Open Source Software

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

COSS C01

Agenda

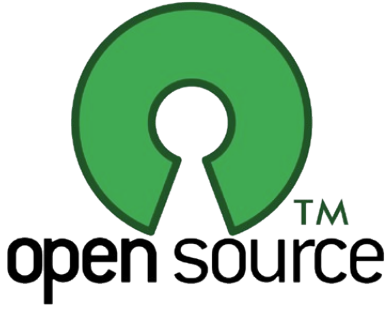
1. Legal definition (open source software)
2. A (very) short history
3. Open source licenses
4. Open source license compliance
5. Open source governance
6. Problems with using open source
7. Open source control mechanisms

1. What is Open Source Software?

Legal Definition of Free and Open Source Software

- 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 is 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

Example Open Source Software

2. A (Very) Short History

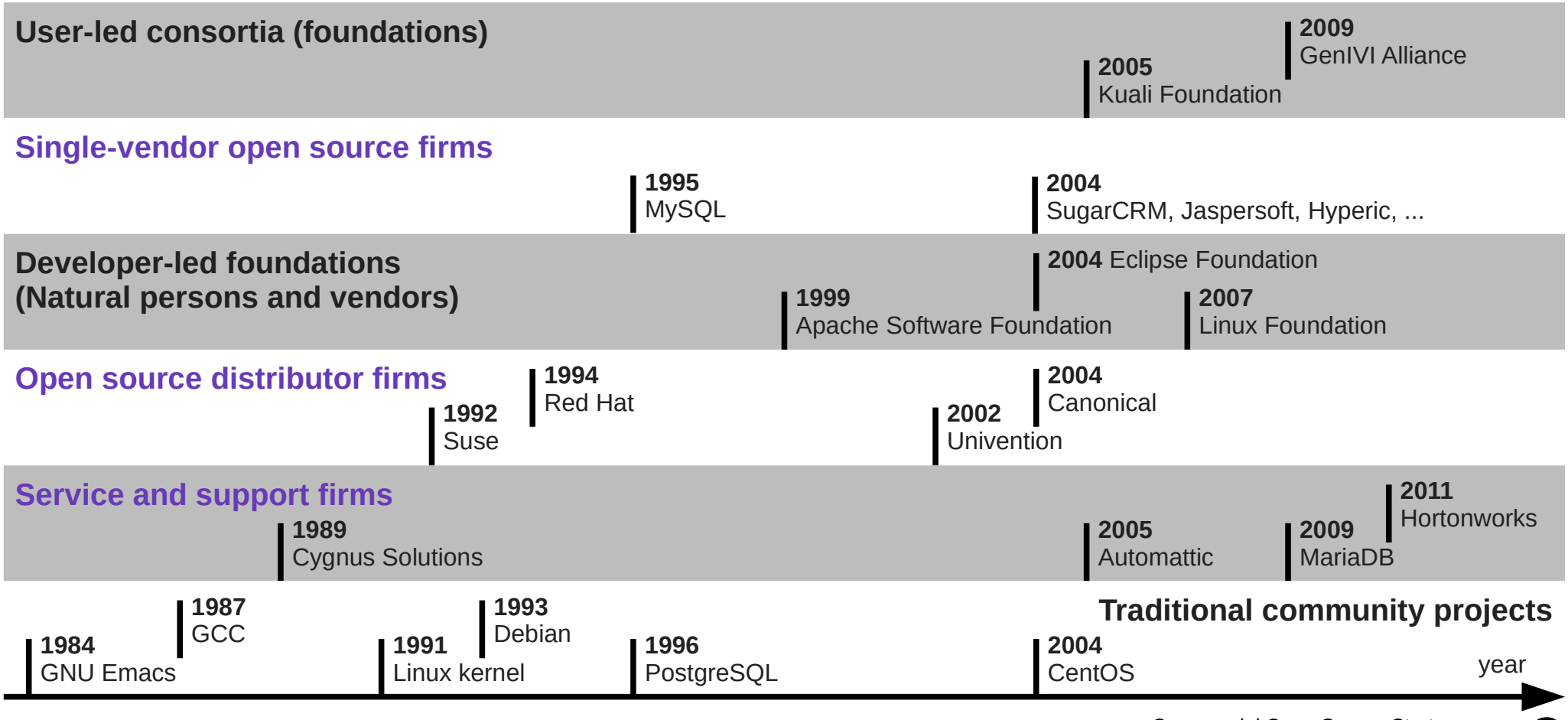
Short History of Open Source

- 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

Traditional Open Source

- A traditional open source software
  - Is software owned by a large number of contributors
    - Who all individually own the copyright to their contributions
- A traditional open source software project
  - Is an open source software + associated community that
    - Has no formal organizational backing but rather relies on individual people

