OPEN SOURCE CULTURE, STANDARDS, RISKS, AND REMEDIATION: A DEEP DIVE

Jeff Luszcz
VP Product Management
jluszcz@flexera.com

@jeffluszcz

Disclaimer

IANAL; // I am not a lawyer;
IANYL; // I am not _your_ lawyer;

IANYP; // I am not _your_ programmer;

The purpose of today's talk is to provide an introduction to the Open Source Compliance

Only your legal counsel can tell you what you need to do

OPEN SOURCE GOVERNANCE BASICS

Jeff Luszcz

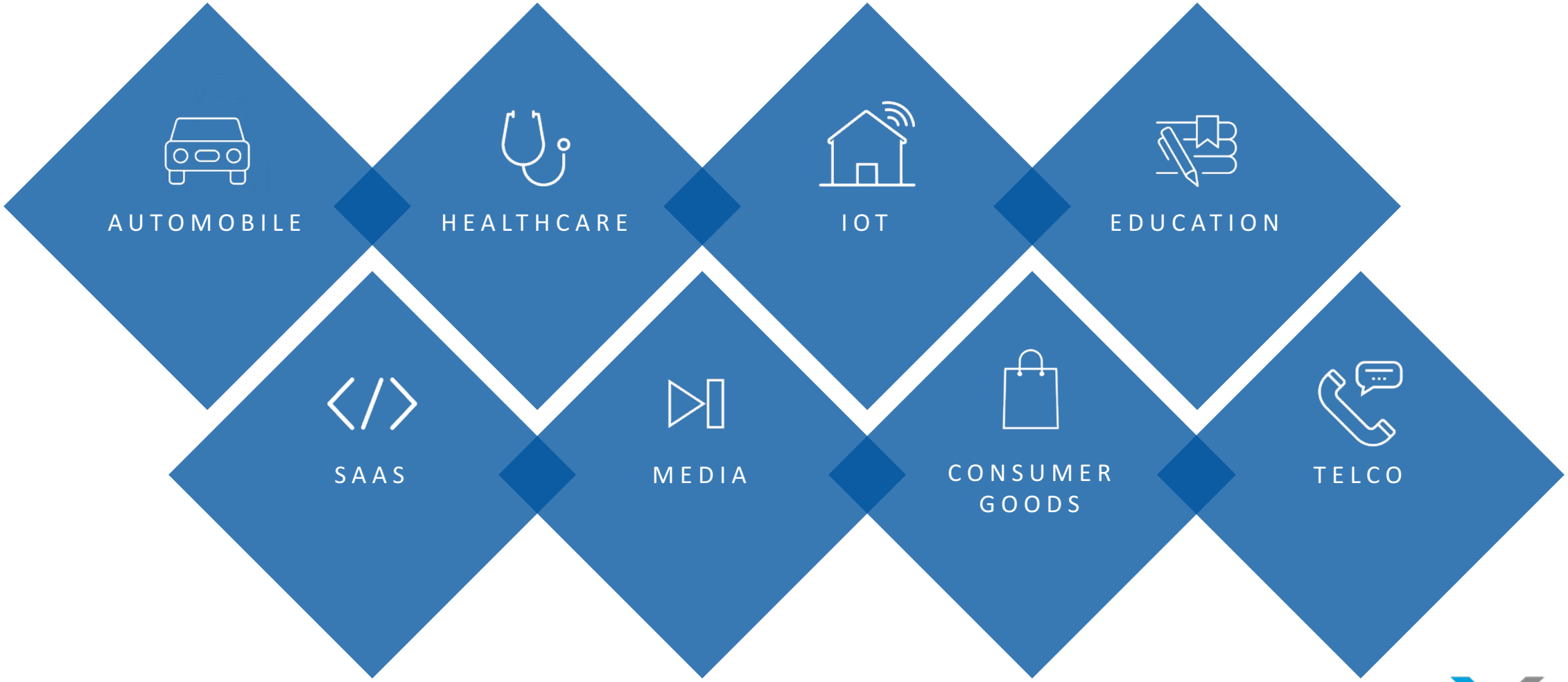
jluszcz@flexera.com

Flexera

Topics

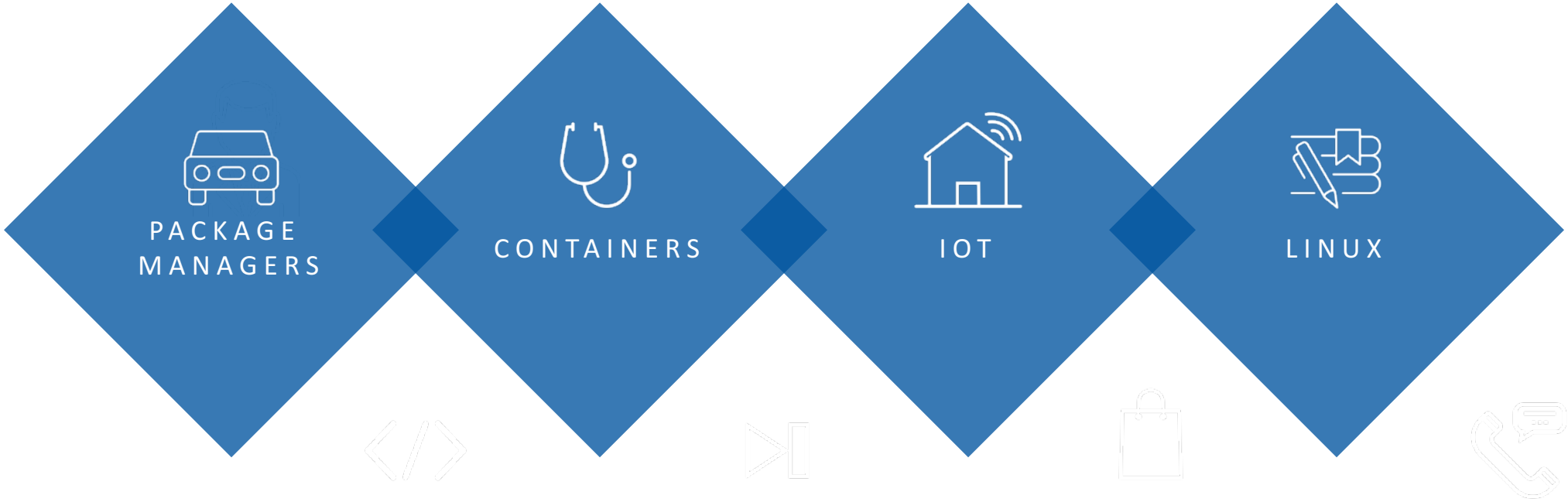
- State of the industry
- A Brief History of Open Source Licensing
- OSS Obligations
- Why do you need a license?
- OSS License Basics
- Distribution models
- Common Misunderstandings
- Best Practices: How are Companies Handling Today?
- Remediation
- Q&A

