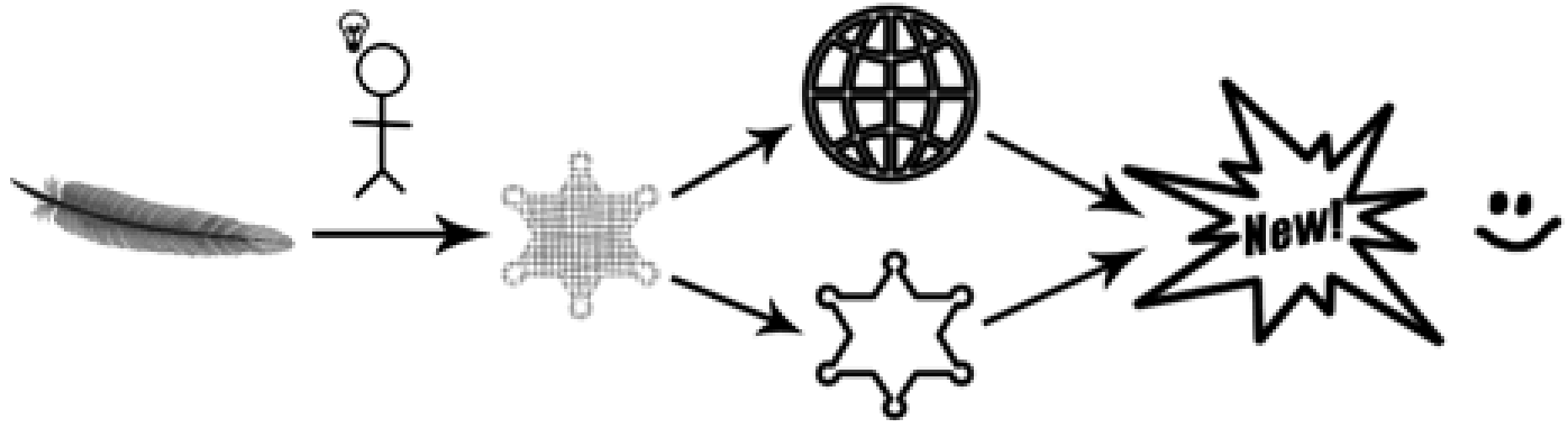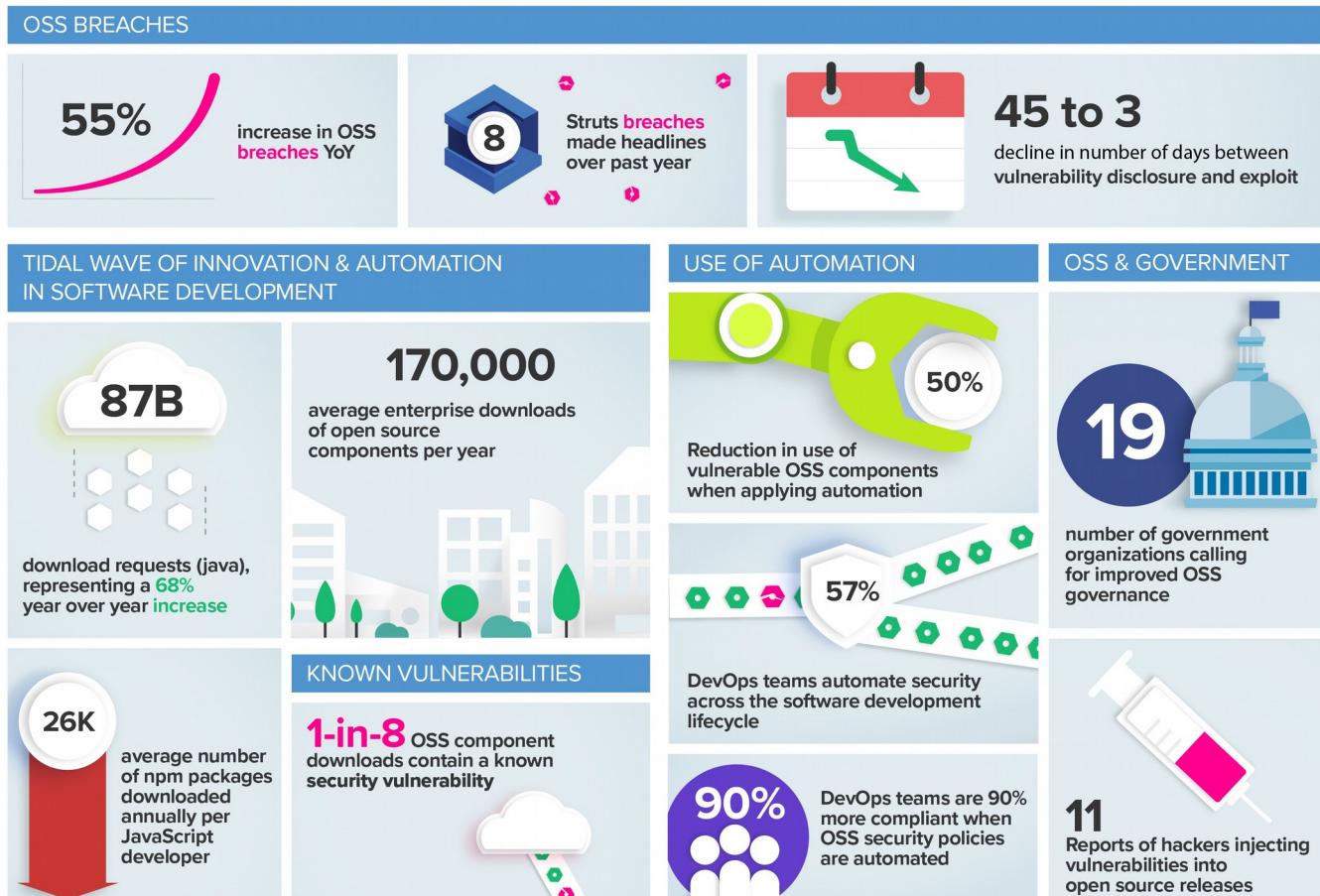
[1] See https://blog.sonatype.com/2018-state-of-the-software-supply-chain-report