Open Source Project Strata and History



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

Sustainable Open Source Projects

- Traditional community projects [1]
- Non-profit open source organizations
  - Open source **community-led foundations**
  - 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**

[1] Riehle, D. (2020). What to Call Traditional Community Open Source Projects Not Hosted by a Foundation?

3. Open Source Licenses

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.

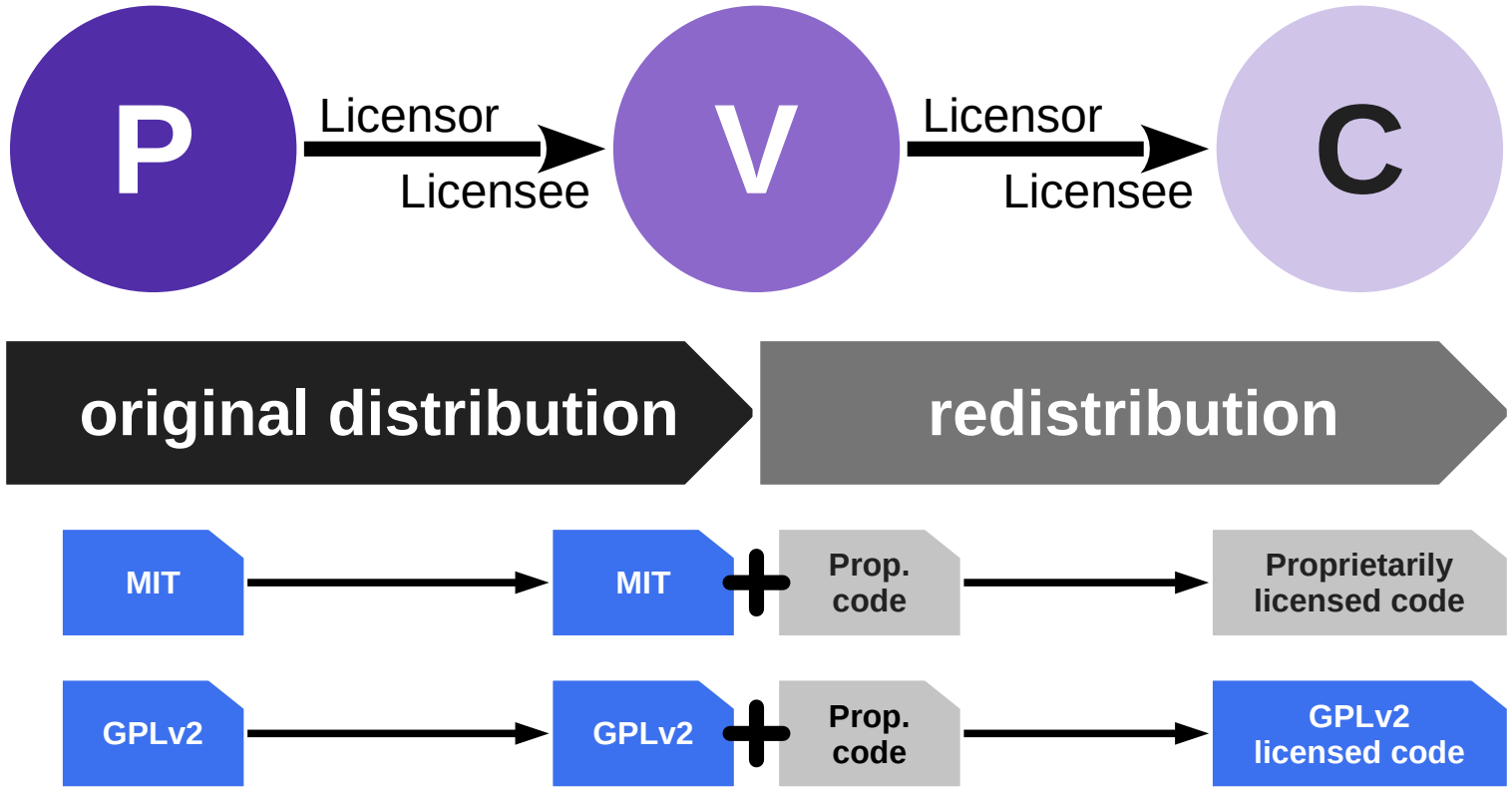
The Main Use-Cases of Open Source Software

