

# Everything You Need To Know About Free & Open Source Licensing

McCoy Smith  
Intel Corporation

# “Source”

A human-readable, -understandable, and -alterable abstract representation of some technological “thing”

Software: source code

“Hardware”: various abstraction languages  
(inter alia: Hardware Description Language)

# “Open Source”

A licensing model in which the “source” for a technology is:

- Made available
- Licensed without “restrictions”\*
- Licensed without the requirement to pay royalties†

\*But see copyleft

†But see “BSD-FRAND” hot topic

# “Free” Software

Similar and/or identical to Open Source

Free Software Foundation (FSF) vs.  
Open Source Initiative (OSI)

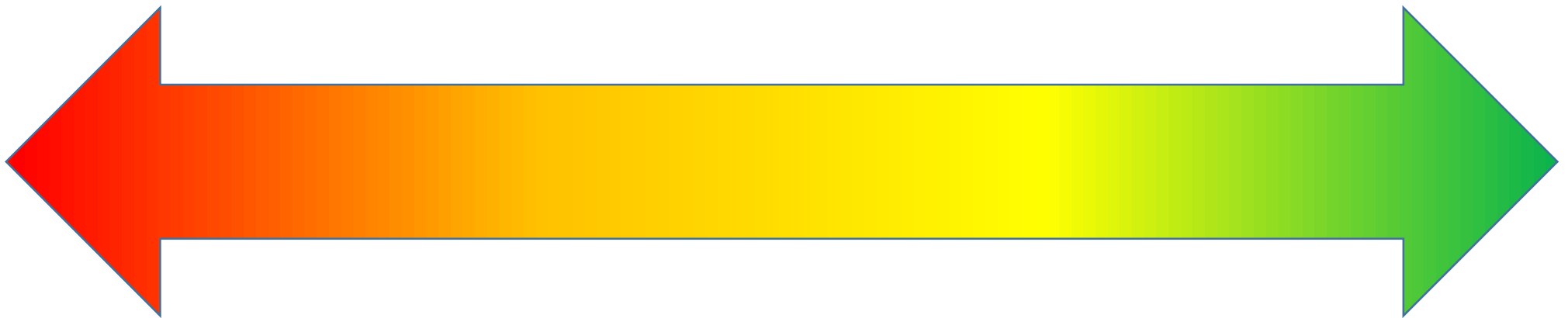
Strong philosophical debates about the  
difference, maybe only interesting to advocates

