Morgan Lewis

THE IP ANATOMY OF THE AUTOMOTIVE NERVOUS SYSTEM

December 8, 2021

Douglas J. Crisman
Brent A. Hawkins

Presenters

Douglas J. Crisman
Partner, Silicon Valley

Brent A. Hawkins
Partner, San Francisco/Chicago

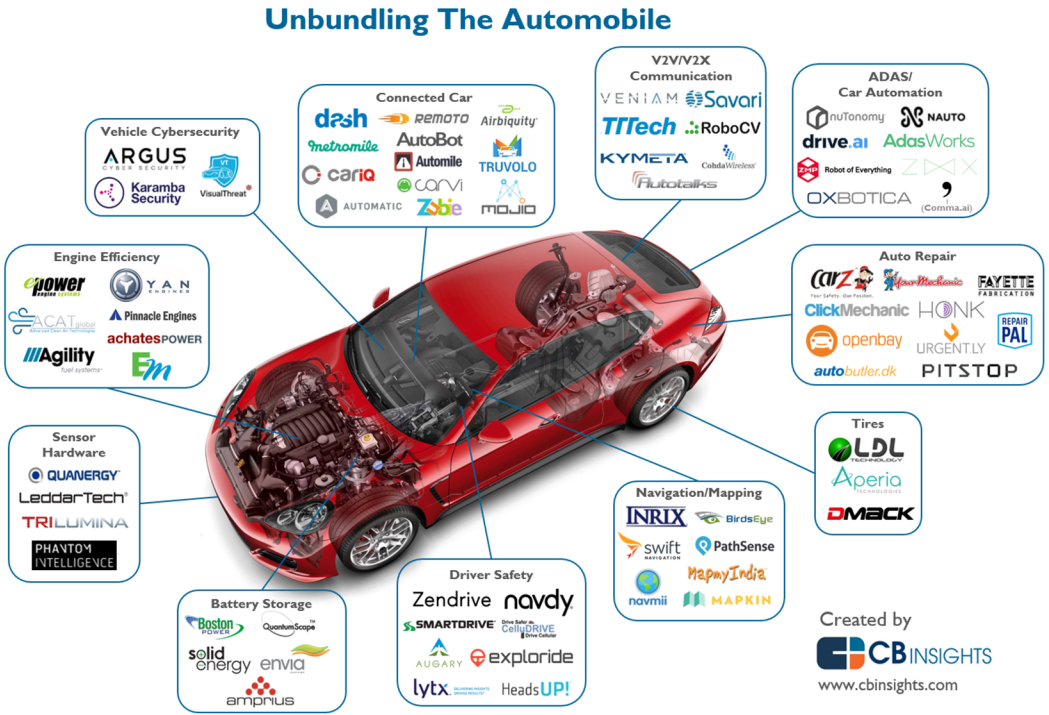
Morgan Lewis

Table of Contents

Automotive Nervous System

Morgan Lewis

The Automotive Nervous System



- Central Computer
- V2V/V2X Communication
- Infotainment
- Global Positioning System (GPS)
- Cameras (Video)
- Light Detection and Ranging (LIDAR)
- Sensors (Hardware/Software)
- Ultrasonic Sensors
- Aftermarket Repair
- Battery Storage

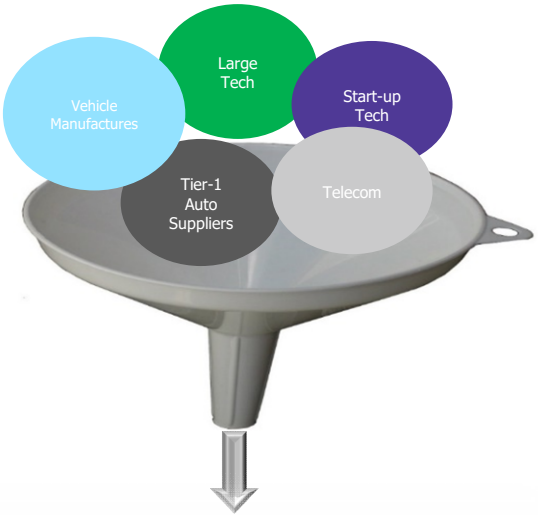
Source: https://cbi-blog.s3.amazonaws.com/blog/wp-content/uploads/2016/05/1-unbundling-car.png