- **In-house use** (everything where you do not pass on code)
  - Personal use
  - Demos to customers
  - Software development tools
- **Distribution** (where you pass on binary or source code)

The Most Common Obligations for the Distribution Use-Case

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

Distribution and Rights Propagation under Copyleft



P = Original open source programmer
V = Software vendor
C = Customer

Types of Licenses by Copyleft Obligation

- Permissive licenses
  - Do not include a copyleft obligation
  - Examples: MIT, BSD-2-Clause, …
- Weak copyleft licenses
  - Limited use of copyleft obligation
  - Examples: EPL-1.0, LGPL-2.1-or-later, …
- Strong copyleft licenses
  - Attempted maximum applicability of copyleft obligation
  - Examples: GPL-2.0-only, AGPL-3.0-or-later, ...

Changes in License Popularity

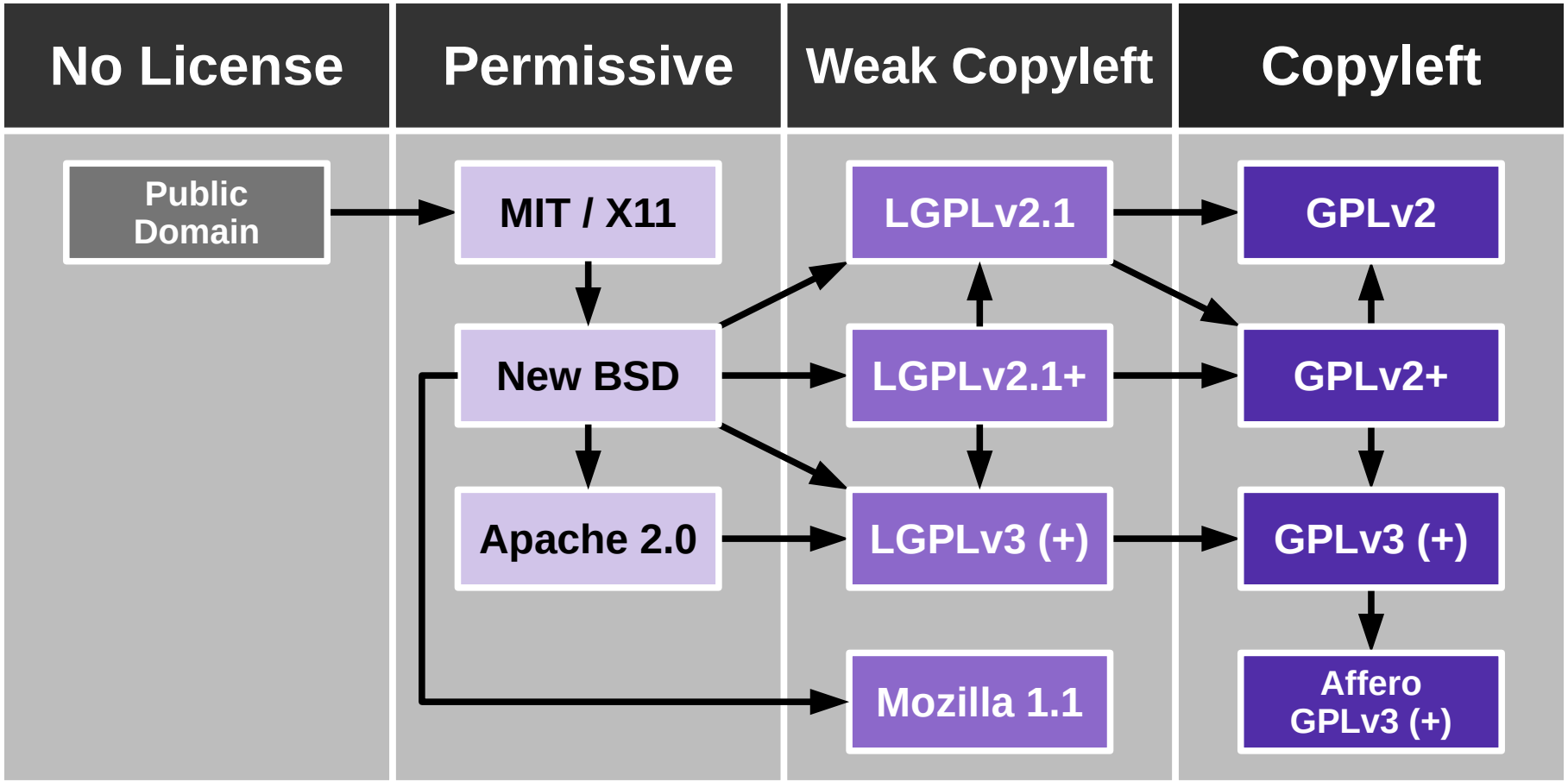