# The Continuum

**“Strong Copyleft”**

**“Weak Copyleft”**

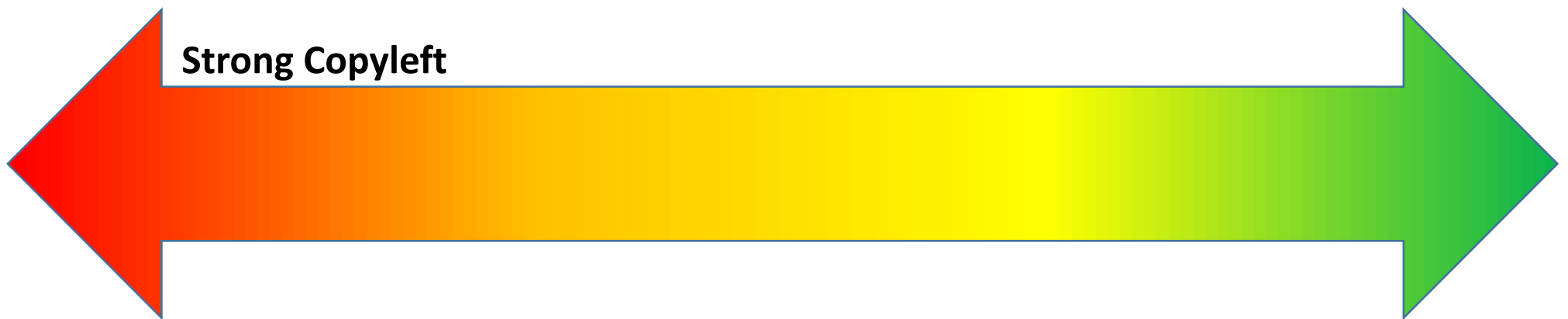
**“Permissive”  
(aka “Academic”)**



# The Continuum

## “Strong Copyleft”

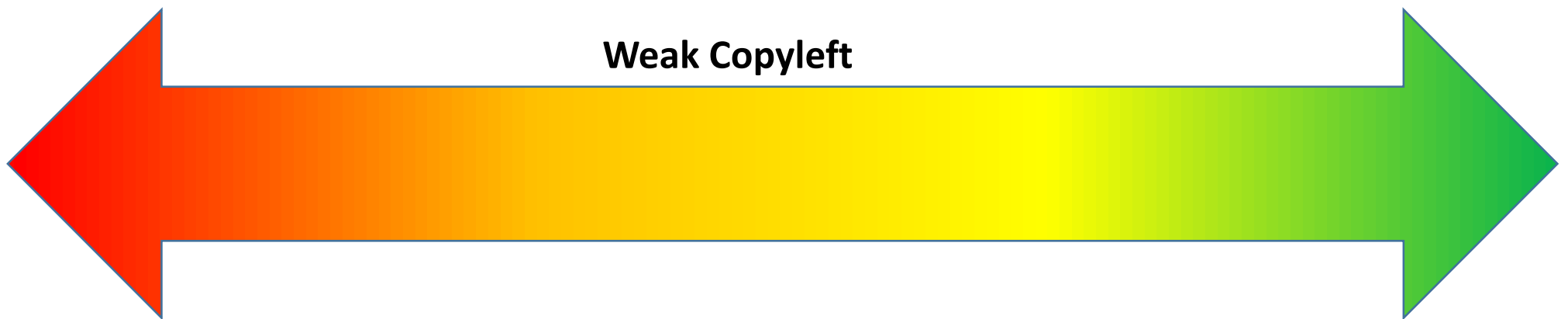
- Copyright derivatives must also be open source
  - What exactly is a derivative?
- What “triggers” this obligation depends on the text of the license
  - Most (but not all) licenses: distribution



# The Continuum

## “Weak Copyleft”

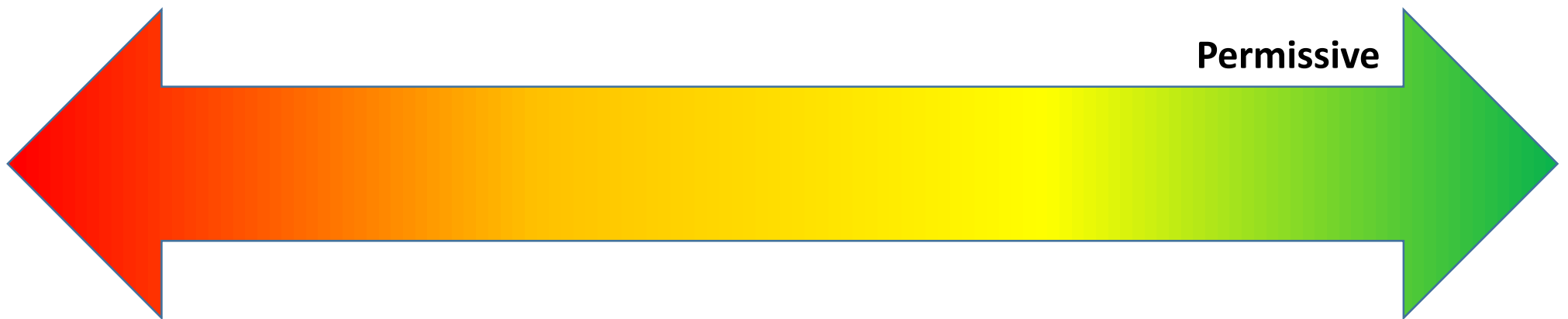
- **Some** derivatives must also be open source
  - Which derivatives is license-dependent
- Non-open source may interact even if a derivative
  - Depends on license, and mode of interaction



# The Continuum

## “Permissive”

- Relicensing under different terms (including non-open source) allowed
- Minimal attribution requirements
  - Compliance is easy; but non-compliance does occur (*see Jacobsen v. Katzer*)