Systemic Challenges

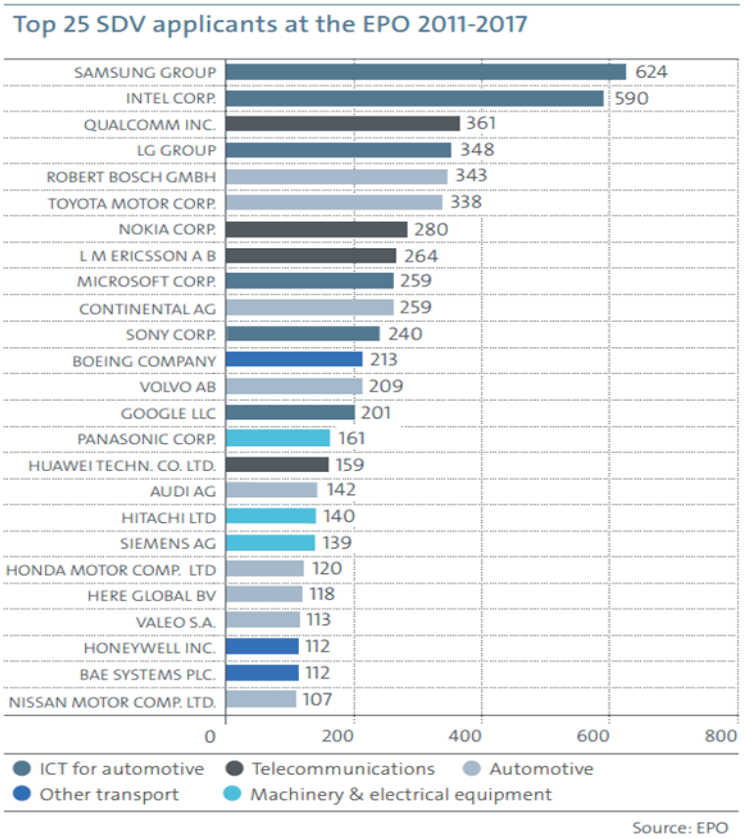
Standards

- Nearly two dozen standard setting organizations developing automobile interoperability standards and V2X communication (*e.g.*, V2V, V2I, V2G and V2N)
  - 4G, 5G, Wi-Fi (including Wi-Fi 6), Bluetooth, digital codecs, data encryption

Intellectual Property Landscape for Autonomous and Connected Vehicles



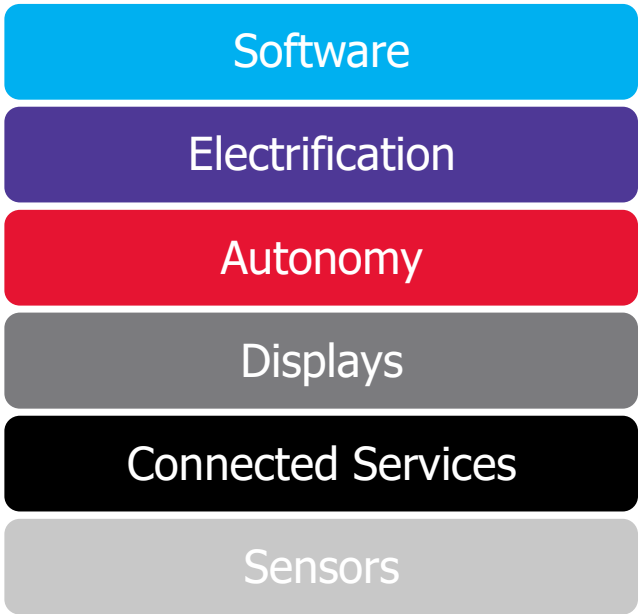
Disruption in Intellectual Property Landscape