2009

#	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%

2019

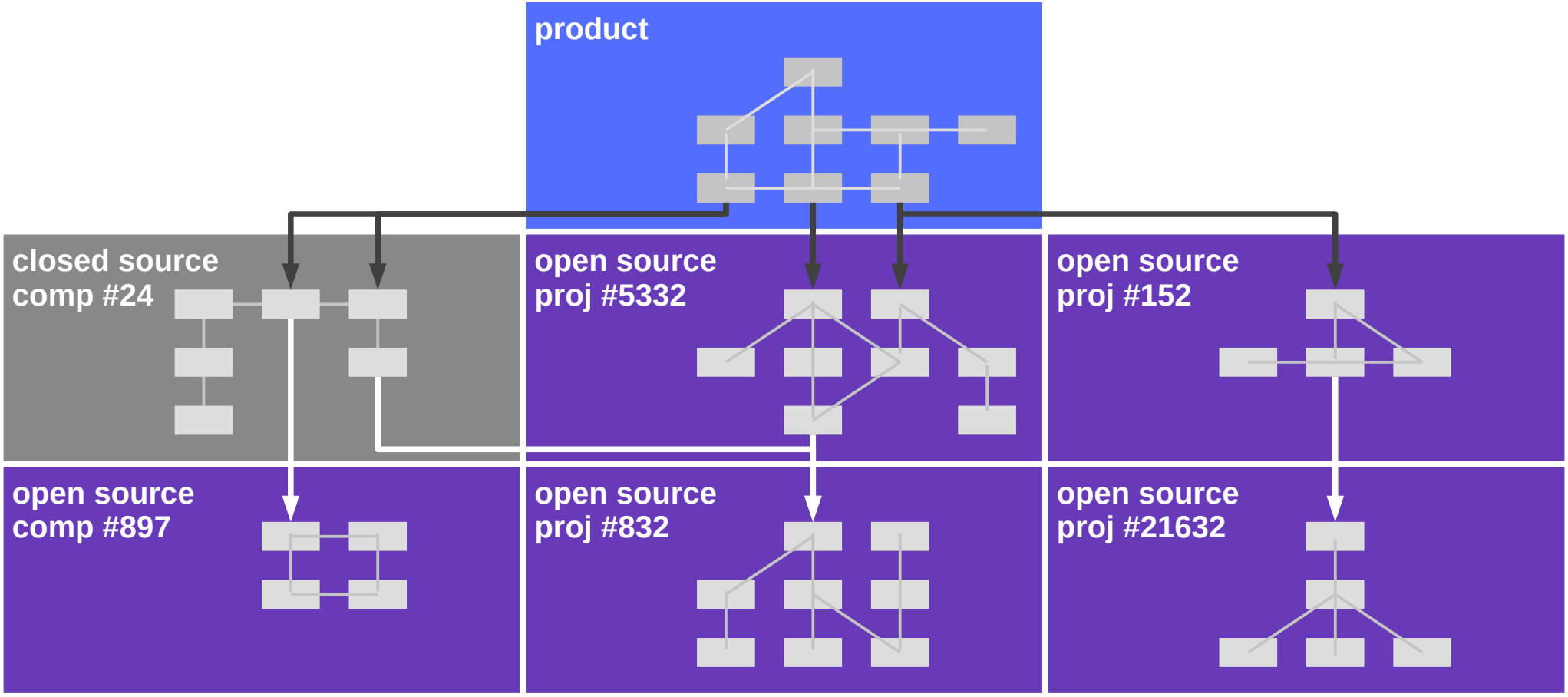
#	Name	Market Share
1	MIT License	32%
2	GNU General Public License (GPL) 2.0	18%
3	Apache License 2.0	14%
4	GNU General Public License (GPL) 3.0	7%
5	BSD License 2.0 (3-clause, New or Revised)	6%
6	ISC License	5%
7	Artistic License (Perl)	4%
8	GNU Lesser General Public License (LGPL) 2.1	4%
9	GNU Lesser General Public License (LGPL) 3.0	2%
10	Eclipse Public License (EPL)	1%
11	Microsoft Public License	1%
12	Simplified BSD License (BSD)	1%
13	Code Project Open License 1.02	1%
14	Mozilla Public License (MPL) 1.1	<1%
15	GNU Affero General Public License 3.0 or later	<1%
16	Common Development and Distribution License	<1%
17	Do What the F**k You Want To Public License	<1%
18	Microsoft Reciprocal License	<1%
19	Sun GPL with Classpath Exception 2.0	<1%
20	zlib/libpng License	<1%

