

Morgan Lewis

# TECHNOLOGY MARATHON

## Open Source Software and Litigation Risks

Douglas J. Crisman  
Corey R. Houmand

June 2, 2022 | 12:00-1:00 pm ET

# Presenters



**Douglas J. Crisman**



**Corey R. Houmand**

**Morgan Lewis**

# Agenda

- Software Copyright Fundamentals
- Software Licensing – Open Source
- Open-Source Overview
- Open-Source Benefits
- Open-Source Risks
- Litigation Risk Examples
- Common Open-Source Licenses
- Open Source Litigation Issues
- Other Open Source Risk Areas
- Best Practices and Practical Takeaways

# Software Copyright Fundamentals

- Copyright law (in the US and elsewhere) protects “original works of authorship” “fixed in a tangible medium,” such as software and documentation. 17 U.S.C. §102(a)
- Under US Copyright law, the author of an original work owns the copyright in the work (unless the work is made “for hire”). 17 U.S.C. §102(a)
  - The single copyright in a joint work is co-owned by the authors
  - The copyrights in separate works in a collective work are owned by their respective authors
- The Copyright is automatically granted to the author – there is no requirement to register a copyright, but registration provides key advantages:
  - Prerequisite to litigation. 17 U.S.C. §411(a)
  - Statutory damages (\$750-30,000 per work infringed, up to \$150,000 if willful). 17 U.S.C. §504
- The use of Copyright Notices is optional, but helpful in litigation or when asserting infringement (or in figuring out who the copyright owner is):
  - E.g., “Copyright 2022, Wile E. Coyote”

# Software Copyright Fundamentals

- The owner of copyright has the exclusive right, among other rights, to do and to authorize others to do the following:
  - **To reproduce the work in copies or phonorecords (17 U.S.C. §115);**
  - **To prepare derivative works based upon the work (17 U.S.C. §§ 103(b), 106(2))**
- A violation of any of the exclusive rights of the copyright holder is a copyright infringement, unless fair use (or a similar affirmative defense) applies.
- Copyright term:
  - **Works created since 1978: term is life of the author plus 70 years, unless work is “work for hire,” then term is shorter of 120 years after creation or 95 years after publication (1998 Copyright Term Extension Act – lobbied for by Disney to prevent early version of Mickey Mouse – published in 1928 – from falling into public domain)**
  - Works created before 1978: it’s complicated
  - Works created before 1926: public domain

# Software Licensing – Open Source

- Open Source Software (OSS) is original work of authorship - > subject to Copyright
- OSS can only be used per License Terms imposed by owner of the Copyright
- Check Copyright/License Notice to confirm owner of work and applicable open-source license
- Review License Terms and be sure to comply with them
- ***Violation of license could expose company to breach of contract and Copyright misappropriation claims.***
- ***Could result in injunction, monetary damages, including statutory damages, and contamination of code base with unlicensed third-party code.***

# Open-Source Software Overview

- Source code freely shared with other programmers subject to an Open-Source License
- It is ubiquitous
  - Per Synopsys, 84 open-source components per commercial application in 2016 to 528 in 2020
- For example:
  - Linux (operating system) (GPL v2)
  - Apache (web server) (Apache License 2.0)
  - MySQL (relational database) (GPL v2)
  - Perl (scripting language) (Artistic License and GPL v2)
  - OpenStack (cloud computing platform) (Apache 2.0)
  - Apache Hadoop (framework for big data) (Apache 2.0)
  - R (statistical computing language) (GPL v2)

# Open-Source Benefits

- Rapid Deployment
- Low Cost
- Open
  - Available
  - Modifiable
  - Maintainable
  - Reliable
  - Secure
- Community
  - Pride of Ownership
  - Peer Development
  - Partnership (individual/non-profit/corporate)
  - Outsource Coding
- Continual Improvement
- Open Standard



# Open-Source Risks – Code

- OSS Provenance?
- No support
- No warranty
- Poorly funded → poorly maintained
- No differentiation
  - Common features
  - Hard to customize
- Vulnerabilities are public
- Out of synch with company needs
  - Bug fixes?
  - New features?
  - Roadmap?
  - Need to update every new release with company customizations/patches
- Community
- Taint proprietary code base and vice-versa if intermingled

