Copyright and Licenses

In the AI era

So thank you very much for coming, and to the organisers for inviting me. This is a talk about copyright and licenses, which I know is nobody's favourite topic at a JavaScript meet-up. That being said, I do hope that at least this talk will be informative.

Alexandru Badiu

GeeIQ

Drupal Romania



My name is Alexandru Badiu, and I work as a software engineer at a company called GeeIQ. In a past life I was really involved in the OSS ecosystem, via the Drupal CMS - I was one of the earliest users, contributors and promoters of Drupal in Romania, where I'm from, I cofounded Drupal Romania, and did a lot of work around that NGO.

IANAL

I am not a lawyer

Status: October 2026

Now, before we start, I need to tell you about this acronym. It means "I am not a lawyer". Even if I were, this is a talk, and I would not be your lawyer. This is not legal advice. When in doubt, hire one.

And one more disclaimer that I didn't need eight years ago. Everything AI-related in here is true as of this month. Some of it changed in September. Some of it will change again before Christmas. Every slide with a court case on it has an expiry date.

Drupal → JavaScript

As a teenager I got into Linux (I am dating myself by saying this), And I'm one of the people who installed Slackware Linux using floppy disks and destroyed his MBR and his Windows ninety five installation. And then I spent a couple of years when I was in high school and early college trying to become a game developer so using the MS DOS and Windows versions of the GNU compiler and using libraries such as Allegro and SDL all of these incidentally being GPL.

Then I became a web dev, using Drupal, and then I moved to JavaScript. Drupal is also GPL. Pretty much everything around it is GPL. So you never really had to think about licenses.

Then I got into the JavaScript world and npm, and suddenly all this whimsical, novel code was one command away. Every `npm install` pulled in a few hundred packages, each with its own license. Some of them were more "legit" than others.

It was fun.

An email from the lawyers

Then we got an email from Corbis' lawyers. They had seen what we were doing and laid down the rules. Licenses X and Y are fine. Never touch Z. Anything else, ask us first.

(You can guess which one Z was. We'll get to it.)

That email sent me down a rabbit hole about copyright and licenses. I turned it into a talk and gave it at meetups and conferences for a couple of years. Then I took a break.

(cough AI cough)

A lot has changed since. So this is the updated version.

Who here shipped code an AI wrote this week?

Quick show of hands. Who here shipped code this week that an AI wrote, at least partially?

[Pause, report the result.]

And who checked the license of every line of it?

Right. Let's talk about that.

Copyright, quickly

Copyright

Automatic

Expression, not ideas

It expires (eventually)

So the basics first, and quickly.

The Berne Convention. You write a bit of text, some code, a drawing, and you get copyright instantly. No registration needed. Almost every country in the world has signed it.

It protects the expression of an idea, not the idea itself. So your sorting function is protected, sorting is not.

And it expires. Eventually. Very very eventually: in the EU and the US, it's the life of the author plus 70 years.

Why 70? Europe got there first. In 1993 the EU harmonised its copyright terms, and it picked the longest one, Germany's: life plus 70.

The US followed in 1998 with the Copyright Term Extension Act. Why? Officially, to match Europe. Unofficially, ask the mouse. It added 20 years to every copyright, including the ones that already existed. And Disney lobbied very hard for it, because Steamboat Willie, the first Mickey Mouse cartoon, from 1928, was about to go into the public domain. So people still call it the Mickey Mouse Protection Act. Somebody took it all the way to the Supreme Court, in Eldred v. Ashcroft, and lost in 2003.

Steamboat Willie finally became public domain on the 1st of January 2024. 95 years later. (The later versions of Mickey are still Disney's. And so is the trademark.)

And 95 is the number for your code too. Code you write at work, in the US, is the company's from day one, and it gets 95 years from publication. In most of Europe the company owns it, but you're still the author, so the clock is your life plus 70. MS-DOS, from 1981? Public domain in the US in 2077.

The first program ever published is public domain - Ada Lovelace's Note G (1843). Everything after that, give it until about 2060.

Who owns the code?

You.

Unless your contract says otherwise.

Who owns the code you write? You do.

Unless you're employed, and your contract says the company owns it, which it almost certainly does. Or you're a contractor, and your contract says the client owns it.

And that's assuming there's something to own. Not every piece of code is protected.

Take hello world in C. Can I copyright it? No. The bar for copyright is low, but it's there. In the US the Supreme Court called it a "modicum of creativity", in Feist, back in 1991. Europe says the code has to be "the author's own intellectual creation". Hello world has no creativity in it. There's basically one way to write it.

So where's the line? Nobody can tell you in lines of code. Realistically, it starts where you start making choices. How you structure things, how you break the problem down, names, the order of things. A clever 50 line function, probably protected. A three line getter, probably not.

Oracle sued Google over Android, and one of the things they fought about for years was a function called rangeCheck. Nine lines. Google did copy them. But the judge, who codes himself, told Oracle's lawyer in 2012 that he'd written code like that a hundred times. "It is so simple."

And one more thing that surprises people. Copyright protects against copying, not against similarity. If you and I write the exact same function, independently, we both own our copy. That matters later, with AI.

Remember this question. We'll come back to it, and the answer gets a lot less clear.

One more thing

The author has to be human

There's one more requirement, and it's one nobody thinks about. The author has to be a human.

I know what you're thinking. This is the part where he starts talking about AI.

Not yet.

