Morgan Lewis

OPEN-SOURCE SOFTWARE: RISKS AND REWARDS

May 13, 2021

Douglas J. Crisman
Janice H. Logan, Ph.D.

Presenters

Douglas J. Crisman

Janice H. Logan, Ph.D.

Morgan Lewis

Agenda

- Open-Source Overview
- Open-Source Benefits
- Open-Source Risks
- Common Open-Source Licenses
- Case Study 1: Microsoft Hyper-V
- Case Study 2: Heartbleed
- Advanced Topics/Recent Developments
- Best Practices
- Appendix (Open-Source License Summaries)

History of Open-Source Software

- Originated in the "free software" movement in the early 80s which was a reaction to the proprietary software model
  - end user dependant on the developer for bug fixes, upgrade and general maintenance of the s/w
  - goal was to obtain and guarantee freedoms for software users – e.g., freedoms to run, study, copy, distribute, and modify the software (https://www.gnu.org/philosophy/free-sw.html)
- The GNU Project was launched in 1984 to develop a complete UNIX like operating system
  - GNU (guh-new) is a recursive acronym for GNU's Not Unix
  - released under the GNU GPL, a "copyleft" license drafted to ensure that free software remains free (not proprietary), including modifications
  - supported by Free Software Foundation (FSF)
- Richard Stallman
  - early leader in the free software movement (GNU operating system, GNU Emacs editor, GPL, etc.)
  - inspired by Xerox refusal to freely provide source code for MIT AI Lab laser printer in late 70s
  - "free" as in free speech; not as in free beer!
- Linus Torvalds
  - developed the Linux kernel and the combined GNU/Linux OS is now referred to as the Linux operating system
  - released under the GPL v.2

Free → Open

- Free software movement was generally viewed as anti-business
- Open-source movement was directed to getting buy-in from Corporate America
  - began when Netscape announced that it was considering sharing the source code of its browser in 1998
- OSS movement was organized into a non-profit corporation, the OSI (Open-Source Initiative)
  - http://www.opensource.org/
- OSI publishes the "Open Source Definition," which outlines distribution terms of open-source software, and list of approved Open-Source Licenses
  - https://opensource.org/osd
  - https://opensource.org/licenses/alphabetical

Open-Source 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)
- Non-negotiable
- As is
- Quirky
  - patent licenses
  - publicity conditions
  - use restrictions
- Gotchas
  - distribution trigger
  - code combination (entire work (GPL), or just files (MPL))
- Ambiguous
- Enforcement
  - political
  - public

Common Open-Source Licenses

Top Licenses

- 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%)

Common Open-Source Licenses

GPL

- Viral: If proprietary software combined/integrated with GPL code AND distributed, combined code could be subject to the GPL
  - Combining Source Code (NOT OK)
  - Static Linking (NOT OK)
  - Dynamic Linking (per GNU, NOT OK – see FAQ: https://www.gnu.org/licenses/gpl-faq.en.html#GPLStaticVsDynamic)
  - Library headers (?)
  - API calls (?)
- Distributing proprietary code with GPL code as two separate programs (OK – no combination)
- Running combined code of any type on a server (OK – no distribution)
- Supporters are true believers in *free software* – not likely to compromise when OSS misused by big corporations
  - http://www.gnu.org/philosophy/philosophy.html

Common Open-Source Licenses

- LGPL
  - OK only if used as intended: libraries called by proprietary code
  - But, modifications to LGPL code subject to GPL
- Apache
  - No need to share modified source code
  - Contributors grant license to necessary patent claims
  - Patent termination clause
    - If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed
- BSD and MIT
  - Completely open – little to no restrictions on what you can do with the source code
  - Older (4 clause) version of BSD includes a publicity clause