# Open-Source Risks – Licenses

- Could be viral (e.g., GPL/copyleft licenses)
- Non-negotiable
- As is
- Quirky
  - can include explicit patent licenses
  - can include publicity conditions (i.e., if publicize feature enabled by OSS, need to credit author)
  - can limit use to specific situations (e.g., academic but not commercial uses)
- Gotchas
  - distribution trigger (no copyleft effects unless distributed – and definition of “distribution” varies)
  - code combination (entire work subject to OSS license (GPL), or just derivatives of licensed files (MPL))
- Ambiguous (rarely enforced or subjected to legal interpretation)
- Enforcement
  - “political”
  - public – can be embarrassing even if risk is manageable

# Litigation Risk Example – Hyper-V (2009) (License Risk)

- Linux driver code (GPL v.2) incorporated in proprietary Hyper-V Linux driver code
- Discovered by user of Hyper-V driver code and reported on Linux Internet blog:
  - “This saga started when one of the user’s [sic] on the Vyatta forum inquired about supporting Hyper-V network driver in the Vyatta kernel. A little googling found the necessary drivers, but on closer examination there was a problem. The driver had both open-source components which were under GPL, and statically linked to several binary parts.” *Network Plumbers Journal*, July 20, 2009.
- Result – Proprietary Hyper-V drivers were open-sourced by owner:
- Lesson 1: Training is important – Coder apparently had access to GPL code and then used it inappropriately.
- Lesson 2: Mixing of open source and proprietary software creates a litigation risk that the proprietary software may need to be disclosed under an open source license.

# Litigation Risk Example – Heartbleed Bug (2014) (Code Risk)

- Heartbleed
  - Bug in OpenSSL (open-source toolkit used to provide secure communications between web clients/browsers and websites)
  - Could be used to capture passwords
  - Affected nearly 2/3 of Internet (not banks or gov't)
  - Public announcement at Openssl.org:
    - “A missing bounds check in the handling of the TLS heartbeat extension can be used to reveal up to 64kB of memory to a connected client or server (a.k.a. Heartbleed).” “Fixed in OpenSSL 1.0.1g (Affected 1.0.1f, 1.0.1e, 1.0.1d, 1.0.1c, 1.0.1b, 1.0.1a, 1.0.1).”  
<http://openssl.org/news/vulnerabilities.html>
- Lesson 1: Ubiquitous OSS components can be vulnerable.
- Lesson 2: Use of open source software can create a litigation risks across a wide range of areas, including cybersecurity and regulatory compliance.

# Litigation Risk Example – SCO Group (2003) (License Risk, Code Risk)

- SCO Group claimed to own some rights in Unix System V.
- Alleged that Unix licensees had impermissibly incorporated Unix code (ranging from “hundreds of lines” to over a million lines) into Linux (over 30 million lines of code) and other open source software.
- In 2003, sued Unix licensee for violation of Unix license, copyright infringement and trade secret misappropriation, seeking license termination and damages - and also sued Linux customers for copyright infringement.
- Unix licensee counter-sued, alleging that SCO suit was violation of GPL (due to SCO’s own distribution of Linux and incorporation of Linux features into SCO UNIX distribution).
- Lesson 1: Users of open source software are exposed to copyright infringement claims if a contribution to the open source software was infringing.
- Lesson 2: Before releasing software to the open source community, a company needs to confirm it has the legal rights to do so.