Open Source License Categories and Families

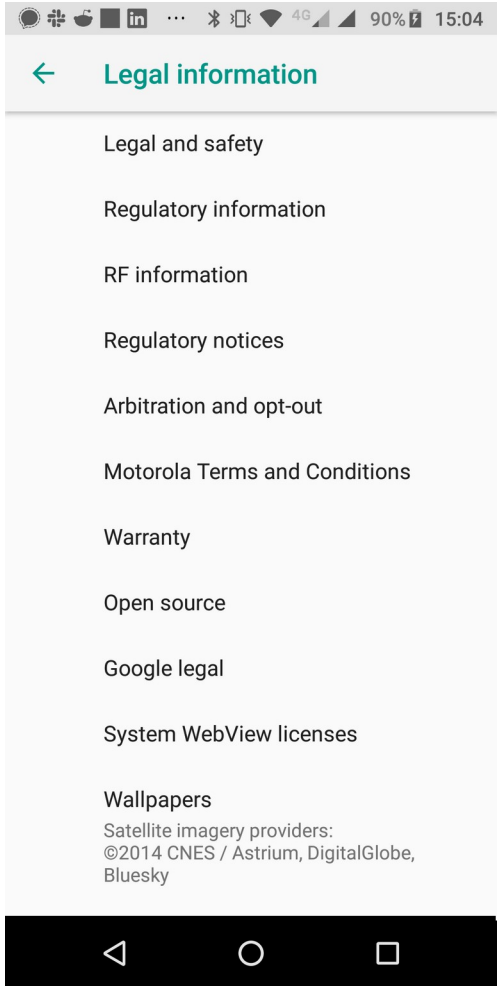


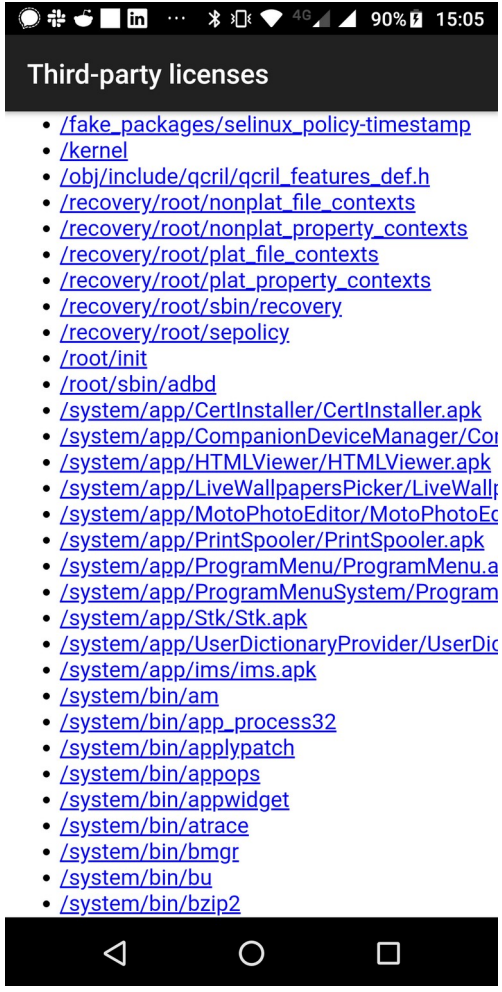
4. Open Source License Compliance

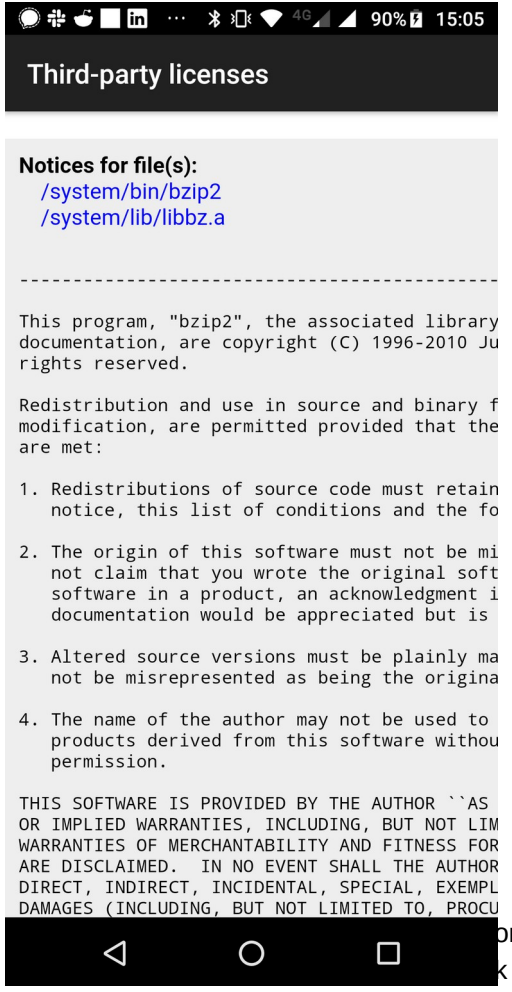
The Software Supply Chain

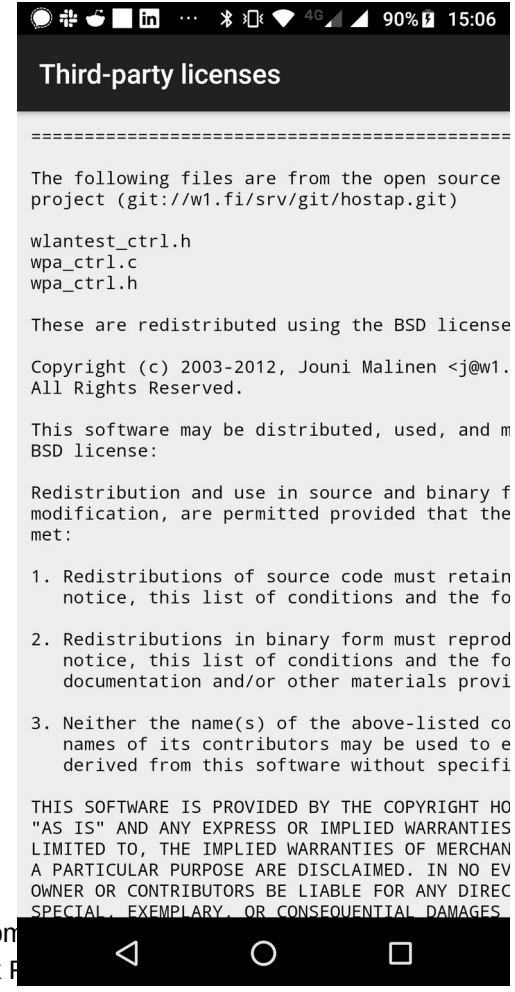


Android’s Legal Notices (Distribution Use-Case)









License Incompliance Discovery Risk

Consumer >> Enterprise

Low price >> High price

Embedded >> Cloud computing

Copyleft license >> Permissive license

5. Open Source Governance

Open Source Governance

- Governance
  - Is the set of processes, practices, institutions, and roles used to lead and manage a social system
- **Open source governance** in companies
  - Is the governance of using open source software in your products
    - Initial selection of components
    - Management of dependency
    - Eventual replacement
  - Usually the prerogative of an **open source program office**
- Example governance for universities
  - University of California
    - https://security.ucop.edu/resources/open-source-software-licensing.html
  - My research group
    - https://goo.gl/2fm4cx

Open Source Don’ts (Example Governance Rules 1 / 3)

- Do not copy source code with unsure license into your project codebase
  - Random code on the web without a license is proprietary code
- Do not copy source code that is copyleft-licensed (from wherever) into your codebase
  - Do not copy from Stack Overflow (code is copyleft-licensed)
  - Do not copy and paste from open source projects
- Do not include copyleft-licensed libraries or other components into your project
- Do not blindly trust the license that an open source component is labeled with
- Do not combine software components with contradicting licenses