# Managing the Technical Dependency (Maintaining Your Interests)
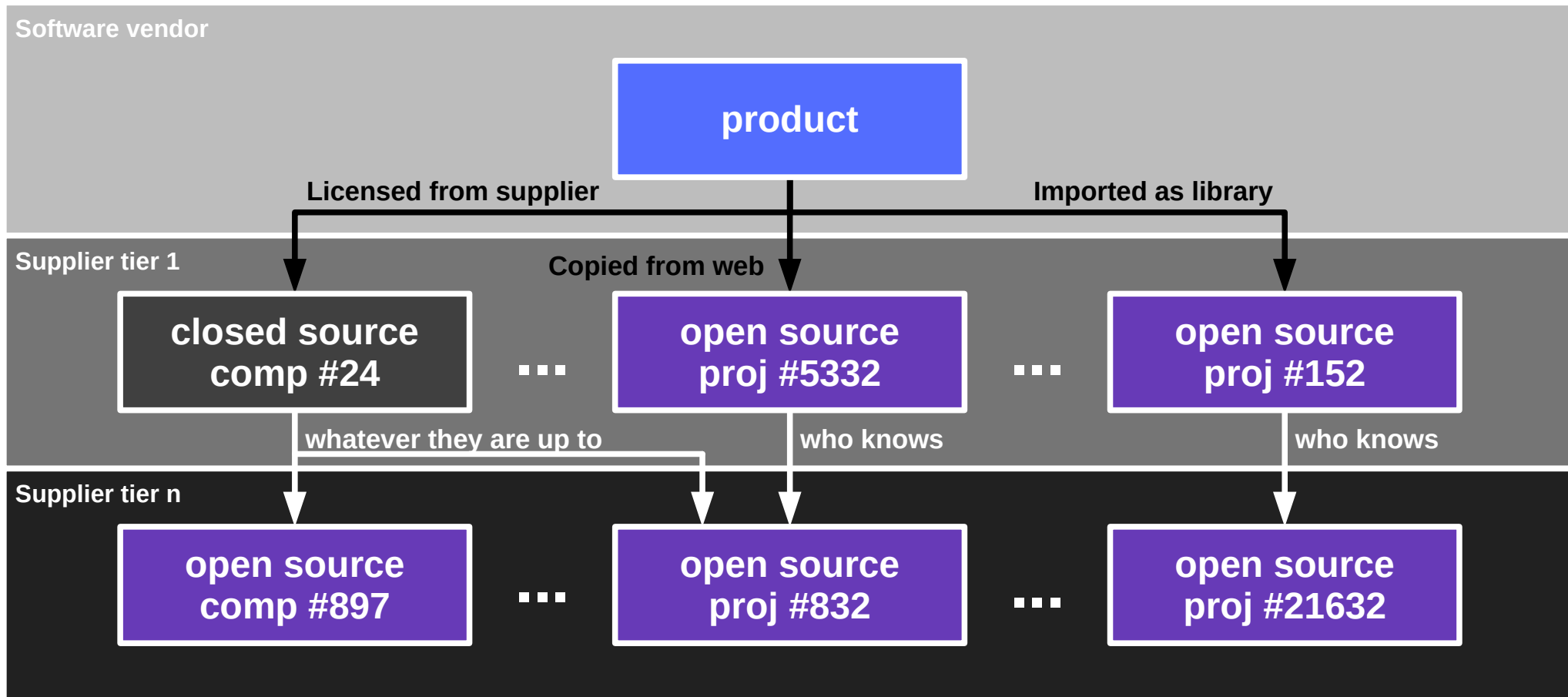
# 5. Open Source Foundations