# Common Open-Source Licenses

Top Licenses (color indicates potential risk)

- MIT (32% of open-source projects)
- GPL General Public License v2.0 (18%)
- Apache 2.0 (14%)
- GPL General Public License v3.0 (7%)
- BSD (Berkeley Software Distribution) 2.0 (6%)
- Artistic License (Perl) (4%)
- LGPL (Lesser/Library GPL) – v2.1 (4%)
- LGPL (Lesser/Library GPL) – v3.0 (2%)

# Open Source Litigation

- Three categories of plaintiffs:
  - **Rights holders:** Parties that own or have rights to copyrighted software subject to an open source license
  - **Non-rights holders:** Parties without rights to software but claim to have been injured based on noncompliance with an open source license (*e.g.*, right to repair advocates)
  - **Other third parties:** Anyone claiming to be harmed from the use of open source software in violation of a statutory, regulatory, or other legal requirement
- Key litigation issues:
  - Standing
  - Copyright preemption
  - Covenants vs. Conditions
  - Damages / Injunctive Relief / Specific Performance

# Standing for Rights Holders

- A party may only bring suit for copyright infringement after the Copyright Office “has registered a copyright after examining a properly filed application.”
  - *Fourth Estate Pub. Benefit Corp. v. Wall-Street.com, LLC*, 139 S. Ct. 881, 892 (2019).
- The average processing time for all copyright applications is 3.6 months.
  - Online applications with an online deposit are nearly three times faster than using a mail submission.
- The copyright owner may recover damages for infringement occurring before registration, so long as it falls within the statute of limitations.



# Standing for Non-Rights Holders

- Non-rights holders would need to establish third-party beneficiary status under the open source license.
- To be a third-party beneficiary under California law, a non-rights holder must show:
  1. The party “would in fact benefit from the contract,”
  2. “A motivating purpose of the contracting parties was to provide a benefit to the third party,” and
  3. “Permitting a third party to bring its own breach of contract action against a contracting party is consistent with the objectives of the contract and the reasonable expectations of the contracting parties.”

*Goonewardene v. ADP, LLC*, 6 Cal. 5th 817, 829-830 (2019)

# Standing for Non-Rights Holders (cont'd)

- No case has held that a non-rights holder qualifies as a third-party beneficiary under an open source license.
- *Software Freedom Conservancy v. Vizio, Inc.*, 2022 WL 1527518 (C.D. Cal. May 13, 2022):
  - SFC alleges that Vizio breached GPLv2 and LGPLv2.1 by not distributing source code for dozens of programs that are part of its SmartCast software.
  - SFC seeks to obtain the source code for these programs for varied reasons including right to repair, consumer protection, and improved accessibility.
  - After removal, the Court remanded the case to state court without deciding the issue of standing.
  - “Whether SFC can successfully show it is a third-party beneficiary of the GPL Agreements is a question of state law that is not before this Court.”

# Copyright Preemption

- The Copyright Act preempts breach of contract and other state law claims that come within the subject matter of the copyright at issue and are equivalent to any of the exclusive rights under the Copyright Act.
- To survive preemption, a breach of contract claim needs to have an “extra element” that takes the claim outside the realm of copyright.
  - *Del Madera Props. v. Rhodes & Gardner, Inc.*, 820 F.2d 973, 977 (9th Cir. 1984).
- In general, a claim of copyright infringement requires: (1) ownership of a valid copyright, and (2) copying of constituent elements of the work that are original.
  - *Range Road Music, Inc. v. East Coast Foods, Inc.*, 668 F.3d 1148, 1153 (9th Cir. 2012).

# Copyright Preemption (cont'd)

- **Preempted:** Claim for breach of an open source license “allege[d] violations of the exact same exclusive federal rights protected by Section 106 of the Copyright Act, the exclusive right to reproduce, distribute and make derivative copies.”
  - *Jacobsen v. Katzer*, 609 F. Supp. 2d 925, 933 (N.D. Cal. 2009)
- **Not Preempted:** “The ‘viral’ component of the GPL is separate and distinct from any copyright obligation. ... An affirmative promise to make its derivative work open source because it incorporated an open source program into its software” provided an “extra element.”
  - *Versata Software, Inc. v. Ameriprise Fin., Inc.*, 2014 WL 950065, at \*4-5 (W.D. Tex. Mar. 11, 2014)
  - *Artifex Software, Inc. v. Hancorn, Inc.*, 2017 WL 1477373, at \*3-4 (N.D. Cal. Apr. 25, 2017)
- **Not Preempted:** “There is an extra element to SFC’s claims because SFC is asserting, as a third-party beneficiary of the GPL Agreements, that it is entitled to *receive* source code under the terms of those agreements.”
  - *Software Freedom Conservancy v. Vizio, Inc.*, C.D. Cal. Case No. 8:21-cv-01943-JLS-KES (Dkt. 30)