Open Source Dos (Example Governance Rules 2 / 3)

- Only use permissively licensed open source components
- Prefer governed sources over ungoverned ones like Github
- Maintain a software bill-of-materials for the creation of legal notices

Projects, Licenses, and Sources

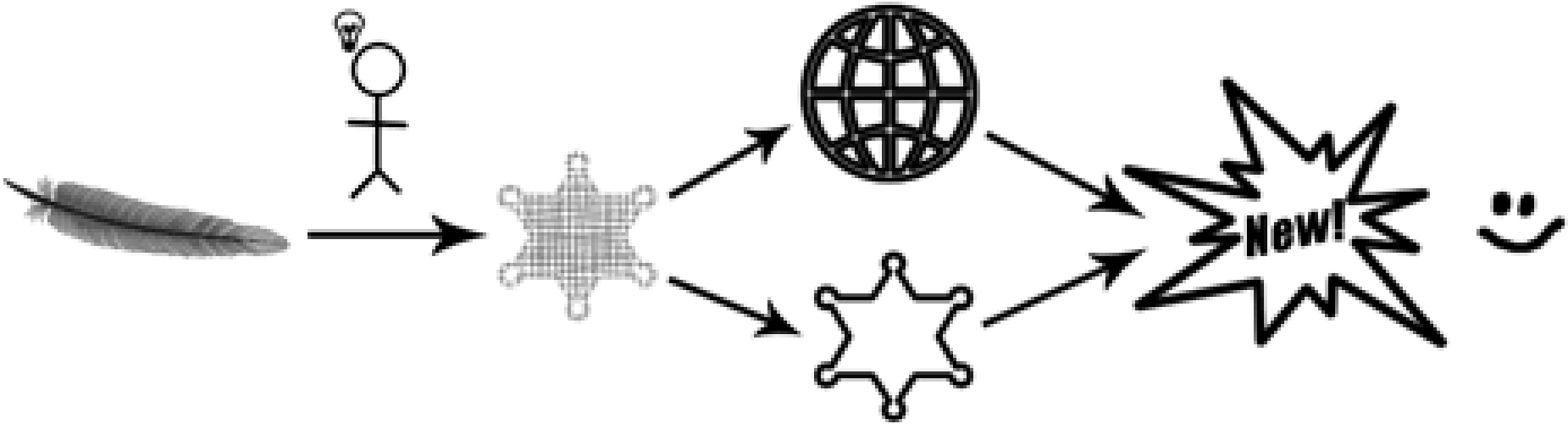
	Allowed	Must-ask	Denied
Projects	• PostgreSQL		
Licenses	• MIT • Apache 2.0 • All BSD variants	• EPL 1.1, EPL 2.0	• Any GPL license
Sources	• ASF website • Google Github repo • FB Github repo	• Linux Foundation • Eclipse Foundation	• Stack Overflow • Random website