Commodification/Conversion

- Similar subsystems, components, chips provided by OEMs and integrated by manufacturers
  - Batteries, power-train components (electrical, hybrid, ICE), displays, audio systems, sensors (radar), controllers/embedded processors, graphics chips, communication chips, interfaces, braking systems, charging infrastructure, passenger safety systems
- Platform sharing (within and between companies)
- Popular styling trends/uniform customer demand (E.g., SUVs, CUVs, no more sedans)
- New technology/laws leading to similar architectures, shapes, features
  - Skateboard EV architecture
  - Aerodynamics (slippery shapes)
  - Fuel economy (turbo 4-cylinder ICEs replacing 6s and 8s)

Deploying/Managing High Tech Features



Diminished Differentiation

- Conversion/commodification -> loss of differentiation
- Allowing Contractors/OEMs to own/develop differentiation -> IP Risk
  - Patent Assertion
  - Patent Defense
  - Tech Development
  - Participation in Standard Setting Organizations
  - Pay Royalties vs. Collect License Revenue
- Loss of differentiation -> lack of customer interest / exposure to price competition
- Different ownership models – does everyone need a car when they drive themselves and are expensive? -> Decreased market size for everyone?

Risks

Morgan Lewis

Connected and Electrification Technology Licensing

- Wireless players are seeking to license automotive OEMs
  - New licensing actions raise SEP issues
- Licensing of technology from software vendors to established players
- Energy and battery component manufacturers
- Patent pools active in the automotive space
  - Avanci, LLC offering first SEP patent pools on connected cars
  - Pools offer aggregated sets of patents for a single price
  - Generally, must be tied to a standard to pass antitrust scrutiny

Global Technology Disputes

- Global patent litigation
  - US District Courts
  - UK
  - Germany
  - China
- ITC investigations are more frequent and an effective tool
- Increasing trade secret litigation
  - Significant expansion of corporate entities in Silicon Valley
  - Litigation is fueled by employees rapidly moving to new companies

Data Privacy and Security

- Data collection and ownership
  - Conflicts of laws relating to content and volume of data that may be collected
    - Statutory and regulatory conflicts
    - Cross border conflicts
  - For what purposes can data be used?
    - Commercial
      - Over-the-air (OTA) software updates and maintenance analytics
      - Inventory management
    - Public safety exemptions

Data Privacy and Security

- Data security
  - Numerous initiatives directed to protecting inter-vehicle and intra-vehicle data
  - Compliance with international standards
    - UNECE WP.29 (World Forum for Harmonization of Vehicle Regulations)
    - ISO/SAE 21434 (The International Organization for Standardization)
  - Liability associated with storage and ownership
    - High number of access points increase potential security risk
      - Theft/break-ins
      - In-vehicle controls
      - Data/privacy breach

Open-Source Software

- Misunderstanding open-source development model
- Not all automotive applications suited to use of open-source software
- Code risk (code provenance)
- Lack of code support
- License risk
- Compliance risk
- Non-existent/outdated open-source policies
- Not adopting open-source software where appropriate