2018 - EVERY INDUSTRY IS SHIFTING toward OSS



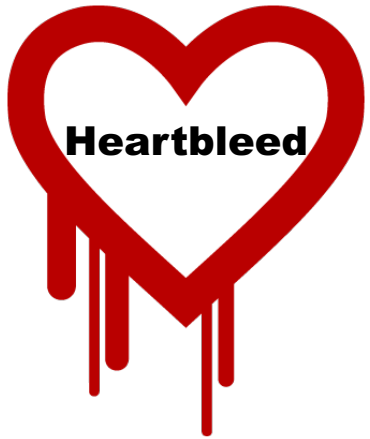
Flexera

The technology stack is changing quickly



Flexera

Software Vulnerabilities are becoming well known



CVE-2014-0160

OpenSSL

- 17% of the Internet's secure web servers (500M) believed to be vulnerable to the attack
- Allowed theft of the servers' private keys, users' session cookies and passwords

- **Typical age: 3-4+ years old**

CVE-2014-6271

GNU Bash

- Potentially affects hundreds of millions of computers, servers and devices
- Shellshock can be used to remotely take control of almost any system using Bash

- **Typical age: 5 years old (seen 13 years!)**

CVE-2015-0235

Linux GNU C Library (glibc)

- Affects almost all major Linux distributions
- Millions of servers on the Internet contain this vulnerability

- **Typical age: 3 years**

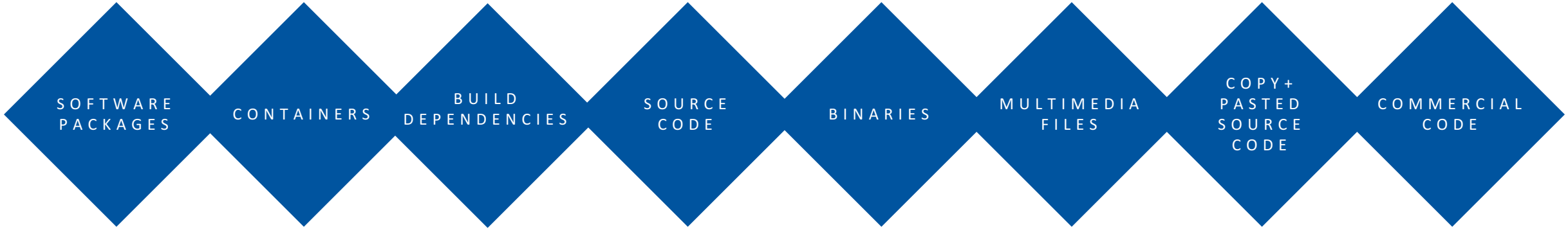
CVE-2017-5638

Apache Struts2

- Remote Code Execution (RCE) vulnerability in the Jakarta Multipart parser
- Allows attacker to execute malicious commands on the server when uploading files
- Exploits are publicly available, simple to carry out, and reliable