6. Problems with Using Open Source

Problems with Using Open Source Software

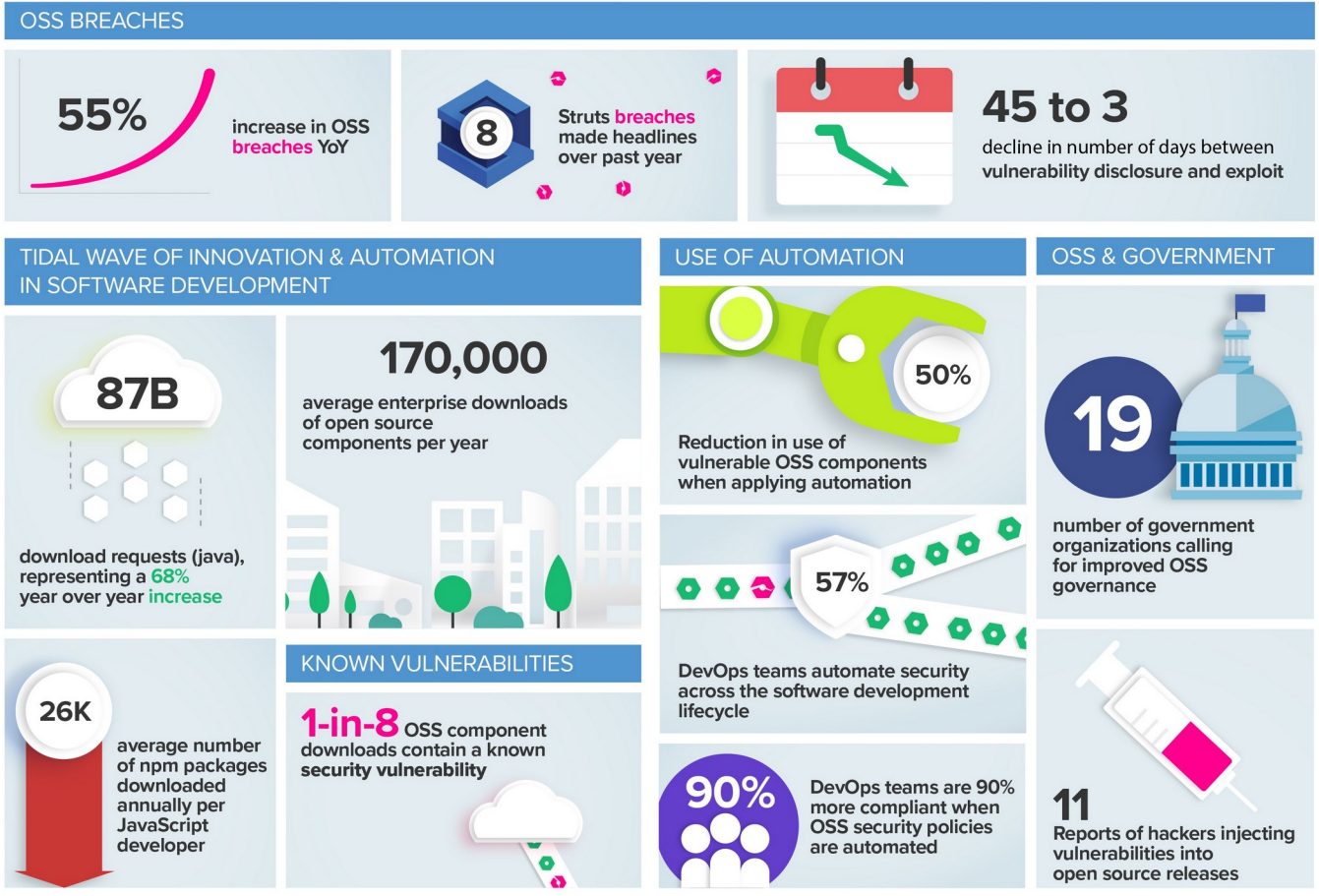
- When using open source software
  - Ensuring clean intellectual property
  - Managing security vulnerabilities
  - Managing the technical dependency
- When building a business on top
  - Ensuring access to source code
  - Ensuring access to trademarks
  - Ensuring access to patents

Ensuring Clean Intellectual Property [1]



[1] See https://www.eclipse.org/projects/dev_process/ip-process-in-cartoons.php

Managing Security Vulnerabilities [1]



[1] See https://blog.sonatype.com/2018-state-of-the-software-supply-chain-report

Ensuring Access to Intellectual Property

7. Open Source Control Mechanisms

Control Points and Steering Mechanisms [R11]

1. Intellectual property control
    1. Copyright control
    2. Patent ownership
    3. Trademark control
    4. Media ownership
2. Position of social leadership
    1. Leadership position
    2. Committer rights

Control Using Intellectual Property Rights

- Through copyright ownership
  - **Changing the license going forward**
- Through trademark ownership
  - **Withdrawing usage trademark right**
- Through patent ownership
  - **Charging patent license fees**
- Through media ownership
  - **Use of media to your advantage**

Steering Using Social Leadership

- Through social leadership position
  - **Splitting the project community, diminishing its power**
  - **Keeping unwanted people out of the project**
- Through committer rights
  - **Delaying or rejecting unwanted contributions**
  - **Leading the technical direction of the project**

Summary

1. Legal definition (open source software)
2. A (very) short history
3. Open source licenses
4. Open source license compliance
5. Open source governance
6. Problems with using open source
7. Open source control mechanisms

Thank you! Questions?

dirk.riehle@fau.de – https://oss.cs.fau.de

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

Credits and License

- Original version
  - © 2021 Dirk Riehle, some rights reserved
  - Licensed under Creative Commons Attribution 4.0 International License
- Contributions
  - None yet