# Conditions vs. Covenants

- If a copyright licensee acts outside the scope of the license, the licensor may sue for copyright infringement.
  - *MDY Indus., LLC v. Blizzard Entm't, Inc.*, 629 F.3d 928, 939 (9th Cir. 2010)
- Not all breaches of a copyright license give rise a copyright infringement claim.
- “To recover for copyright infringement based on breach of a license agreement, (1) the copying must exceed the scope of the defendant’s license and (2) the copyright owner’s complaint must be grounded in an exclusive right of copyright (*e.g.*, unlawful reproduction or distribution).”
- **Conditions:** “contractual terms that limit a license’s scope, the breach of which constitute copyright infringement.”
- **Covenants:** all other license terms, actionable under contract law.

# Conditions vs. Covenants (cont'd)

## Examples of conditions:

- Right to copy, modify, and distribute open source software “provided that” the user include a notice stating how the software was changed and comply with other restrictions on distribution of its modifications.
  - *Jacobsen v. Katzer*, 535 F.3d 1373, 1380-81 (Fed. Cir. 2008)
- No creation of derivative works without the licensor’s consent.
  - *MDY Indus., LLC v. Blizzard Entm’t, Inc.*, 629 F.3d 928, 940 (9th Cir. 2010)

## Examples of covenants:

- Open source disclosure obligations under the GPLv2.
  - *Vizio, Inc.*, 2022 WL 1527518 at \*4.
- Prohibitions against bots; no disrupting another player’s gaming experience.
  - *MDY*, cited above.

# Damages

The lack of money changing hands in open source licensing should not be presumed to mean that there is no economic consideration, however. There are substantial benefits, including economic benefits, to the creation and distribution of copyrighted works under public licenses that range far beyond traditional license royalties. For example, program creators may generate market share for their programs by providing certain components free of charge. Similarly, a programmer or company may increase its national or international reputation by incubating open source projects. Improvement to a product can come rapidly and free of charge from an expert not even known to the copyright holder.

*Jacobsen v. Katzer*, 535 F.3d 1373, 1380-81 (Fed. Cir. 2008)

# Damages (cont'd)

- A court has awarded statutory damages for infringement of open source copyrights (albeit in the context of default).
- Categories of contract damages could include (at least under California law):
  - Reasonably royalty (including the use of similar, commercial licenses)
  - Unjust enrichment
  - Disgorgement of profits
    - *Artifex Software, Inc. v. Hancom, Inc.*, 2017 WL 4005508, at \*4-5 (N.D. Cal. Sept. 12, 2017)



# Injunctive Relief

- *Jacobsen v. Katzer* is the leading case on injunctive relief in open source litigation.
- In a pre-*eBay* decision, the Federal Circuit reversed the denial of a preliminary injunction, ruling that the open source copyright holder may be able to demonstrate irreparable harm or otherwise have it presumed.
- On remand, the district court held that intervening rulings required the copyright holder to show irreparable harm.
- “The Federal Circuit court’s list of potential harms that a copyright holder may face in the open source field are just that—*potential* harms. There is no showing on the record before this Court that Jacobsen has actually suffered any of these potential harms.”
  - 609 F. Supp. 2d 925, 937.