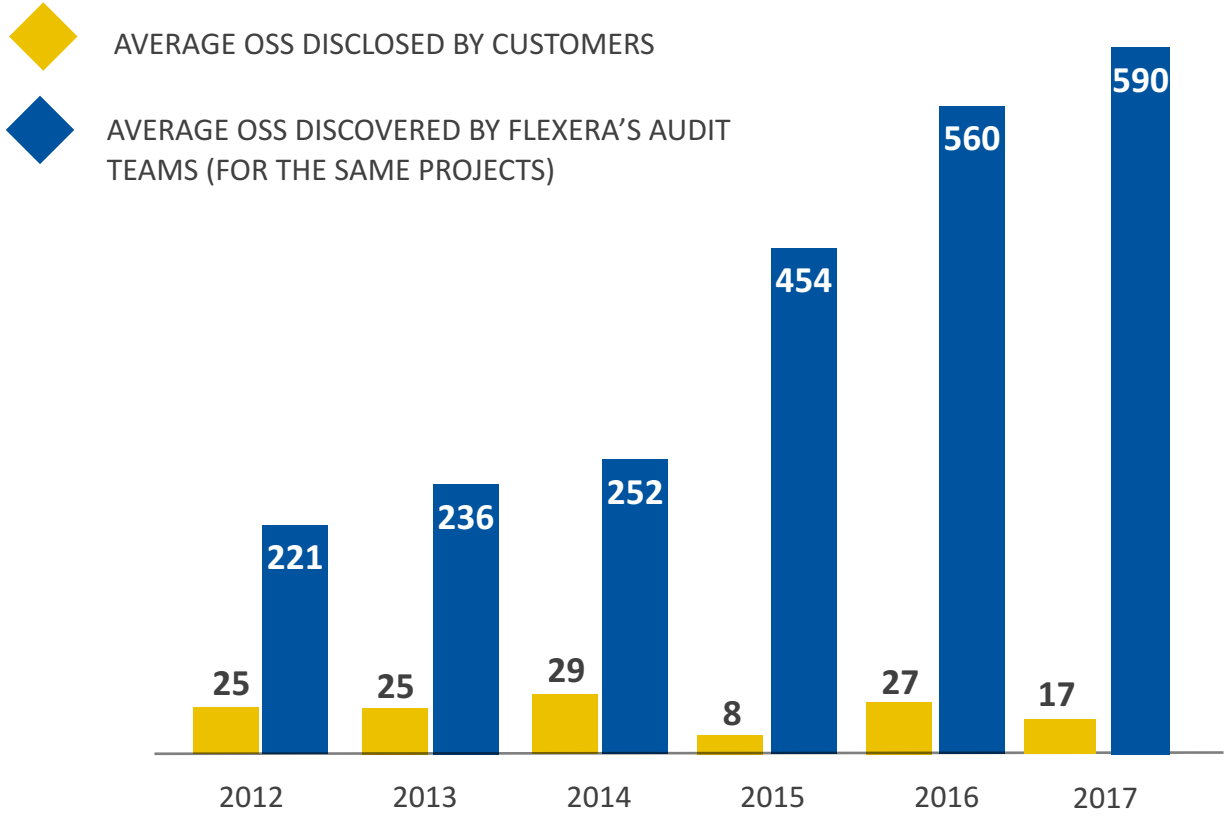
Flexera

THE SOFTWARE SUPPLY CHAIN IS BECOMING MORE COMPLEX

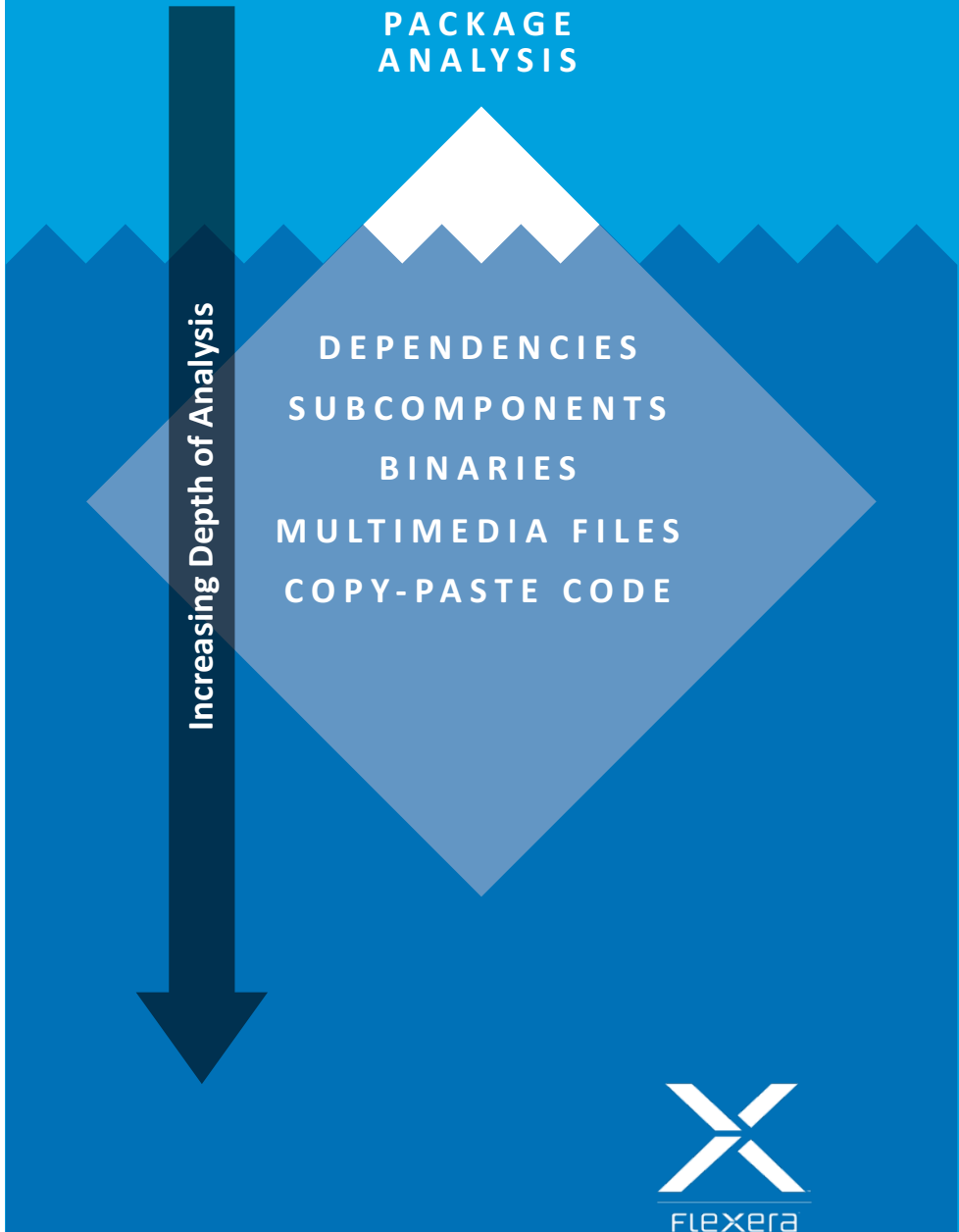


Flexera

THE STATE OF COMPLIANCE IS POOR



Source: Flexera Professional Services Audit data 2012 - 2017



A BRIEF HISTORY OF OPEN SOURCE LICENSING

1940s-1980s Commercial, one-off and public domain dominate
1976 US Copyright Act of 1976
198x “Freeware” and one off licenses
1985 X11/MIT license
1988 first GPL licenses for Emacs/Bison/etc.
1988 BSD license
1989 GPL v1
1991 GPL v2 / LGPL v2
2002 Affero GPL v1
2007 GPL v3 / LGPL v3 / Affero GPL v3

OPEN SOURCE – OBLIGATIONS

Open Source is commonly confused with “Free” as in no cost software

Open source may be Free of Cost, but is not Free of Obligations

Common referred to as “**Free as in Speech, not Free as in Beer**”

Open Source licenses have a list of obligations that **users** must follow in order to legally use the open source library under that license

The act of following these obligations is called OSS Compliance or License Compliance

Your Compliance actions depends on how you are using these OSS components

Most licenses have Multiple Obligations

Flexera

COMMONLY SEEN OBLIGATIONS

Obligation	Type of Obligation	Definition
Share Source	Copyleft aka Viral	Author requires user to share source code
Give Credit	Notice or Attribution	Name of author must be reported in About Box, Documentation, Website, etc…
Share Patents	Patent Clause	Author requires permission to use patents or license patents in this open source project
Restrict use	Restrict who can use this code	Restriction on military use, restriction on nuclear facilities, geography/countries, commercial use etc..
Vanity/One Off Licenses	Give me free beer, say a prayer, Do No Evil	Requests by the author to do some sort of action not typically seen in contracts or licenses
Preserve Attribution	Attribution	Requires attribution/copyrights to be preserved in the source code
Provide Disclaimer	Disclaimer	Explain that the open source author is not responsible for the use of the software, even if it has defects
Supply Original License text	License Text	Requires text of entire license to be provided to users
Commercial Terms	Pay for use of code	Classic software business model license

OPEN SOURCE – TWO COMMON LICENSE PHILOSOPHIES

Copyleft/Viral – Requires release of source code (some or all)

General Public License (GPL)

– You must supply all source if you link against GPL code and distribute the product

Lesser General Public (LGPL)

– You must supply source to linked library if you link against LGPL license library

Affero General Public License (AGPL)

– You must give source away if you use AGPL code and provide Network Access to the product (specifics may murky depending on who you talk to!)

Permissive – Requires a notice in About Box, documentation, source code, NOTICE file, etc..