IP Strategies

Morgan Lewis

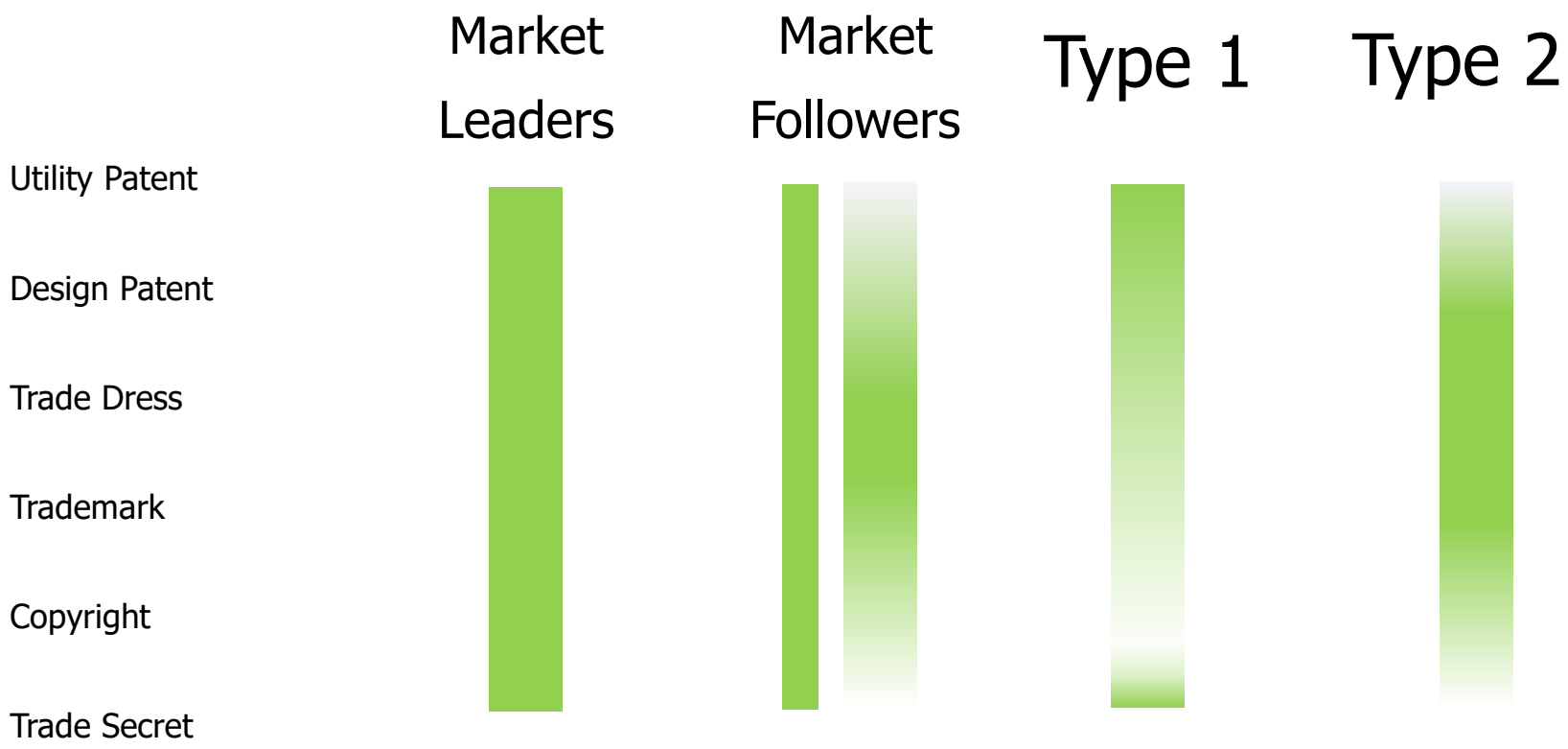
Automotive Market – Two Types of Players

- Market pioneers or leaders: first to market, integrate all aspects to offer unique user experience
  - Product category
  - Hardware feature
  - Software feature
  - Electrification
  - Autonomy
  - Interoperability (system integration)
  - Ecosystem architecture (V2X)
- Market followers: playing catch-up
  - Independent and strong innovation on one or more aspects
  - Reverse engineering or copying in some cases