# The Continuum

**“Strong Copyleft”**

**“Weak Copyleft”**

**“Permissive”**

Affero GPL

GPLv3

GPLv2

LGPL

Eclipse

Mozilla

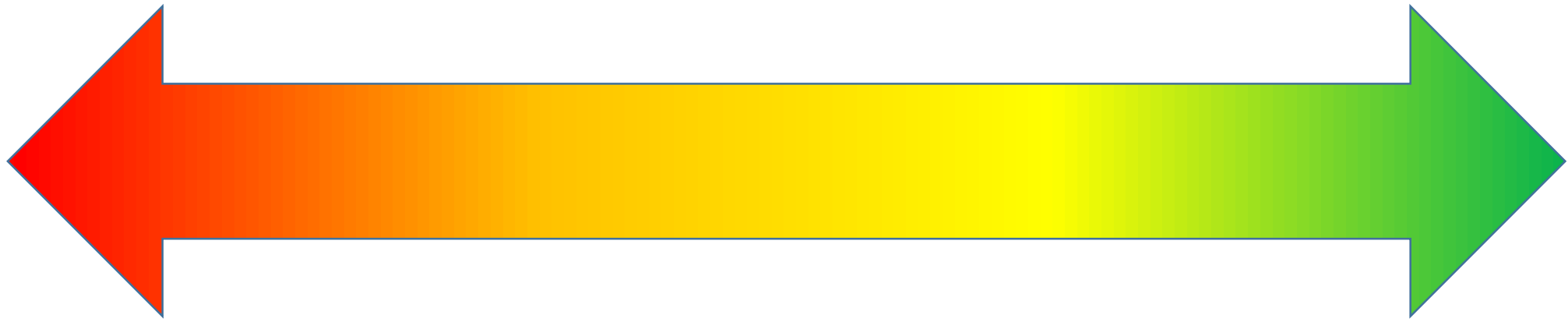
BSD

MIT

Apache

BSD+Patent

UPL



[Plus many others across this continuum]

# Client Advice Basics

- **“If you’re writing software, you need advice on open source”**
  - Open source is pervasive
    - Clients who say “we don’t use it” usually are wrong
- **“License compliance needs to be part of your build process”**
  - Having a compliance program is important
    - If clients don’t have one, standard programs now available
      - E.g., OpenChain – [openchainproject.org](http://openchainproject.org)
    - Tools are available, although standardizing is the next phase
      - E.g., Automated Compliance Tooling – Linux Foundation

# Client Advice Basics

- **“Mixing open source is complicated, and sometimes impossible”**
  - Avoiding license conflicts can be a mess
    - Copyleft licenses generally don’t mix well
    - Sometimes even permissive licenses don’t mix well
      - I.e., Apache → GPLv2
        - Free Software Foundation says this mix violates GPLv2
- **“If you have a patent portfolio, pay attention to the license”**
  - Patents can be a key concern
    - Study how license does (or doesn’t) handle patents
      - And this includes how implied patent licensing might apply
    - Also consider defensive termination clauses
    - See also BSD+FRAND issue

# Client Advice Basics

- **“I need to see your stack diagram, and to talk to someone who can describe how the modules interact”**
- Tests for: a) software derivative works, b) software copyright eligibility, c) software copyright fair use, makes giving definitive advice difficult
  - Advice is often a mix of legal and software programming analysis
  - Different programming paradigms have different ways in which code interacts, and case law – and licenses – don’t always clearly set forth whether, and how, a license might apply
    - See API hot topic issue

# Hot Topics

- **APIs**
- **Defensive patent termination**
- **Copyright “trolling”**
- **Copyleft & SaaS/CSPs**
- **BSD/MIT & FRAND Standards**
- **Importing “ethics” into licenses**
- **Open hardware**

# APIs

- The continuing *Oracle v. Google* saga
  - Are “APIs”:
    - Copyrightable?
    - Subject to fair use?
    - Sufficient to create a derivative if replicated in interacting code?
  - And what counts as an “API”?
  - Will we get a comprehensive (and nationwide) test for what is copyright infringement for software?
- Outcome of case (if Supremes take it) could change how advice on copyleft license compliance is given
  - What used to be thought to impose copyleft requirements may no longer do so