BSD

MIT

Apache Software License 1.1

Apache Software License 2.0

Flexera

OPEN SOURCE – OTHER LICENSE TYPES

Vanity/One Off Licenses

Free Beer License

(e.g. Poul-Henning Kamp malloc)

```
* "THE BEER-WARE LICENSE" (Revision 42): *   <phk@FreeBSD.ORG> wrote this file. As long as you    retain this notice you * can do whatever you want with      this stuff. If we meet some day, and you think * this  stuff is worth it, you can buy me a beer in return
Poul-Henning Kamp
```

Good Not Evil terms

The author requires you to do “Good” not “Evil” with their software

(e.g. Json.org)

```
The Software shall be used for Good, not Evil.
```

Flexera

WHY DO YOU NEED AN OPEN SOURCE LICENSE?

Copyright law (in many places) means that all source is explicitly copyright the original author EVEN if not marked

You have no right to use someone else's code without permission

Open Source (and commercial) licenses are the way of giving permission to use source code

Lack of license shows lack of maturity for the OSS project, often a sign of other problems!

It is **not** Open Source if you don't have a license

WHAT DOES COMPLIANCE LOOK LIKE?

You provide copyright notices in your About Box, Documentation, etc..
You pass along License text to your users
You provide the source code for GPL, LGPL, etc. modules
You mark changes in source files
You pay required Patent licensing
You pay for commercial libraries as needed
You respect web service SLAs
You do this for every release

WHAT DOES COMPLIANCE LOOK LIKE – LICENSE NOTICES

about:license

Binaries of this product have been made available to you by the Mozilla Project under the Mozilla Public License 2.0 (MPL). Know your rights.

All of the **source code** to this product is available under licenses which are both free and open source. A URL identifying the specific source code used to create this copy can be found on the build configuration page, and you can read instructions on how to download and build the code for yourself.

More specifically, most of the source code is available under the Mozilla Public License 2.0 (MPL). The MPL has a FAQ to help you understand it. The remainder of the software which is not under the MPL is available under one of a variety of other free and open source licenses. Those that require reproduction of the license text in the distribution are given below. (Note: your copy of this product may not contain code covered by one or more of the licenses listed here, depending on the exact product and version you choose.)

- Mozilla Public License 2.0
- GNU Lesser General Public License 2.1
- GNU Lesser General Public License 3.0
- GNU General Public License 3.0
- ACE License
- acorn License
- Adobe CMap License
- Android Open Source License
- ANGLE License
- Apache License 2.0
- Apple License
- Apple/Mozilla NPRuntime License
- ARM License
- Backbone License
- bspatch License
- Cairo Component Licenses
- Chromium License

WHAT DOES COMPLIANCE LOOK LIKE – SOURCE BUNDLES

about:buildconfig

Source

Built from https://hg.mozilla.org/releases/mozilla-release/rev/b6609650a911cbc5267fe8c34f6a8071599d6205

Build platform

target
i686-pc-mingw32

Build tools

Compiler	Version	Compiler flags
cl	1800	-TC -nologo -D_HAS_EXCEPTIONS=0 -W3 -Gy -arch:IA32 -FS -wd4244 -wd4267 -wd4819 -we4553
cl	1800	-TP -nologo -D_HAS_EXCEPTIONS=0 -W3 -Gy -arch:IA32 -FS -wd4251 -wd4244 -wd4267 -wd4345 -wd4351 -wd4800 -wd4819 -we4553 -GR- -DNDEBUG -DTRIMMED -Zi -UDEBUG -DNDEBUG -GL -wd4624 -wd4952 -O1 -Oi -Oy

Configure arguments

--enable-crashreporter --enable-release --enable-update-channel=release --enable-update-packaging --enable-jemalloc --enable-require-all-d3dc-versions --with-google-api-keyfile=/c/builds/gapi.data --with-google-oauth-api-keyfile=/c/builds/google-oauth-api.key --with-mozilla-api-keyfile=/c/builds/mozilla-desktop-geoloc-api.key --enable-warnings-as-errors --enable-eme=adobe --enable-official-branding --enable-verify-mar

COMMON MISUNDERSTANDINGS

Just because code is available, this does not give you any permission to use it.

“Freely Available” != Open Source

“Public Domain” is different than “Open Source”

You still have Compliance tasks even if you don’t ship your product (SaaS or internal use)

Belief that Commercially licensed code has no OSS obligations

MINIMIZATION AND JAVASCRIPT

Most organizations are minimizing their JavaScript to save download time, speed up execution and obfuscate their code

In many cases, only the minified versions of open source JavaScript libraries are being checked into SCM

Additionally, many OSS packages will be concatenated together.

Over minimization is hiding version info and prevents humans for identifying old versions

Always store originals in un-minified form

YOUR DELIVERY METHOD AFFECTS OBLIGATIONS

SaaS vs shipping product (e.g. a distribution)

- Most OSS Licenses only come into effect upon Distribution

Embedded Linux vs Application running on Linux

- Are you shipping Linux or are your users bringing their own?