Common Open-Source Licenses

Resources

- Top 20 Licenses
  - https://www.synopsys.com/blogs/software-security/top-open-source-licenses/
- Enforcement
  - Free Software Foundation and Software Freedom Conservancy
  - https://www.gnu.org/licenses/gpl-violation.html
  - https://sfconservancy.org/copyleft-compliance/
  - GPL Compliance Lawsuit funded by SFC
    - 2015 VMware Suit – unlicensed use of Linux code in proprietary "vmkernel" (https://sfconservancy.org/news/2015/mar/05/vmware-lawsuit/)
    - 2019 Case dismissed (procedural grounds), but Vmware agreed to remove vmklinux from vSphere product (https://sfconservancy.org/news/2019/apr/02/vmware-no-appeal/)

Case Study 1 – Microsoft Hyper-V (2009)

- Linux driver code (GPL v.2) incorporated in Microsoft's 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. http://linux-network-plumber.blogspot.com/2009/07/congratulations-microsoft.html
- Result – Microsoft open-sourced its Hyper-V drivers:
  - "Nice. Microsoft has released the Hyper-V drivers as GPLv2." (Id.)
- Lesson 1: Training is important – Coder apparently had access to GPL code and then used it inappropriately.
- Lesson 2: Misuse of GPL can generate a firestorm in community – may be hard to avoid open sourcing proprietary code combined with GPL code.
- Lesson 3: Maybe developers of proprietary code should not have access to GPL code.

Case Study 2 – Heartbleed Bug (2014)

- 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 component was vulnerable.
- Lesson 2: OpenSSL community was transparent about the bug and released fix same day as discovered and announced (April 7).
- Lesson 3: Review level of support for key projects – in 2014, OpenSSL project, used by thousands of companies, had one developer and was earning no more than $2,000 in donations each year. https://www.theregister.com/2021/05/10/untangling_open_sources_sustainability_problem/

Advanced Topics/Recent Developments

- Trends in Open-Source Projects
- Patents and Open-Source Software
- Treatment of Open-Source License as a Contract
- New Licenses Targeted at Cloud Uses of OSS
- Right to Repair

Trends in Open-Source Projects

- Trending Open-Source Projects show wide interest in big data analytics, machine-based learning, cloud platforms, blockchain)
  - TensorFlow (Google – neuron-based machine learning library)
  - Hyperledger (Linux Foundation – modular tools to promote commercial applications of blockchain technology)
  - Open Stack (cloud operating system that allows vast compute, storage, and networking resources to be provisioned and controlled through user-friendly dashboard)
  - R programming language (popular open-source tools for data manipulation, calculation and graphical display)
  - Life sciences (extensive libraries of open-source tools available from institutions like Fred Hutchinson Cancer Research Center, but check licenses – may not allow commercial use under permissive licenses)

Advanced Topics

- Patents and Open-Source Software
  - Consider obtaining patents and trademarks related to functionality that will be contributed to an open-source project
  - This allows a standard to be created related to the project (since use of the trademark associated with the standard would not be licensed under an open-source license)
  - Allows non-licensed uses of the open-source project to be pursued as a patent infringement
- Treatment of Open-Source License as a Contract
  - In *Artifex Software, Inc. v. Hancom, Inc.*, (N.D. Cal. Apr. 25, 2017), the Court found that the plaintiff adequately plead a breach of contract claim based on alleged violation of terms of the GNU GPL (e.g., due to incorporation of GPL software in proprietary code by the defendant)
  - Court also found that the plaintiff's contract claim would not be preempted by its copyright infringement claims

Recent Developments

- New Licenses Targeted at Cloud Uses of OSS
  - Mongo DB – popular database software used on servers adopted new license in 2018 (Server Side Public License, or SSPL) – requires disclosure of source code of modified versions of the program if a licensee enables third parties to interact with functionality of the program via a network
  - Elasticsearch – popular search software used in cloud applications, recently transitioned from Apache 2.0 License to SSPL
    - Note: in response to Elasticsearch transition to SSPL, open-source fork of Elasticsearch ("OpenSearch") made available on GitHub in January 2021 under the Apache 2.0 license
  - Plausible Analytics – web analytics software transitioned in October 2020 from MIT License to AGPL v.3
  - Takeaway: be alert to license changes and evaluate risk associated with your particular uses of the affected software
- Right to Repair – similar to what Richard Stallman was seeking in 70s
  - Movement to enable owners to repair their own machines/equipment
  - Including right to modify source code
  - Laws have been discussed
  - Debate: Is it dangerous to allow owners to modify mission critical software? What about software security vulnerabilities exposed due to source code disclosure?
  - Stay tuned