# Developer-led Open Source Foundations [R10a]

- An **developer-led open source foundation** is
  - a **non-profit organization** (foundation, consortium)
  - with the purpose of **managing and performing the development** of
  - **non-differentiating open source software**
  - made available to foundation members and **the general public**

- Typical members of a developer-led foundation are
  - Individual developers
  - Software vendors
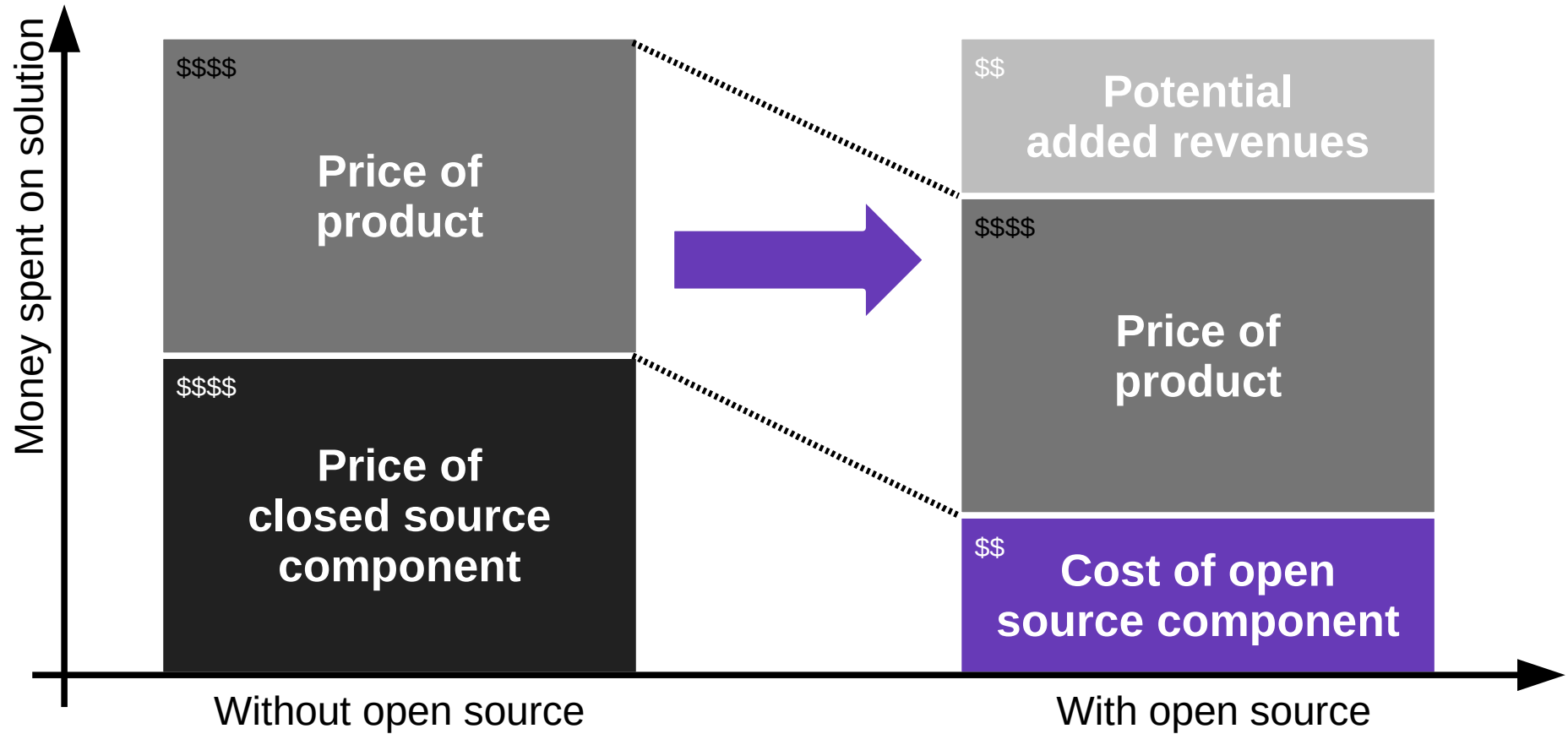  - Service providers
  - Consulting firms