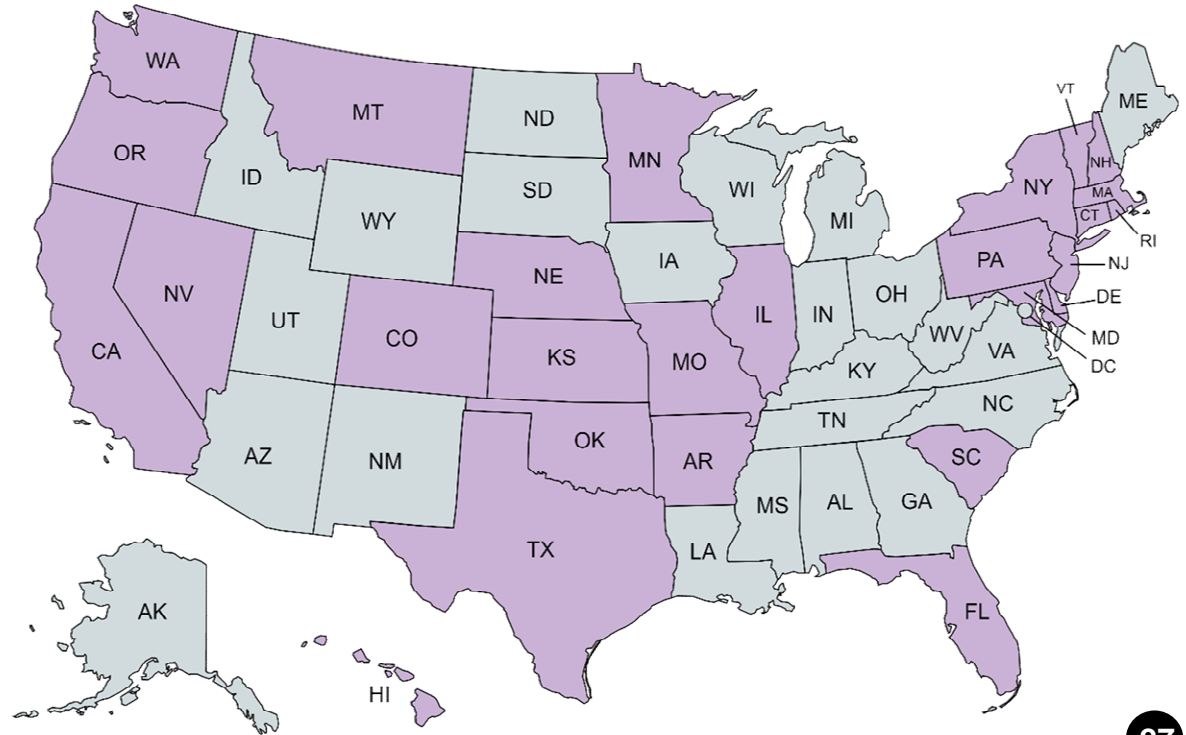
# Specific Performance

- In California, the elements of specific performance are well established:
  1. Inadequacy of legal remedy
  2. Underlying contract that is reasonable and supported by adequate consideration
  3. Existence of a mutuality of remedies
  4. Sufficiently definite contract terms for the court to enforce
  5. Substantial similarity of the requested performance to that promised in the contract

*Tamarind Lithography Workshop, Inc. v. Sanders* (1983) Cal.App.3d 571, 575.

# State Right to Repair Laws

- At least 27 states have introduced right to repair laws in their legislatures.



# California's Proposed Right to Repair (SB 983)

- SB 983, as currently introduced, would require that manufacturers of electronic devices and appliances make available to product owners and service and repair providers:
- For products between \$50 to \$100:
  - “sufficient service literature, **at no charge**, and functional parts and tools, inclusive of any updates, **on fair and reasonable terms**, to effect the diagnosis, maintenance, or repair of a product **for at least three years**” after manufacture AND
  - “**on fair and reasonable terms**, any documentation, tools, software and parts needed to disable the lock or function, and to reset the lock or function when disabled during the course of the inspection, diagnosis, maintenance, or repair of a product for **at least three years**” after manufacture
- For products above \$100, these obligations extend for **at seven years** after a product was manufactured.
- The bill does not require a manufacturer “to divulge a trade secret, except as may be necessary to provide service literature, documentation, tools, software, and parts on fair and reasonable terms.”
- As amended, the bill also *“shall not be construed to require the distribution of a product’s source code.”*
- The bill has not yet made it out of committee.