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)

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, dyanamically-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)
- What's 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

- Adopt Internal Review Process/Committee
- Open-Source Contribution Request:
  - Reasons for contributing:
    - Improve functionality of strategic OSS
    - Promote wider use of company technology
    - Add customizations to open-source project
    - Outsource coding to OS community
    - Improve standing with OS community, press, customers
  - 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


APPENDIX
LICENSE SUMMARIES
Morgan Lewis

GNU GPL v.2

- The GPL is a strong copyleft open-source license
  - Any work that, in whole or in part, contains code licensed under the GPL is governed by the GPL
  - Freedom to modify and distribute stipulates that changes to the GPL code can only be made with notice of such changes and identification of who made the changes
  - The GPL must accompany copies of the program that are distributed, including the disclaimer of warranties with respect to the software
- Section 2 states that the licensee may modify a copy of the program thus forming a work based on the program
  - A "work based on the program" means either the program or any derivative work under copyright law: that is to say, a work containing the program or a portion of it, either verbatim or with modifications and/or translated into another language
  - If licensee combines the program with other code, is this a modification of the program?
    - According to the Free Software Foundation (FSF) any code combined with GPL code is covered by the GPL
    - The GPL suggests that linking of two independent works where one is covered by the GPL, results in the combined works being covered by the GPL ("tainted")

GNU GPL v.2 (cont'd)

- You cannot charge any fees for the license of rights under the GPL. However, you can charge for:
  - costs of distribution
  - any warranties that you offer
  - any services that you offer
- Types of Linking
  - Static linking: linked at compile time
  - Dynamic linking: linked at run time
- According to FSF, both static and dynamic linking result in the linked code being covered by the GPL
- General industry standard:
  - Static linking results in the linked work being covered by the GPL
  - Dynamic linking does not result in the linked work being covered by the GPL because the two works are physically and logically separate
  - Aggregating GPL code with proprietary code on a volume of a storage or distribution does not require the company to license the company's proprietary code to third parties free of charge
  - Distribution of GPL code separately from, but in connection with, the company's proprietary code does not subject the company's proprietary code to the GPL, and the company may charge a fee to obtain the proprietary code

Note that this issue, like most open-source issues, has not been decided by a court.

GNU GPL v.3 (cont'd)

Basic activities that do not trigger viral effects under GPL v.3:

- Internally running unmodified program
- Internally running program with your modifications
- Redistributing unmodified source code must comply with GPL v.3 requirements, but does not affect other code
- Outsourcing development and hosting – must be under your direction or control and exclusively for you
- Running GPL v.3 software in an ASP environment does not constitute "conveying" the software (meaning no requirement to provide ASP users with code of such software). But Section 13 of Affero GPL v.3, a GPL v.3 extension, does impose such a requirement. (GPL v.2 does not address the ASP issue.)

Affero GPL v.3

Extension of GPL v.3:

- Enables disclosure conditions of GPL v.3 to apply to software with which users interact through a network

  13. Remote Network Interaction; Use with the GNU General Public License. Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software.

- Inherits other aspects of the GPL v.3
- Takeaway: Use AGPL v.3 code with care if using it as part of a cloud service.