Client / Server pieces

- Some parts hosted, some parts distributed

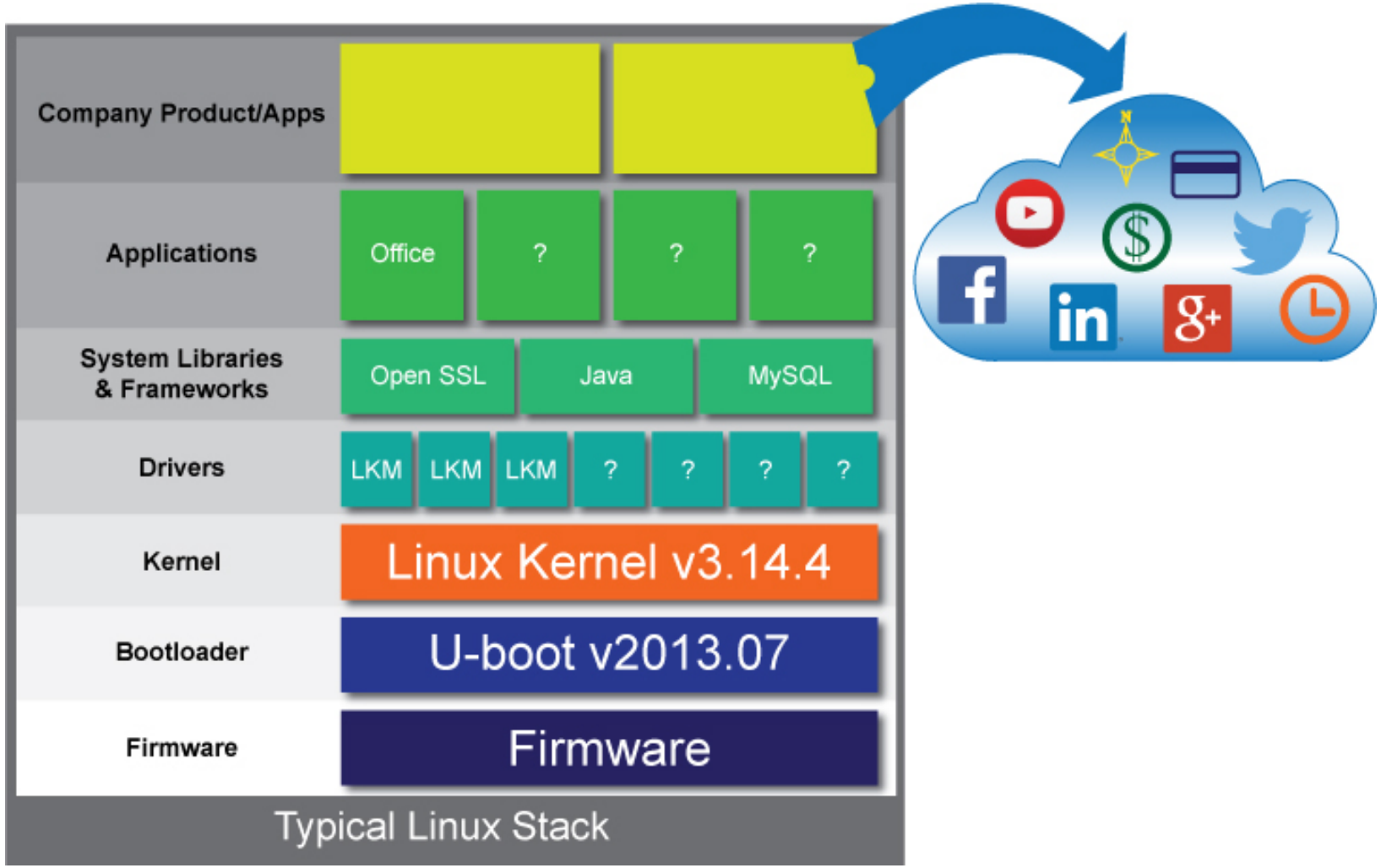
Mobile applications

- Classic distribution with some possible Appstore implications

Web / JavaScript front ends

- The Javascript, HTML, CSS sent to users browsers

YOUR PRODUCT LIVES IN A DEEP STACK OF OSS AND $



FULL LINUX STACKS HAVE MANY OWNERS

The software development team is often different than the release team the puts a product into productions.

Things often fall in the gaps between these teams.

They often have different management, legal contacts, understanding of OSS licensing.

Companies many know some OSS from the release team (e.g. Linux, Apache httpd, MySQL, etc..) or some OSS from the software team (zlib, openssl, etc..) but not always from both.

A “good” list for the release team is often confused for a “good” list for the actual product.

Linux distributions often lead to long lists of OSS components but not always a clear understanding of the company’s OSS choices in the product.

LINUX: COMMON AREAS OF CONCERN

- **Linux can be complicated and contain many moving pieces**

- **The base for the OSS often comes from the outside**

- **While Everything is required to be declared, this is often hard**

- **Components that MUST be declared**
  - Linux Kernel
  - Busybox
  - iptables / ipchains
  - U-boot
  - Multimedia & Codecs (e.g. ffmpeg, h264, etc..)

Additions to your base Operating System (e.g. RPMs, etc..)

Modifications to Device Drivers

Flexera

COMMON AREAS OF CONCERN

While it is best to have a “Full” accounting of all third party software, certain components may have a higher priority than others.

1) Linux related technologies w/ GPL licensing
2) Cryptographic components – often highly targeted, and also have legal tracking requirements for export anyway
3) Compression components – similar to cryptography in terms of usage and programming techniques. Often highly targeted.
4) Multi-media components. Wildly used, often contains crypto and compression routines themselves. Patent concerns
5) Applications Platforms – widely used, often contain crypto and compression, complex
6) Databases – central to all systems, complex

QUESTIONS TO ASK YOUR DEVELOPMENT TEAM

- ❑Do we have a list of the open source and commercial libraries we are using?
- ❑How deep have we looked? How complete is this list?
- ❑What Cryptography, Compression, Multimedia and Application Server libraries are we using?
- ❑Does this lists include all libraries brought in though repository managers like Maven / Ruby Gems / npm, etc…?
- ❑Do we have a list of all the web services we depend on? (e.g. credit card processors, stock price lookup, etc…)
- ❑What Databases are we using? (including sql, nosql, embedded, etc..)
- ❑Do we ship VMs or Applications to our customers? What OS, OSS components and software stack are we shipping?
- ❑What is the “Full Stack” required to run our product – including the OSS, DB, etc…?
- ❑Do we have a “Disclosure List” from our commercial vendors

Commercial Compliance Issues

COMMERCIAL COMPONENTS COMPLIANCE ISSUES

Commercial components are not often well marked, often move around

Get a list of known commercial components / check names / paths

Commercial components often contain large amounts of undeclared OSS code

All commercial components should come with a disclosure list of OSS that it uses

Push for such a list in contracts and via email discussions w/ a vendor

It's usually not your job to perform a **full review** but you may have to

Find 1-5 undeclared OSS components to "force the issue" as needed

- Zlib / libpng / openssl / glibc / ffmpeg are all good candidates for easy discovered undisclosed OSS components

Flexera

SUPPLIERS CODE AND SDKS COMPLIANCE ISSUES

- You may also receive source code from Commercial companies
- Vendors do not always mark code as clearly as they should
- GPL code will be right next to Commercial or GPL/Commercial code
- Often open source code is NOT marked and its licensing is unclear
- Know your contact person and have a process for logging IP bugs or Questions
- Developers often get confused about whether this code is commercially or GPL licensed

Flexera

DEALING WITH COMMERCIAL COMPONENTS

- Binary analysis is often needed
- The suppliers code may be in a special format (encrypted, stripped of symbols, compressed, etc…) see if you can get an unmodified file from before these modifications were performed
- Push for an independent outside review as needed
- Set a contractual standard for disclosure levels
- Understand that Linux OS full system compliance is difficult and the use of “ALL” in contract language may be difficult to enforce

Flexera

SaaS Compliance Issues

WHAT'S DIFFERENT ABOUT SAAS?

Traditionally software is distributed to end users through physical means (via CD, embedded device, download, etc…)

Classic open source and commercial licenses were written with this in mind.

Many open source licenses only come into effect with a classic distribution (esp. many people's concern the GPL)
This is sometimes known as the "ASP loophole"

SaaS projects are not distributed in the classic way but instead run on a network server

Users come to the software instead of the software coming to the users.