# Motivation for Developer-led Foundations

1. Support and protect people and their projects
   - Protect developers from lawsuits
   - Accept donations; manage books
   - Legally represent project interests

2. Create a stronger more competitive ecosystem
   - Increase share of customer wallet
   - Reach more price-sensitive customers
   - Grow addressable market

3. Create a fair and equal playing field for members
   - Regulate access to intellectual property
   - Regulate access to marketing channels
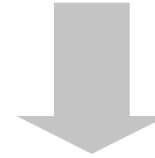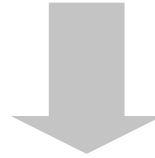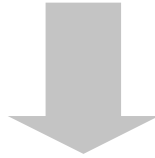   - Regulate product roadmap and development

**The Apache Software Foundation […] provides an established framework for intellectual property and financial contributions that simultaneously limits contributors potential legal exposure. Through a collaborative and meritocratic development process, Apache projects deliver […] software products […]**

# Increase Share of Customer Wallet



Money spent on solution

$$$$ **Price of product**

$$$$ **Price of closed source component**

$$ **Potential added revenues**

$$$$ **Price of product**

$$ **Cost of open source component**

Without open source

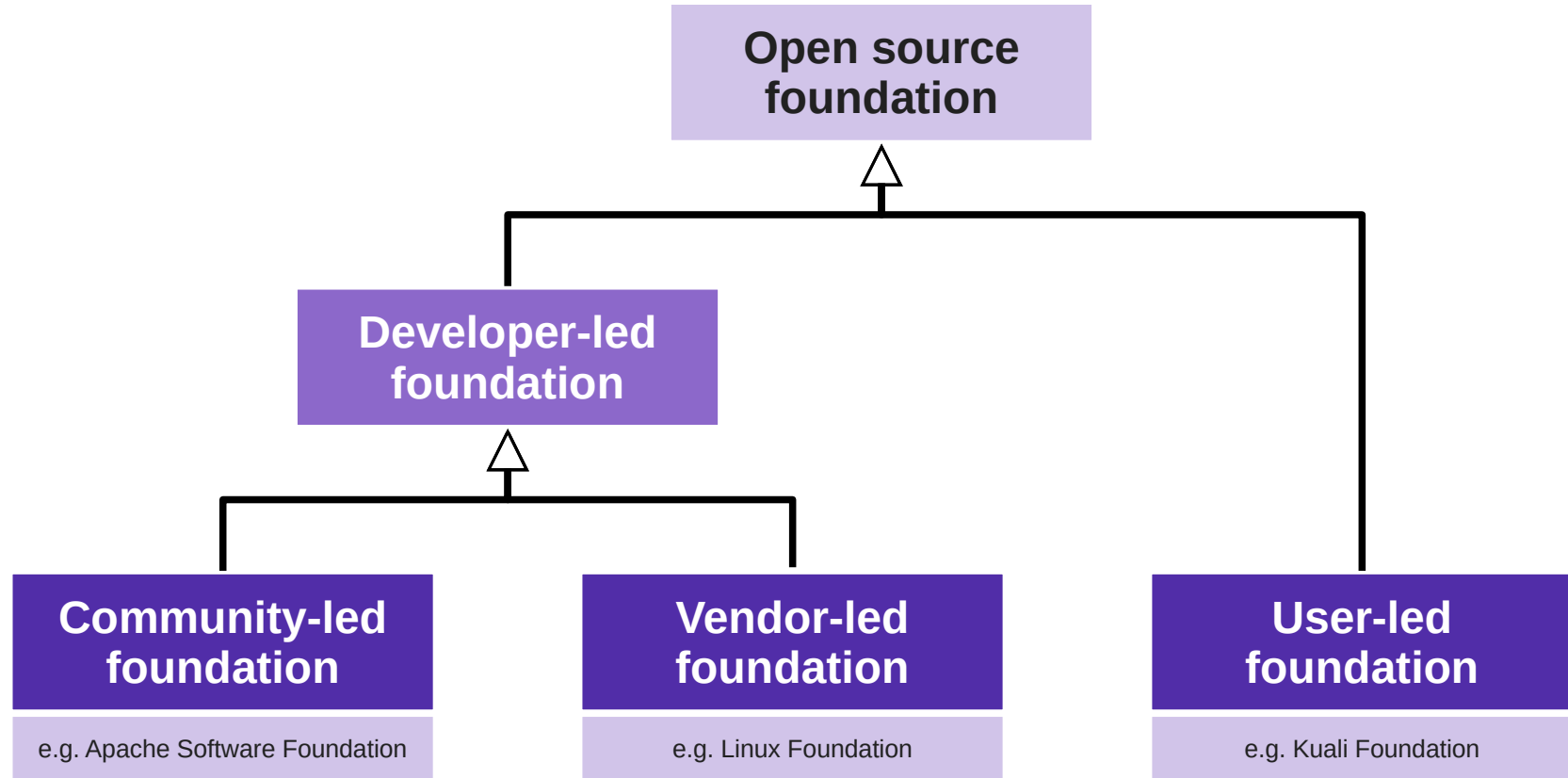With open source

The point of the foundation is to ensure free access, in perpetuity, to the software projects […] As part of this mission, the Foundation will be responsible for protecting the WordPress, WordCamp, and related trademarks. A 501(c)3 non-profit organization […]

# Types of Open Source Foundations



Open source foundation

Developer-led foundation

Community-led foundation
e.g. Apache Software Foundation

Vendor-led foundation
e.g. Linux Foundation

User-led foundation
e.g. Kuali Foundation

# 6. Open Source Control Mechanisms

# Control Points and Steering Mechanisms [R11]

1. Copyright control
2. Domain ownership
3. Trademark control
4. Social leadership
5. Committer rights

# Copyright Control and How it Can be Used

- Set-up
  - You are the only party with full copyright or relicensing right
    - Use a contributor (license) agreement to receive these rights from contributors

- Power
  - You can change the license going forward
    - Including to no open source license at all

# Domain Ownership and How it Can be Used

- Set-up
  - You own the domain and/or identifying properties
    - Related properties are social media accounts etc.

- Power
  - Your can pull access to domains and accounts
    - You force the project to fork and start a new identity

