

# AI VS. THE LAW

---

 JSConf 2025



The background image shows a serene marina at sunset. In the foreground, calm water reflects the warm orange and yellow hues of the sky. A long wooden pier extends from the left side of the frame towards the right, where several small boats are docked. On the far left, a large, multi-story white building with a prominent gabled roof and a central tower is visible, its windows glowing with warm light. The sky above is a dramatic canvas of orange, yellow, and blue, with wispy clouds catching the last light of the day.

**JSConf**

# Hi!

👋 I'm Ryan Roemer

N\_- I lead technology & OSS at Nearform

💻 I'm normally a geek...



[nearform.com/open-source](https://nearform.com/open-source)

... but today

⚖️ Former IP & patent attorney

 I was a MoFo lawyer → 

🔥 Keep an eye on code vs. law...

# Roadmap

© Copyright

→] Inputs

[→ Outputs

🚀 Future

# Admonitions & disclaimers

- IAAL but IANYL (and even there, IAAIL)
- Raise awareness, not provide answers or advice
- We'll talk about US law and computer code
- Talk to your own legal team

A wooden gavel and an open book are positioned on a light-colored surface against a dark background. The gavel is on the right, with its head resting on the sound block. The book is on the left, its pages fanned out. The lighting creates strong shadows, emphasizing the textures of the wood and paper.

# Law stuff

# Copyright law

A valid copyright protects the original (creative) expression of a work.

Provides exclusive rights to: reproduction, distribution, preparation of "derivative works"

# Fair use

A **defense** to copyright infringement. (17 U.S. Code § 107)

1. the purpose and character of the use
2. the nature of the copyrighted work
3. the amount and substantiality of the portion used
4. the effect of the use upon the market value of the copyrighted work

# Inputs



# Inputs

- What are you allowed to input into an AI system?
- How about for scraping?



# Training data

# Training data

What materials are you allowed to train on?

- Public domain works?
- Open source works?
- Private copyrighted works?
- Does how you collect the data matter?

# Training data

Why this matters:

- Which **models** and tools you will be allowed to use.
- How you will be able to **gather** and **use** data for your own ML/AI projects.

# Training data litigation

- *Thomson Reuters v. ROSS Intelligence* (D Del)
- *In re OpenAI Litigation* → OpenAI, MSFT.
- *Bartz v. Anthropic* (ND Cal) → Claude
- *Kadrey v. Meta Platforms* (ND Cal) → Llama

# Thomson Reuters v. ROSS (D Del)

Facts



Unauthorized metadata

Holdings



Infringement



No fair use

# In re OpenAI Litigation

## Parties



Plaintiffs: News  
companies, authors,  
etc.



Defendants: OpenAI,  
MSFT

## Status



Ongoing

# Bartz v. Anthropic (ND Cal)

Facts



Online sources



Scanned books



Pirated books

Holdings



Pirated = Infringing



Non-pirated + online = fair use



Settlement

# Kadrey v. Meta (ND Cal)

Facts



Permitted sources



Pirated books

Holdings



All fair use



Even pirated!

# Scraping



# Scraping

- *Reddit vs. Anthropic* (ND Cal) → non-copyright claims, licensing path
- CloudFlare's AI bot protections
- Really Simple Licensing\_(RSL): Reddit, Stack Overflow, Fastly, etc.



Inputs Takeaways

# Inputs: Takeaways

- The courts are split on fair use and this is all going to evolve.
- The cautious course is to gather data without pirating.
- You don't necessarily need permission, but we're seeing more paths and programs.

# Outputs

# Outputs

- What happens when AI produces copyrighted content/code?
- Can AI-generated content be protected by copyright?

A weathered metal sign is mounted on a chain-link fence. The top half of the sign is red with the word "DANGER" in white, bold, sans-serif capital letters. The bottom half is white with the words "KEEP OUT" in large, black, bold, sans-serif capital letters. The sign shows signs of age, including rust and peeling paint. The chain-link fence in front of it has some dried vines or branches entangled in its wires.

# Infringement

# Infringing AI outputs

- A big concern in the early days of GitHub Copilot.
  - *Doe(s) v. GitHub Inc.* (ND Cal)
  - Now has features to try and prevent.
- Separate tools for auditing and prevention. (Socket, Black Duck, etc.)

# Infringement: Takeaways

- You do need to be careful with AI-generated code outputs.
- But there are guardrails at various levels.



# Ownership

# Ownership of AI outputs

Is the code you create with AI assistance protected by copyright?

- If you use tab completions, inline chat?
- How about a full IDE chat or CLI tool?

# Ownership of AI outputs

Why this matters:

- You might care about copyright ownership (OSS protections, etc.)
- Your company probably cares about it.