WHAT'S DIFFERENT ABOUT SAAS? (CONT.)

Because of the perceived reduced compliance needs around the GPL many companies stopped or reduced urgency in tracking OSS licensing for SaaS projects.

Little or no credit was being giving to the OSS backbones of popular SaaS products and changes were not being passed back to the community.

This lead to concern in the OSS community about "Free Riders"

Members of the OSS Community responded with the Affero General Public License (AGPL) in 2002 and updated it in 2007.

WHAT IS THE AFFERO GPL / AGPL?

The AGPL was designed to close the ASP loophole by treating network access as similar to a distribution.

The basic intent is to require source code for the entire application to be offered to the end users.

COMMON AGPL-STYLE LIBRARIES

The most common AGPL style libraries we see are:

- iText PDF generation library (dual licensed AGPL or commercial)
- MongoDB (Dual license AGPL w/ exception or Commercial)
- Berkeley DB/Sleepycat (now AGPL or Commercial)
- Funambol (AGPL or Commercial)
- Ghostscript (now AGPL or Commercial)
- Noe4J (GPLv3/AGPL or commercial)
- Magento (OSL – similar to the AGPL)

Many of these are dual licensed with commercial options.

SAAS COMPLIANCE – TOP CONCERNS

Untracked Libraries with Vulnerabilities – old versions of OSS libraries

The AGPL is the classic OSS concern for SaaS vendors

Other AGPL like licenses include:

- Common Public Attribution License
  http://en.wikipedia.org/wiki/Common_Public_Attribution_License
- Open Software License
  http://en.wikipedia.org/wiki/Open_Software_License

Other licenses that require review and compliance include:

- Commercially licensed libraries and tools
- Components marked “Not For Commercial Use”
- Components with restrictions on types of use (e.g. no military use)
- Licenses based on use, not just distribution
- Web attribution licenses (e.g. put a link on your homepage)
- Components with Unknown license terms

OTHER SAAS COMPLIANCE ISSUES

Images, Icons, Fonts and Sounds

People are very good at recognizing these types of resources and their history often gets confused by the developers

Javascript and CSS

Often treated as a distribution with all the classic compliance requirements

Patent Licenses

Certain technologies like MPEG or other codecs may require license fees even if open source libraries are providing the functionality

Private Installations

Certain large customers may require private installs.

These are a classic distribution

CONTAINER / PRIVATE CLOUD / PUBLIC CLOUD ISSUES

Business models change, sometimes overnight.

“Everyone” is a SaaS-only company until they get at least one very large company who wants a privately hosted version

SaaS projects often have many more GPL dependencies than a classic application and are hard to refactor or fix when going “Private” and trying to comply with Distribution-style obligations

The time scales for reviewing OSS dependences is often very short, sales team driven, not development team driven.

We found things we shouldn’t be using; Now what?

HOW ARE COMPANIES HANDLING TODAY?

Option 1:

Remove and rewrite / get new OSS

A company may remove the rejected code and rewrite / re-implement the feature with new code

Very common during M&A and for risk adverse orgs

<u>Risks / Drawbacks:</u>

Time require for rewrite

“Dirty-room” re-implementations

New code’s license may be no better

HOW ARE COMPANIES HANDLING TODAY?

Option 2:

Contact Author and ask for license

A company may try to contact the author and ask / suggest an acceptable license (commonly MIT/BSD)
Sometimes through an intermediary (outside legal)

Risks / Drawbacks:

Author is now aware of use
Author may desire stronger license than you
Author may require Commercial license
The license is longer than the code!

HOW ARE COMPANIES HANDLING TODAY?

Option 3:

Wait and See

A company may decide to do nothing, ship software and see if problems occur

Common for old code & risk tolerant orgs

Risks / Drawbacks:

Copyright infringement
License problem if forced to comply
Can’t properly disclose OSS licenses

PICKING ALTERNATIVES TO REJECTED OSS LIBRARIES

- **In many cases GPL v2 or GPL v3 libraries are appropriate and expected (especially lower in the stack)**

- **If you expect to keep your source closed you will more likely be required to remove GPL licensed code if found in these closed areas**

- **Picking an appropriate Alternative library has certain considerations**
- You development team is likely the best team to pick an alternative

- Legal should specify allowed licenses (e.g. MIT/BSD/Apache/Commercial)

- Legal should specify forbidden licenses (e.g. GPL/Affero/CC-SA)

- If an open source project can NOT be found, a “build” decision is made

- Projects will sometimes (rarely) provide commercial re-licensing of GPL code

- Do not let your team try to “relicense ” the project without permission

Flexera

WHAT IS SOFTWARE COMPOSITION ANALYSIS?

Today, developers are leveraging more than 50% of open source software (OSS) in their proprietary applications to speed up time to market and drive innovation.

Security risk - Vulnerable OSS Components

IP risk - Non compliance with OSS obligations

Reputation

Flexera

BEGIN BY ESTABLISHING A PROCESS FOR SCA

Start with automated high level analysis of all your code.