# DMCA: New Exemptions to Anticircumvention Measures (17 U.S.C. § 1201)

- Section 1201(a)(1)(A) of the DMCA prevents circumvention of technological measures that control access to a copyrighted work.
- DMCA authorizes the Copyright Office to determine exemptions to this restriction on a 3-year basis through rulemaking.
- In October 2021, the Copyright Office published a Final Rule on renewed and new exemptions (<https://public-inspection.federalregister.gov/2021-23311.pdf>).
- The Final Rule included new and expanded exemptions for right to repair advocates in several computer program classes, including: (10) computer unlocking, (11) jailbreaking, (12) repair, and (13) security research.
- The Final Rule also included a new exemption for investigation of open source license violations.

# **New Open Source Exemption: 37 C.F.R. § 201.40(b)(20)**

(20) Computer programs, solely for the purpose of investigating a potential infringement of free and open source computer programs where:

- (i) The circumvention is undertaken on a lawfully acquired device or machine other than a video game console, on which the computer program operates;
- (ii) The circumvention is performed by, or at the direction of, a party that has a good-faith, reasonable belief in the need for the investigation and has standing to bring a breach of license or copyright infringement claim;
- (iii) Such circumvention does not constitute a violation of applicable law; and
- (iv) The copy of the computer program, or the device or machine on which it operates, is not used or maintained in a manner that facilitates copyright infringement.

# Executive Order on Promoting Competition in the American Economy

- President Biden issued a July 9, 2021 Executive Order “affirm[ing] that it is the policy of my Administration to enforce the antitrust laws to combat the excessive concentration of industry, the abuses of market power, and the harmful effects of monopoly and monopsony ....”
- Among other markets, the Executive Order identified “repair markets.”
- Under Section 5(h)(ii), the Executive Order directed the FTC to consider exercising its rule making authority for “unfair anticompetitive restrictions on third-party repair or self-repair of items, such as the restrictions imposed by powerful manufacturers that prevent farmers from repairing their own equipment.”

# Best Practices for Using Open-Source Software

- Overall goal: promote safe use of OSS to leverage benefits and mitigate risks
- Establish an open-source policy and internal processes to implement it
  - Review and approve OSS use requests
  - Track use of open-source software
- Keep accurate records
- Involve legal and developer organizations
- Training program
- Limited scope of approval
- Different review tracks for different types of uses/licenses (e.g., strictly internal uses of unmodified OSS vs. OSS used in distributed code)
- Consider fast track approval process
  - Limited set of licenses
  - Limited set of uses
- Reevaluate if OSS use changes
- Audit OSS use (code inspection or tool, such as Black Duck)



# Open-Source Use Requests

Request to use OSS for company project should identify:

- OSS version
- Proposed OSS use:
  - Company product
  - Modified?
  - Internal use only?
  - Integrated with proprietary code? If so, how? (e.g., copy-paste, statically-linked, dynamically-linked, API call?)
  - Server only?
  - Distributed?
  - Part of Cloud/SaaS offering?
- Known vulnerabilities
- Applicable license
- Availability of same code under non-open license
- Strength of open-source community
- Internal champion

# Open-Source Use Guidelines

- **Generally safe:**
  - Using OSS under BSD or MIT licenses
  - Running company code on Linux OS
  - Using LGPL libraries without modification
  - Running OSS only on servers with no distribution (though beware AGPL and SSPL code)
  - Caveat (risky to integrate any OSS with proprietary code)
- **Be cautious:**
  - Developing non-GPL software that is compatible with functionality of GPL software (use “clean room” process – check GPL header files; may want to use an interface layer)
  - Calling an executable GPL program via an API (check header files)
- **Risky (Prohibit?):**
  - Allowing developers to use GPL source code
  - Accepting any third-party code for use in one of your software products without understanding where it came from, under what license
- **Check code dependencies**
  - OSS can incorporate other OSS
  - Good practice to check all licenses associated with dependencies

# Best Practices for Contributing to Open-Source Software

- If company uses OSS, will probably want to contribute code to open-source community
- Adopt Internal Review Process/Committee
- Open-Source Contribution Request:
  - Reasons for contributing?
  - What license will apply?
  - Is contribution subject to third-party encumbrances?
  - Does contribution use company patents?
  - Strength of open-source community?
  - Level of company commitment to OSS code in future?
  - Need two source code trees in future?
  - Harm to revenue?