# Trademark Control and How it Can be Used

- Set-up
  - You own one or all relevant trademarks
    - In open source, these are mostly textual and visual marks (logos)

- Power
  - You can pull trademark usage right permissions
    - You force the project to fork and start a new identity

# Social Leadership and How it Can be Used

- Set-up
    - You are one or the recognized leader of a project
        - You do not have to be a developer, you just need the social recognition

- Power
    - You can split the project community, diminishing its power
    - You can keep unwanted people out of the project

# Committer Rights and How it Can be Used

- Set-up
  - You are a committer (core developer) of the project

- Power
  - You can vote to keep unwanted people out of the project
  - You can delay or reject unwanted contributions
  - You lead the technical direction of the project

# Control Points and Steering Mechanisms are Negotiation Tools

- You cannot retroactively revoke the open source license
- Your opponents can always (threaten) fork the project

# 7. Community vs. Commercial Open Source

# Definition of Commercial Open Source

- **Commercial open source software** is open source software that
  - Is being managed and developed by one or more software vendors
    - For the purposes of generating revenues through derived products

- A **commercial open source software vendor** is a software vendor that
  - Manages and develops commercial open source software

- The product itself, sold by a vendor, is never open source software

# Community vs. Commercial Open Source

| | Community Open Source | | Commercial Open Source |
|---|---|---|---|
| | **Traditional** | **Foundation** | |
| **Copyright control** | Distributed | Governed | Centralized |
| **Domain ownership** | Individual | Foundation | Centralized |
| **Trademark ownership** | Individual, if any | Foundation | Centralized |
| **Social leadership** | Distributed | Distributed | Centralized |
| **Committer rights** | Earned | Earned | Assigned |

# Summary

1. Legal definition (open source software)
2. A (very) short history
3. Traditional open source
4. Challenges to traditional open source
5. Open source foundations
6. Open source control mechanisms
7. Community vs. commercial open source

# Thank you! Questions?

**dirk.riehle@fau.de** – **http://osr.cs.fau.de**

dirk@riehle.org – http://dirkriehle.com – @dirkriehle

DR

# Credits and License

- Original version
  - © 2020 Dirk Riehle, some rights reserved
  - Licensed under Creative Commons Attribution 4.0 International License

- Contributions
  - None yet