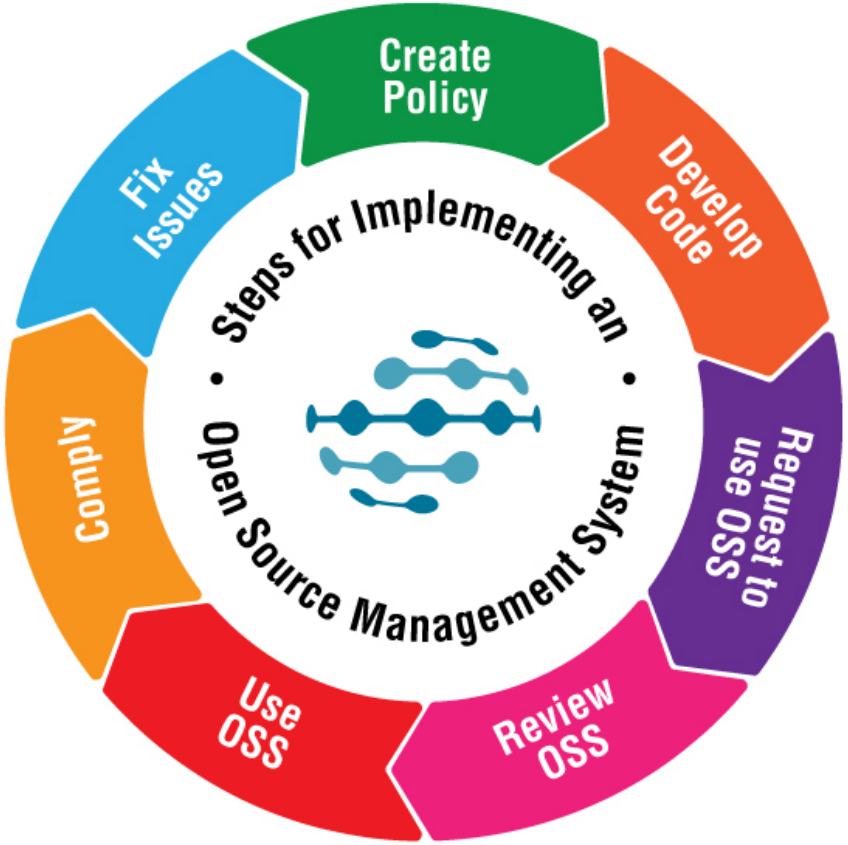
GET A TEAM IN PLACE

FIX HIGH PRIORITY ISSUES

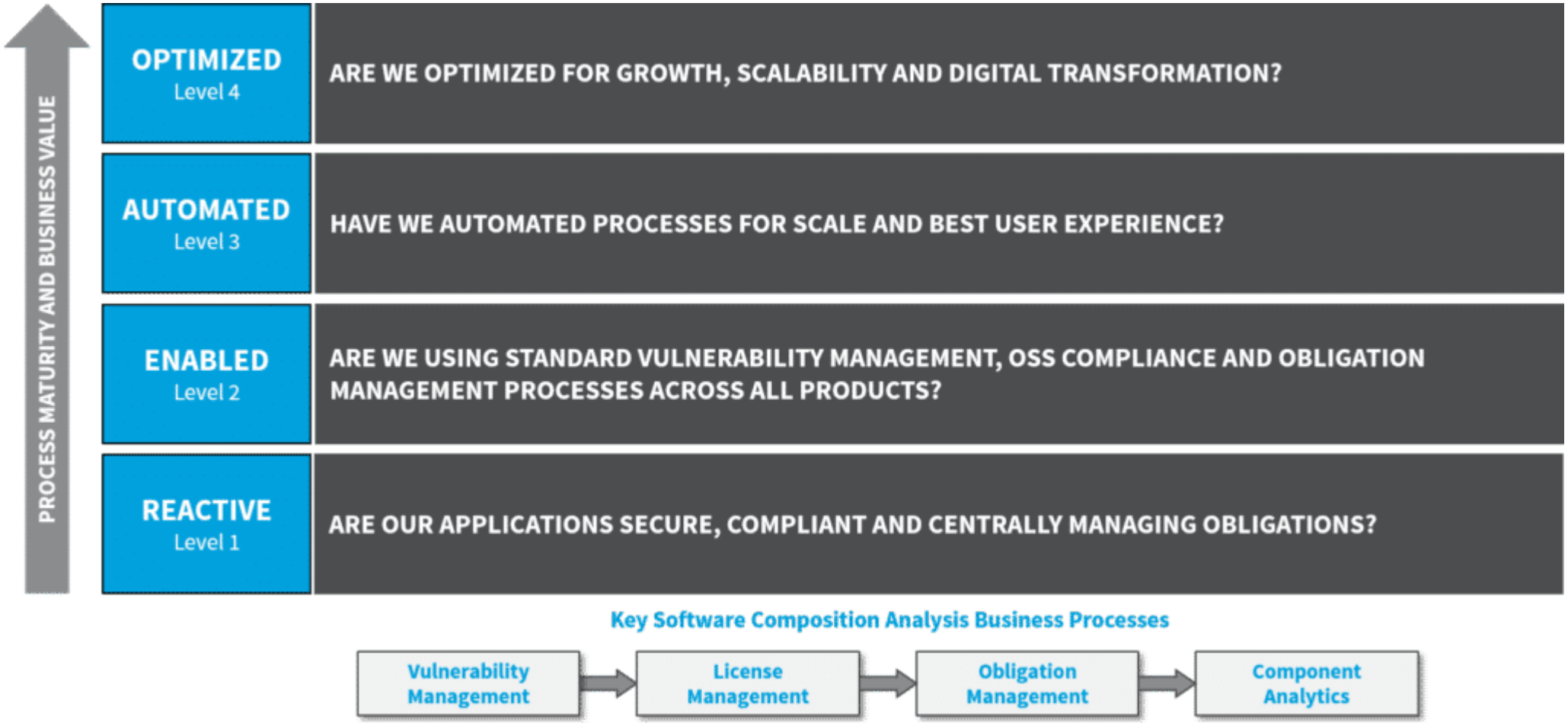
PAY ATTENTION TO VULNERABILITY ALERTS

+ **Get detailed** analysis of high risk code.

CREATE A PROCESS THAT WORKS FOR YOUR COMPANY



HOW MATURE IS YOUR SCA PROCESS?



Key Software Composition Analysis Business Processes

Vulnerability Management → License Management → Obligation Management → Component Analytics