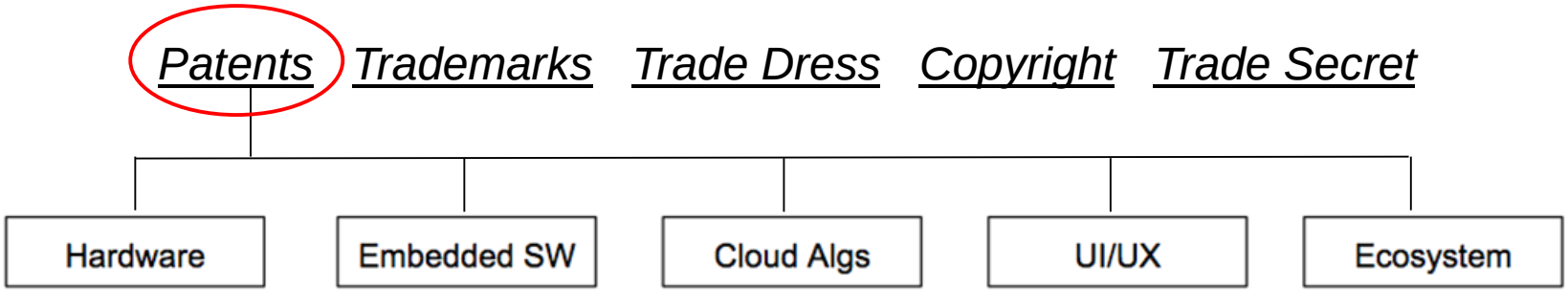
Automotive Market – Two Types of Competitors

- Type 1
  - Organically building competing product verticals
  - Innovating and integrating similar underlying technologies
  - Targeting similar consumer base
  - Interested in building brand recognition
- Type 2
  - Rebuilding successful products on a one-off basis
  - Reverse engineering and minimizing cost
  - Targeting entry-level consumer base
  - Leveraging third-party brand recognition rather than building organic brand
- Two questions determine your patent strategy:
  - What type of market player is your company?
  - What types of competitors are your company competing with?

Intellectual Property Protection Forms



Case Study



camera assemblies
lens modules
light rings
stands/mounts
controllable IR
radio coexistence
antenna designs
packaging
sensor tech

client-side zoom
USB communication
radio coexistence
mesh networking
API access
security/authentication
lens distortion
audio processing
adaptive bitrate
video/audio encoding

event categorization
event detection
false alarm suppress
person recognition
object recognition
zone definition/alert
zone search
smart timelapse
video backfill
3D imaging
3D object recognition
repurposing IR tx
day/night switching

zone definition/alert
automatic zones
user edited zones
timeline
timeline update
event notification
video history
video clips
cropping/gesture
smart timelapse
contextual UI

device registration
BLE provisioning
remote control
remote monitoring
parameters/control/status
sw update
mesh networking
cross-device operation
3rd party integration
- authentication
- pairing
- data/control sharing

Intentional Patenting

- With rise in software, electrification, autonomy, other high tech features, reconsider patent strategy – more like a consumer product than a traditional car.
- Consider all product technology layers – from chips to components to communications to cloud support to UI/UX – and their interactions.
- Patent technology that is core and differentiating.
- Patent technology that impacts the user experience (infotainment, displays, services, convenience, safety, charging, autonomy, v2x).
- File patents in your key markets and your competitors' key markets.
- Consider what to patent in view of the technology and your position in market vis a vis your competitors.
- If you put it in your ads, consider patenting it.

Strategies for Responding to SEP Assertions

- Categorize assertion entity
  - SSO/SDO participant
  - NPE aggregator
  - Licensing agent
- Understand standard(s) covered by assertion entity and FRAND obligations
- Consider geographic impact of a potential injunction
- Conduct an economic stack assessment
- Respond to SEP assertion letters and trigger *Huawei* and *Sisvel* requirements

SEP Licensing Consideration

- Build infrastructure/networking patent portfolio
  - May provide cross-licensing negotiation leverage
- Consider public rates when negotiating to SEP licenses
- Maintain use case distinctions
- Evaluate scope of pass-through rights through existing license agreements
- Consider upstream/downstream indemnification obligations
- Consider possible ADR mechanism to resolve FRAND rate

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 Origins

- Precursor - 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 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

Best Practices for Using Open-Source Software

- Overall goal: promote safe use of OSS to leverage benefits and mitigate risks
- Establish 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
- Ensure that selected OSS works for the intended application
- 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?