GNU LGPL v.2.1

- Created to address concerns of businesses regarding tainting of proprietary code; Less restrictive than the GPL.
- LGPL allows proprietary code to be used in connection with GPL code through the use of programming Libraries.
- If you use LGPL governed Libraries (without modification), but do not co-mingle the Library licensed under the LGPL with your proprietary code (either in source or though linking) prior to distribution, you need only meet the requirements of providing notice of the license for the Library and a copy of the source code for it.
- If you modify the Library, then: (1) the modification itself must be a Library; (2) you must notify users that you modified the Library and the dates of the changes; (3) you must license the modified work for free; (4) you may not make the modified Library dependent on any other code.
- If you co-mingle the code of the Library with your code, you must: (1) supply source code of everything; OR (2) use a shared library mechanism and provide the source code for the Library; OR (3) provide a written offer valid for at least 3 years to do (1); OR (4) offer the code for download if applicable.

GNU LGPL v.3

- Incorporates all of the terms of GPL v.3, except:
  - Excludes DRM-related provisions in GPL v.3
- Retains ability for libraries to be used and modified without contaminating other portions of combined works

BSD

- Less restrictive than the GPL
- Redistribution of source code of the licensed software, with or without modification, requires:
  - retention of applicable copyright notice
  - retention of conditions and disclaimer contained in license
- Redistribution of binary form of the licensed software, with or without modification, requires:
  - reproduction of the applicable copyright notice
  - reproduction of conditions and disclaimer contained in the license in the documentation and/or other materials provided with binary form
- No obligation to provide the source code
- Derivative works can be commercialized
- 4 Clause version vs. 3 Clause version (best – no publicity reqmt.)

Server Side Public License (SSPL)

- Not an OSI approved open-source license
- Used for Mongo DB and Elasticsearch
- Similar to AGPL v.3 but riskier when used as part of a cloud service:

  13. Offering the Program as a Service

  If you make the functionality of the Program or a modified version available to third parties as a service, you must make the Service Source Code available via network download to everyone at no charge, under the terms of this License. Making the functionality of the Program or modified version available to third parties as a service includes, without limitation, enabling third parties to interact with the functionality of the Program or modified version remotely through a computer network, offering a service the value of which entirely or primarily derives from the value of the Program or modified version, or offering a service that accomplishes for users the primary purpose of the Program or modified version.

  "Service Source Code" means the Corresponding Source for the Program or the modified version, and the Corresponding Source for all programs that you use to make the Program or modified version available as a service, including, without limitation, management software, user interfaces, application program interfaces, automation software, monitoring software, backup software, storage software and hosting software, all such that a user could run an instance of the service using the Service Source Code you make available.

https://www.mongodb.com/licensing/server-side-public-license

Apache v.2.0

- License:
  - Perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and Derivative Works in Source or Object form
  - Perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (unless assert patents) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license only applies to those patent claims licensable by a given Contributor that are necessarily infringed by such Contributor's Contribution alone or in combination with the Work
- Obligations:
  - Must provide a copy of the Apache License with all distributions of the Work or Derivative Works
  - All files you modify must carry prominent notice that the file has been modified
    - All Derivative Works that you distribute must retain all copyright, patent, and other attribution notices from the source form of the Work
    - If the Work contains a Notice file as part of its distribution, then any Derivative Works you distribute must include a readable copy of the attribution notices in such Notice file (excluding the notices that do not apply to any portion of the Derivative Works) in at least one of the following places:
      - within a NOTICE text file distributed as part of the Derivative Works;
      - within the Source form or documentation, if provided along with the Derivative Works; or,
      - within a display generated by the derivative Works, if and wherever such third-party notices normally appear
    - Modifications to the Work contributed back to the licensor are governed by the terms of the Apache license
      - Includes license to patent claims infringed by use of Contributors modifications
    - If add own terms to licenses, must indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability

Biography

Douglas J. Crisman
Silicon Valley
+1.650.843.7508
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

Janice H. Logan, Ph.D.
Washington, D.C.
+1.202.739.5234
janice.logan@morganlewis.com