Flexera

FLEXERA OSS AUDIT TEAM

No Disclosures

87%

M&A Audits

41%

Baseline Audits

49%

16%

Priority 1 Issues
eg. GPL, APGL

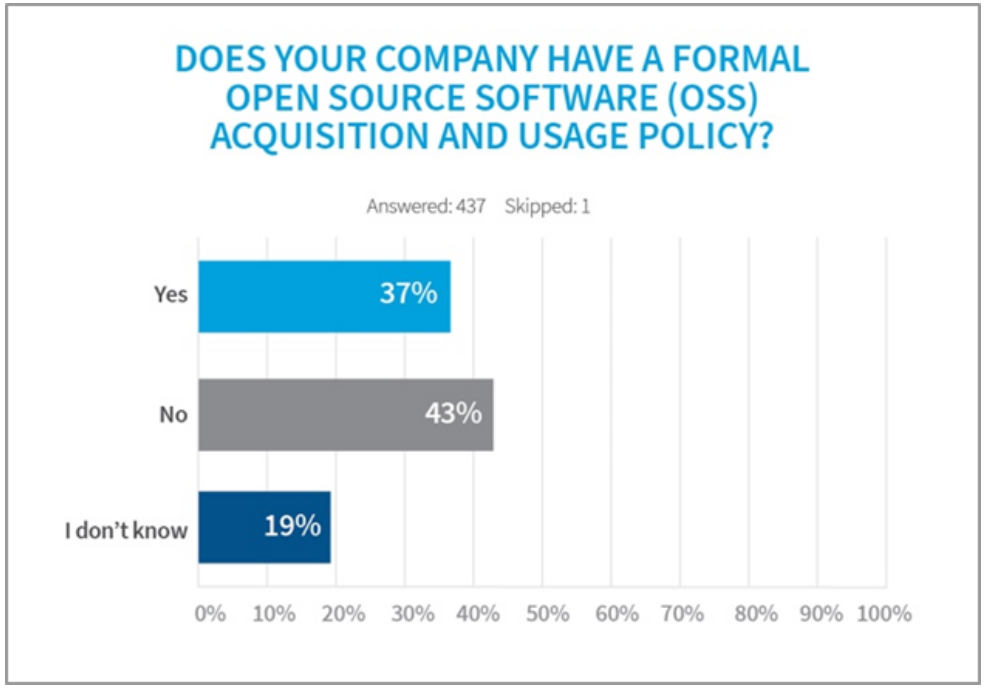
11%

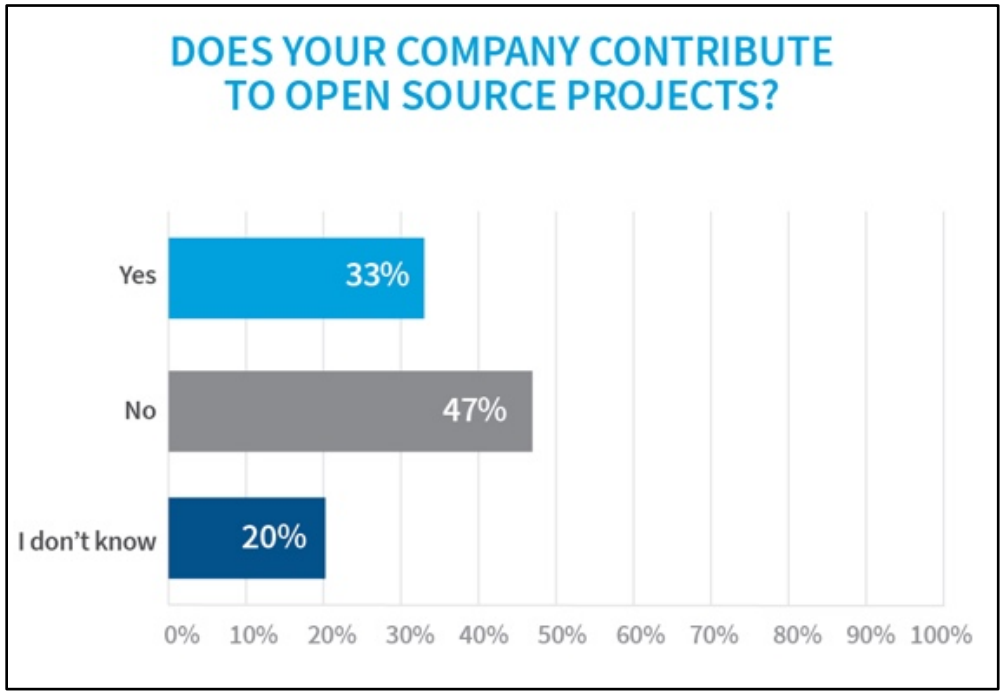
Priority 2 Issues
eg. commercial, unknown

Flexera

FLEXERA SURVEYS THE INDUSTRY

Increasing Open Source usage and lack of Open Source governance

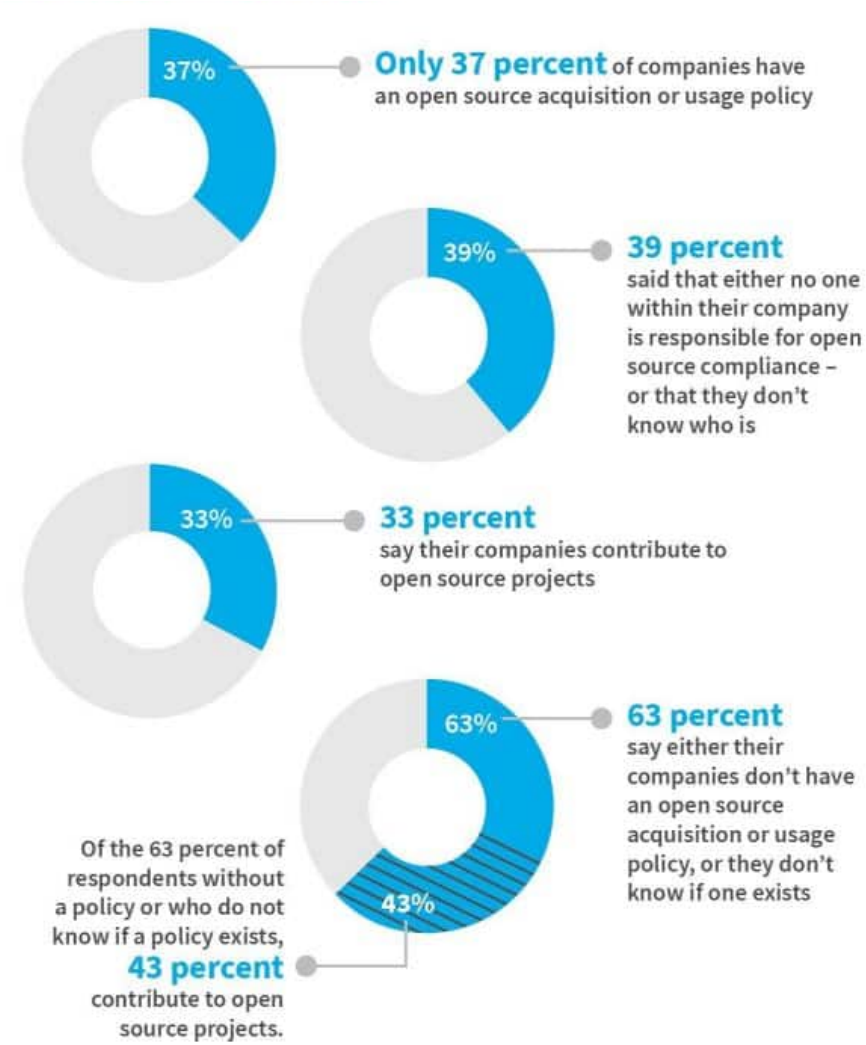




Many individuals and companies contributing to open source projects lack their own internal open source acquisition and usage policies. **43% of developers contributing to OSS are not aware of a formal OSS usage policy.**

Flexera

SETTING STANDARDS



Questions to ask your teams

"Are we using the latest version of Apache Struts 2?"

What if a customer said "Our IT dept refuses to deploy any applications with OpenSSL"?

"Are we vulnerable to that CVE in the news?"

Flexera

THANK YOU

@ JLuszcz@Flexera.com

www.flexera.com/sca

@jeffluszcz