Questions?

Brent A. Hawkins, San Francisco/Chicago
+1.415.442.1449 | brent.hawkins@morganlewis.com

Douglas J. Crisman, Silicon Valley
+1.650.843.7508 | douglas.crisman@morganlewis.com

Coronavirus COVID-19 Resources

We have formed a multidisciplinary **Coronavirus/COVID-19 Task Force** to help guide clients through the broad scope of legal issues brought on by this public health challenge.

To help keep you on top of developments as they unfold, we also have launched a resource page on our website at **www.morganlewis.com/topics/coronavirus-covid-19**

If you would like to receive a daily digest of all new updates to the page, please visit the resource page to **subscribe** using the purple "Stay Up to Date" button.

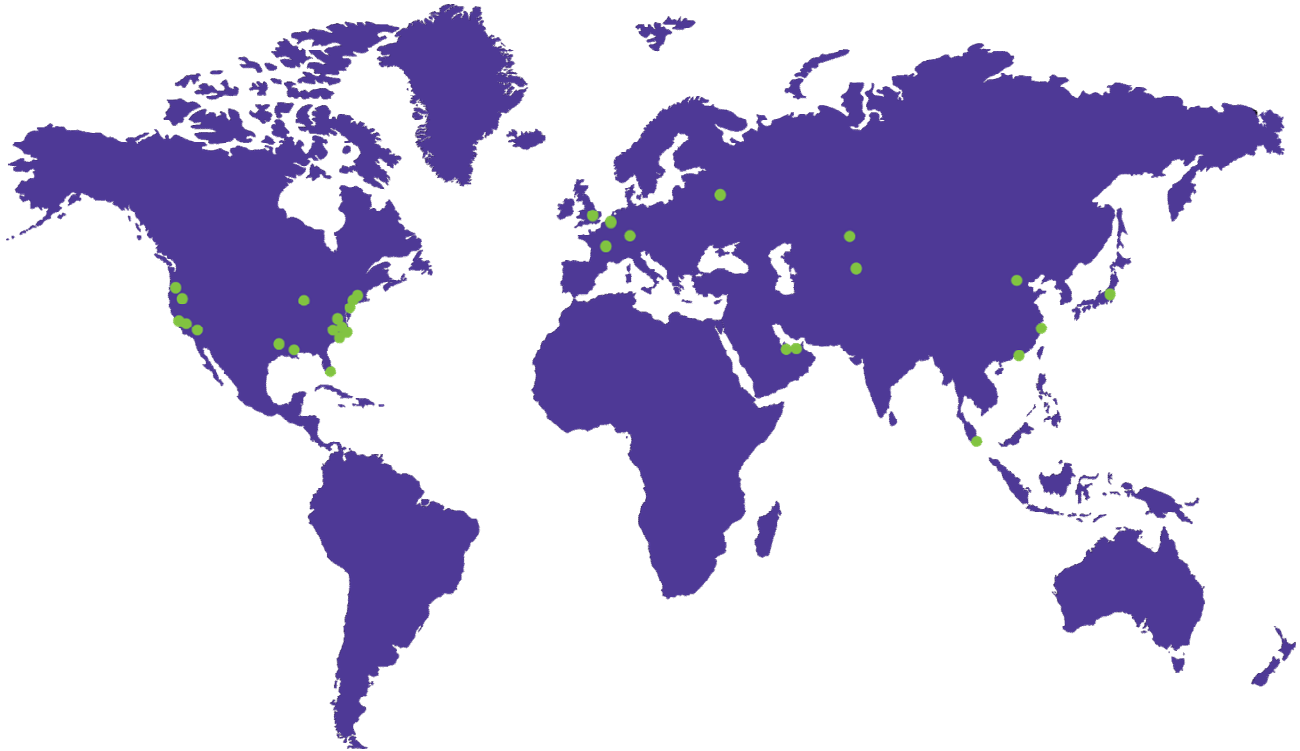
Morgan Lewis

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