# Defensive patent termination

- Many newer open source licenses contain this concept
  - “If you institute patent litigation against this software, any [patent/IP] rights granted to you in this software are terminated”
  - See, e.g., Apache, GPLv3, Mozilla, Eclipse
- Generally considered to create “patent (or IP) peace” around a software project by anyone participating in or using it
- Some defensive patent termination clauses considered to be overreaching and imbalanced
  - “If you institute patent litigation against ***any of [the author’s] products***, any [patent/IP] rights granted to you in this software are terminated”
  - Controversy over React.js license by Facebook (“BSD+Patents”)
    - Mirrors controversy over Common Public License 15 years ago

# Copyright “trolling”

- Open source contributor decides to monetize license violations
  - A copyright counterpart to “patent trolling”
- (Currently) a phenomenon only in Germany
  - Model depends somewhat on quirks of German civil procedure
- Scenario
  - Company fails to comply with (typically) GPL source code requirements
  - Author of GPL code settles with company in return for cease-and-desist
  - Author later claims violation of cease-and-desist, either for trivial violations or violations that may not even be requirements of the license
  - Escalating monetary penalties requested
- Opposition is tricky, and needs coordination with German counsel
  - See article by Marcus von Welser:  
<https://ifosslr.org/index.php/ifosslr/article/view/128/246>
- See also GPL Cooperation Commitment



# Copyleft & SaaS/CSPs

- Most open source license trigger obligations upon distribution of source or binary code
  - Internal use generally without restrictions or obligations
- “Affero GPL” license changed this paradigm
  - Network access to code triggers obligation to provide source
  - Directed to SaaS business models
- New licenses even more aggressively targeted at SaaS and Cloud Service Providers
  - “Server Side Public License” and “License Zero Public License”
  - Trying to reach into back-end stack even if customers never get network access to the open source code
- Vigorous debate about the limits of copyleft

# BSD/MIT & FRAND Standards

- BSD & MIT licenses:
  - Early (80s) permissive licenses
  - No mention of patents; no (or little) discussion of patent statutory rights
  - Pervasive use in open source projects, particularly older ones
- FRAND Standards:
  - “Fair Reasonable & Non-Discriminatory” licensing obligations
  - Used for standards where members wish royalty payments from adopters
- Some advocate use of BSD or MIT to license code developed in FRAND standards
- Theory: BSD & MIT are copyright-only licenses, therefore allow an additional royalty-bearing patent license for those using the BSD/MIT code
  - See David Kappos article in Colum. Sci. & Tech. L. Rev:  
[stlr.org/2018/10/15/the-truth-about-oss-frand-by-all-indications-compatible-models-in-standards-settings/](http://stlr.org/2018/10/15/the-truth-about-oss-frand-by-all-indications-compatible-models-in-standards-settings/)
  - Also see Van Lindberg response:  
[stlr.org/2019/03/04/oss-and-frand-complementary-models-for-innovation-and-development/](http://stlr.org/2019/03/04/oss-and-frand-complementary-models-for-innovation-and-development/)

# Importing “ethics” into licenses

- Open source & free software licenses disallow prohibitions on certain “fields of endeavor”
- Recent proposed licenses for AI include restrictions against certain uses perceived to be unethical
  - I.e., racial, genetic or criminal profiling
  - Such licenses, per rule above, cannot be called “open source”
- Vigorous debate about whether rule above should change
  - Which is more important – tech ethics or open source purity?
- Developers continue to struggle with thorny issues around how software is being used
  - See various internal employee protests about certain military or immigration enforcement uses for company’s technology

# Open hardware

- Can open source software principals be used for hardware?
- Efforts span at least 15 years
- License terms used are often those created for software
  - Concepts aren't always directly transferable to hardware
    - Software licenses often heavily dependent on exercise of copyright rights
    - Patent rights may be more important in hardware
    - How does one comply with copyleft license for hardware?
      - What are “source” and “binaries”/“executables”?
- See Katz survey on current license usage for open hardware:  
<https://ifosslr.org/index.php/ifosslr/article/view/130/248>