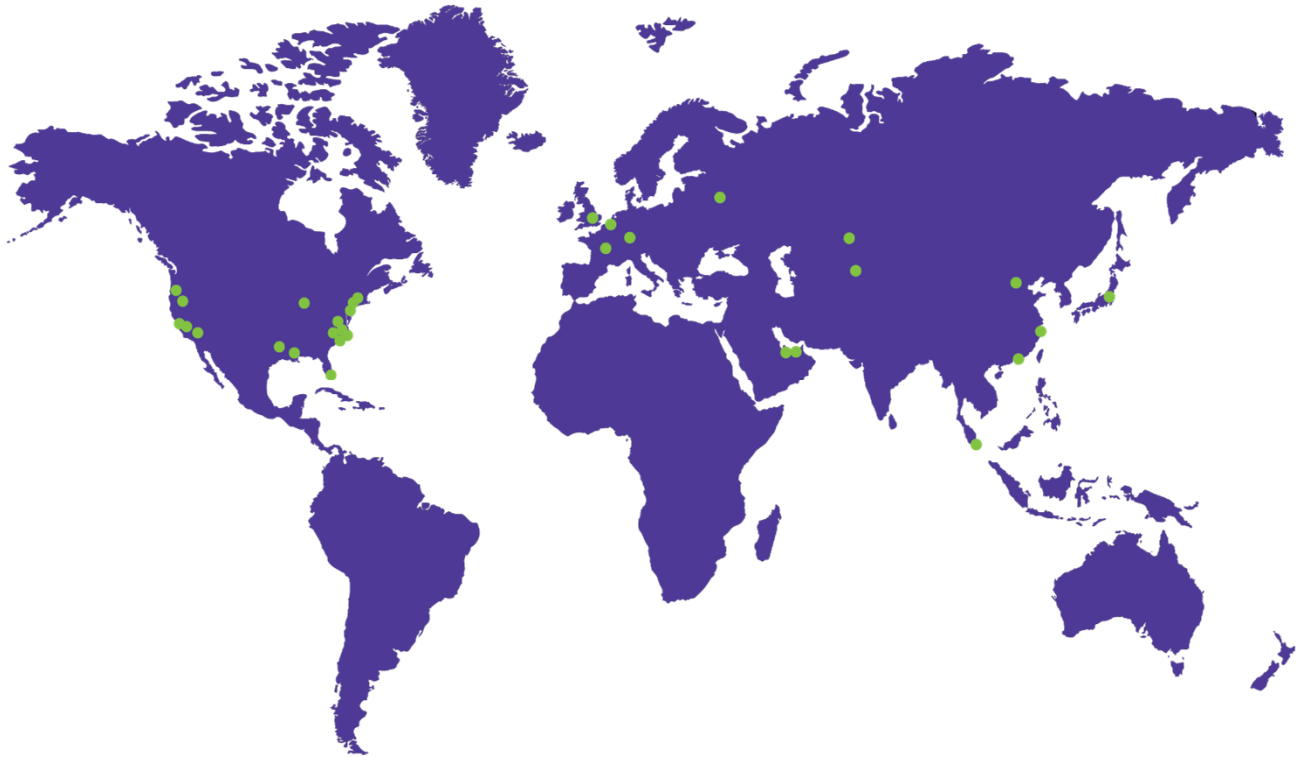
Janice (Lee) Logan brings a deep background in science and engineering to her IP law practice, focusing primarily on chemistry, biotechnology, and medical devices. Janice guides clients through complex patent procurement and patent litigation matters. She also manages due diligence for intellectual property asset transactions. In particular, Janice has years of experience developing and managing patent portfolios for large and startup companies utilizing artificial intelligence in biotechnology, including digital therapies and diagnostics. Janice is fluent in Korean and Japanese.

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
Moscow
New York
Nur-Sultan
Orange County
Paris
Philadelphia
Pittsburgh
Princeton
San Francisco
Shanghai*
Silicon Valley
Singapore*
Tokyo
Washington, DC
Wilmington



Morgan Lewis

THANK YOU