# Thaler v. Perlmutter (D DC)

Facts



AI image tool



AI is the "author"

Holdings



No copyright



Only human  
authorship is  
protected

# US Copyright Office - 2nd report

- Purely AI-generated content is not protected
- Prompts & instructions alone are not sufficient
- Human control of "selection and placement" of the "creative elements" may be

A black and white photograph of a person's hands typing on a vintage typewriter. The typewriter is dark-colored with a visible keyboard and a carriage return lever. A sheet of paper is inserted, showing some handwritten text that appears to be encoded or ciphertext. The background is slightly blurred, focusing on the hands and the typewriter.

# A coding break

# High level

*Write a JS CLI tool to translate a package.json license field to a full license name.*

Full SPDX JSON data is available [on GitHub](#).

Detailed task description on [slide 40](#).

# Example 1

```
1 import { readFile } from "fs/promises";
2
3 async function findLicense(packageJsonPath) {
4   const packageJson = JSON.parse(await readFile(packageJsonPath, "utf8"));
5   const license = packageJson.license;
6   const response = await fetch(
7     "https://raw.githubusercontent.com/spdx/license-list-data/main/json/licenses.json",
8   );
9   const data = await response.json();
10  const found = data.licenses.find((l) => l.licenseId === license);
11  return found
12    ? `Found ${license}. Full name: ${found.name}`
13    : `Could not find ${license}.`;
14 }
15
16 const packageJsonPath = process.argv[2];
17 console.log(await findLicense(packageJsonPath));
```

# Example 2

```
1 import fs from "fs/promises";
2
3 async function findLicense() {
4   const packagePath = process.argv[2];
5   const license = JSON.parse(await fs.readFile(packagePath, "utf8"))?.license;
6   const licenses = await fetch(
7     "https://raw.githubusercontent.com/spdx/license-list-data/main/json/licenses.json",
8   ).then((res) => res.json())?.licenses;
9
10  const licenseName = licenses.find((l) => l.licenseId === license)?.name;
11  if (licenseName) {
12    console.log(`Found ${license}. Full name: ${licenseName}`);
13  } else {
14    console.log("Could not find license.");
15  }
16}
17
18 findLicense();
```

# Example 3

```
1 import { readFile } from "fs/promises";
2
3 async function findLicense(packagePath) {
4   const packageJson = JSON.parse(await readFile(packagePath, "utf8"));
5   const response = await fetch(
6     "https://raw.githubusercontent.com/spdx/license-list-data/main/json/licenses.json",
7   );
8   const spdxData = await response.json();
9   const license = packageJson.license;
10  const match = spdxData.licenses.find((l) => l.licenseId === license);
11  if (match) {
12    console.log(`Found ${license}. Full name: ${match.name}`);
13  } else {
14    console.log(`Could not find ${license}.`);
15  }
16}
17
18 findLicense(process.argv[2]);
```

**So why'd we do this?**

# Task description

- 1 Create a JS file "index.js" that:
  - 2 - Runs: `node /PATH/T0/index.js /PATH/T0/package.json`
  - 3 - Reads `package.json` license field
  - 4 - Fetches SPDX data from  
`<https://raw.githubusercontent.com/spdx/license-list-data/refs/heads/main/json/licenses.json>`
  - 5 - Matches license against `licenseId` in `licenses` array
  - 6 - Outputs: `Found {license}. Full name: {name}` or `Could not find {license}.`
- 7
- 8
- 9 Requirements:
  - 10 - ESM with `import`
  - 11 - No extra logging or error handling
  - 12 - Use `async/await`, `fetch`, `fs/promises`
  - 13 - One `findLicense` function (max 10 lines).
  - 14 - ALL code logic inside `findLicense` function
  - 15 - Call `findLicense` in script

# Human vs. AI code

- What are the differences between these?
- Extrapolate to larger pieces of code?

# Spec-driven development?

*We're moving from "code is the source of truth" to "intent is the source of truth." With AI the specification becomes the source of truth and determines what gets built. ☺*

---

```
1 // GENERATED FROM SPEC - DO NOT EDIT
2 // @generated with <TOOL> v0.22.1 from ./web-app/backend-api.spec.md
3 // (spec:b499c242) (code:ab3f180f)
```

# Ownership: Takeaways

- Collision between the lawyer and the engineering worlds
- The cautious approach is to favor as much human involvement as possible
- Assess if it's OK to have some of your code lack copyright protection



The future

# Just the beginning

- Open source vs. open weights
- AI beyond code (images, video, etc.)
- AI in other IP areas (patents, trademarks, etc.)

# This deck!

- ❑ See the deck at:

[ryan-roemer.github.io/ai-vs-the-law](https://ryan-roemer.github.io/ai-vs-the-law)

- 🐱 ...and check out the code at:

[ryan-roemer/ai-vs-the-law](https://github.com/ryan-roemer/ai-vs-the-law)



# Thanks!

-  [ryan-roemer](https://github.com/ryan-roemer)
-  [in/ryanroemer](https://www.linkedin.com/in/ryanroemer)
-  [@ryan-roemer.bsky.social](https://bsky.social/@ryan-roemer)
-  [ryan.roemer@nearform.com](mailto:ryan.roemer@nearform.com)