# Open-Source Due Diligence

- Ask Company to identify:
  - Specific OSS items used by Company, including OSS license associated with each item and each item's dependencies
  - Context of each use. For example:
    - Admin tool
    - Run on company server as part of SaaS/Cloud offering
    - Distributed to end users/licensees
  - Extent of integration with proprietary code
  - OSS modifications
  - OSS contributions
- May want to request a commercial software composition scan to identify license conflicts
- May want to ask Company to document how it manages OSS, trains employees to safely use OSS, and addresses OSS vulnerabilities

# Practical Takeaways

- Be alert to possible OSS license violations or contamination in code that you distribute, use in a product, or use as part of a customer/end user service;
- When negotiating an inbound software license, ask licensor for: 1) list of copyleft OSS in the software, and 2) open-source indemnity to protect against misuse of the OSS by the licensor.
- When using a contractor to develop code, 1) review their policies, processes and training for working with OSS, 2) review and audit their use of copyleft OSS during the engagement, 3) ask for an open-source indemnity to protect against their misuse of OSS.
  - Lower risk: modifying server code or existing OSS
  - Higher risk: modifying company code or code that will be distributed or used in a product
- When negotiating an outbound software license, 1) resist providing a broad open-source indemnity, especially if you didn't develop the code.
- Refer all OSS questions to Legal.
- Refer to Legal any questions regarding internal use of open-source software.
- When distributing OSS subject to a copyleft license, Be sure to be prepared to disclose the code when requested and as required (e.g., GPL v. 2, Section 3).

**Morgan Lewis**

**QUESTIONS?**

# Biography



## **Douglas J. Crisman**

Silicon Valley

+1.650.843.7508

[douglas.crisman@morganlewis.com](mailto:douglas.crisman@morganlewis.com)

Douglas J. Crisman brings the perspective of a software designer and IP director for a leading computer hardware company to his patent law practice, which includes patent preparation, licensing, and prelitigation opinions, as well as IP transactions, due diligence, and counseling. He routinely works with standards-setting bodies and consortia on IP issues, and provides advice on strategic IP management and open-source legal issues ranging from software development to code review and licensing.

# Biography



## **Corey R. Houmand**

Silicon Valley

+1.650.843.7524

corey.houmand@morganlewis.com

Corey R. Houmand represents clients in litigation involving patents, trademarks, copyright, trade secrets, and related intellectual property matters such as unfair competition. Corey joined Morgan Lewis after serving as a law clerk for Judge Mary Beck Briscoe of the US Court of Appeals for the Tenth Circuit and Judge Samuel G. Wilson of the US District Court for the Western District of Virginia. In law school, Corey was articles editor for the *Wake Forest Law Review*.

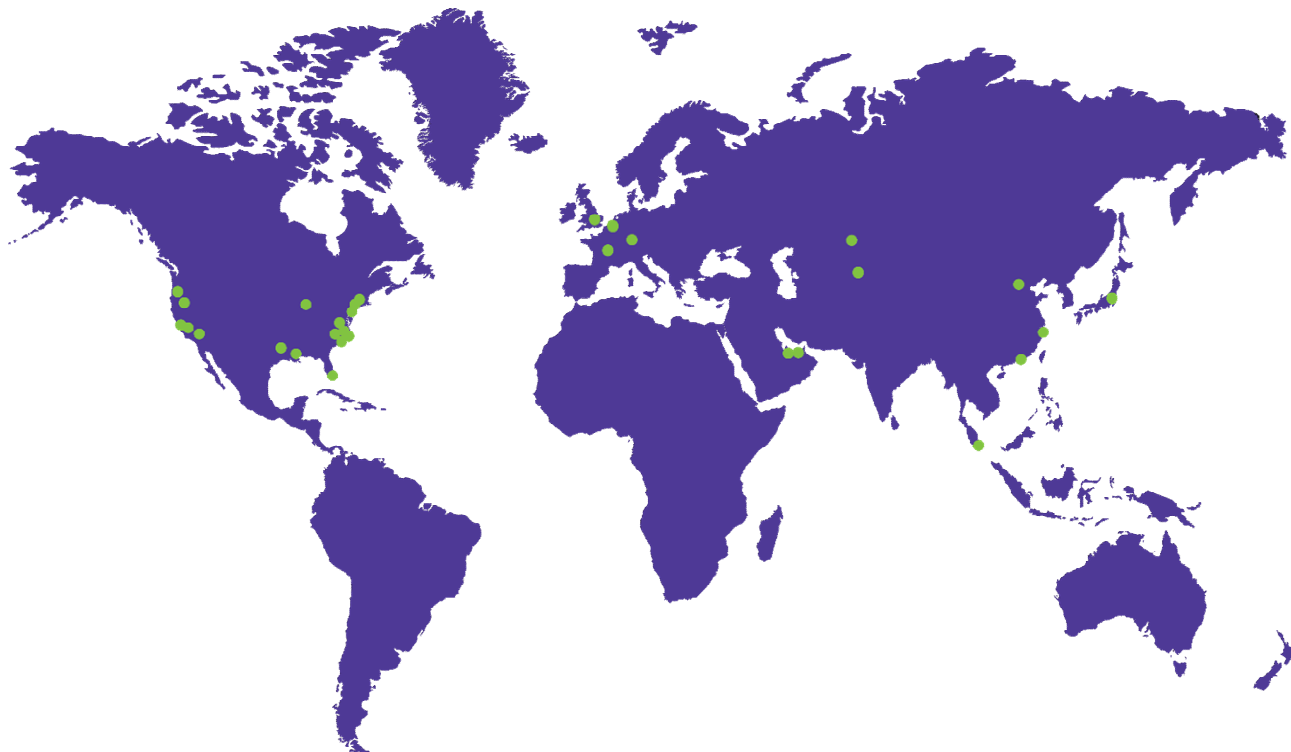


## Our Global Reach

Africa  
Asia Pacific  
Europe  
Latin America  
Middle East  
North America

## Our Locations

Abu Dhabi  
Almaty  
Beijing\*  
Boston  
Brussels  
Century City  
Chicago  
Dallas  
Dubai  
Frankfurt  
Hartford  
Hong Kong\*  
Houston  
London  
Los Angeles  
Miami  
New York  
Nur-Sultan  
Orange County  
Paris  
Philadelphia  
Pittsburgh  
Princeton  
San Francisco  
Shanghai\*  
Silicon Valley  
Singapore\*  
Tokyo  
Washington, DC  
Wilmington



# Morgan Lewis

Our Beijing and Shanghai offices operate as representative offices of Morgan, Lewis & Bockius LLP. In Hong Kong, Morgan, Lewis & Bockius is a separate Hong Kong general partnership registered with The Law Society of Hong Kong. Morgan Lewis Stamford LLC is a Singapore law corporation affiliated with Morgan, Lewis & Bockius LLP.

# THANK YOU

© 2022 Morgan, Lewis & Bockius LLP  
© 2022 Morgan Lewis Stamford LLC  
© 2022 Morgan, Lewis & Bockius UK LLP

Morgan, Lewis & Bockius UK LLP is a limited liability partnership registered in England and Wales under number OC378797 and is a law firm authorised and regulated by the Solicitors Regulation Authority. The SRA authorisation number is 615176.

Our Beijing and Shanghai offices operate as representative offices of Morgan, Lewis & Bockius LLP. In Hong Kong, Morgan, Lewis & Bockius is a separate Hong Kong general partnership registered with The Law Society of Hong Kong. Morgan Lewis Stamford LLC is a Singapore law corporation affiliated with Morgan, Lewis & Bockius LLP.

This material is provided for your convenience and does not constitute legal advice or create an attorney-client relationship. Prior results do not guarantee similar outcomes. Attorney Advertising.