The monkey selfie



Naruto v. Slater

In 2011, a wildlife photographer called David Slater was in Indonesia, photographing crested macaques. One of the monkeys grabbed his camera and took a bunch of selfies. And they were good selfies.

Someone uploaded one of them to Wikimedia Commons, the Wikipedia photo library, as public domain. Slater asked them to take it down, since it was his camera. Wikimedia said no: the monkey took the picture, and a monkey can't own a copyright. So nobody does.

The US Copyright Office agreed. In 2014 it updated its rules, and the list of things it won't register literally includes "a photograph taken by a monkey".

Then PETA sued Slater, on behalf of the monkey, claiming the monkey owned the copyright. They named him Naruto. The court said animals can't sue under the Copyright Act. The appeals court said the same thing in 2018, and was not very nice about PETA while doing it.

So: no human, no copyright. Remember the monkey.

A license

Permission

Without giving up the copyright

A license is a permission. You keep the copyright, and you tell other people what they're allowed to do with your code.

No license

No permission

What happens if there's no license? Then nobody has permission to do anything. You can look. You can't use, copy, or modify.

Even if it's public on GitHub. Public is not the same as open source. And no license is definitely not the same as public domain.

90% of public GitHub repos have no license

So how many repos on GitHub have no license? I checked a few days ago, using GitHub's own search.

About 90%. 324 million out of 362 million public repos.

When I first gave this talk, GitHub's own number was about 80%. It's gone the wrong way.

To be fair, GitHub only detects a LICENSE file in the root, so a license that only lives in your package.json doesn't count. But still.

---

81 million

New public repos in 2026

(58 million in all of 2025)

And this one is fun. In the first ten months of this year, 81 million new public repos were created on GitHub. In all of 2025, it was 58 million.

I can't prove why. But I have a guess, and it rhymes with "vibe coding".

Nine out of ten of those have no license either.

MIT

70% of licensed JavaScript repos

75% of licensed TypeScript repos

Among the repos that do have a license, in JavaScript, MIT wins. 70% for JavaScript, 75% for TypeScript. Across all of GitHub it's 60%, up from 45% in 2015.

So your node_modules folder is mostly MIT. Which matters later.

License families

Permissive

MIT BSD

ISC

Apache

Copyleft

GPL (2 and 3)

AGPL

LGPL

Two big families.

Permissive: MIT, BSD, ISC, Apache. Pretty much do what you want, keep the notice.

Copyleft: the GPL family. If you distribute something built on GPL code, the whole thing has to be GPL, and you have to give out the source.

GPL and JavaScript

A bundle in a browser is a copy

Here's the JavaScript bit that people miss.

The GPL triggers when you convey the code, when you give someone a copy. Code that only runs on your Node server and never leaves it? GPL version 3 says that's not conveying.

But your front-end bundle gets downloaded into every visitor's browser. That's a copy. So GPL code in the frontend is a different thing from GPL code on the backend.

And that's not just my reading. The FSF's own FAQ says GPL JavaScript is "distributed to the user when they visit the web site", and the source has to be released to them.

Has anyone been sued over the GPL at all? Yes. In Germany, in 2004, a netfilter developer, Harald Welte, got an injunction against a router maker. Two years later a Frankfurt court said the GPL is valid.

Same project, about ten years later, the opposite story. Another netfilter developer, Patrick McHardy, started sending GPL claims to companies and asking for money. People called him a copyright troll. The FSF and the Software Freedom Conservancy wrote down the rules: enforcement is about getting compliance, and must "never prioritize financial gain". And in 2016 the netfilter team suspended him.

For JavaScript, not that I could find. In 2014 the Software Freedom Law Center said GPL violations in JavaScript on the web were getting more common, and that disputes were "eminently foreseeable". Not being sued isn't the same as being compliant.

AGPL

Network use counts too

And then there's the AGPL, which closes the server loophole. If users interact with your AGPL code over a network, you have to offer them the source.

This is Z, by the way. From the lawyers' email. Never touch the AGPL.

(anyone remember the WTFPL?)

And then you had the fun ones. The WTFPL. I'll let you look up what it stands for. The Unlicense. Licenses written as a joke, or as an attempt to put code in the public domain, which doesn't really work in the EU because of moral rights.

Don't use them. Use MIT if that's what you mean.

Moral rights

The part of copyright you can't give away

Quick detour, because I just said "moral rights".

Copyright is two things. Economic rights: copy, distribute, sell. Those you can license, sell, give to your employer. And moral rights: the right to be named as the author, and to object to someone mangling your work. Those come from the Berne Convention, and in most of Europe you can't sell them or give them up. In France they're perpetual. Victor Hugo's descendant was in court over a Les Misérables sequel until 2007. (Hugo died in 1885.)

Germany goes further: you can't transfer copyright at all, only license it. So "I release this into the public domain" isn't something you can legally do there. That's why CC0 has a fallback: if the waiver doesn't work where you live, it becomes a very permissive license.

And the fun part: in the UK, moral rights don't apply to computer programs. It's in the act. The US only has them for visual art.

And code the model wrote? Moral rights belong to a human author. No human, no moral rights. (Remember the monkey.)

1984 → 2010

Sun RPC

One last story before AI, and it's my favourite.

Did you know that until 2010 every Linux distribution depended on code from 1984 that didn't have a free license?

It was Sun's RPC code, in glibc. It was written before free software licenses existed, and its license had restrictions that made it non-free. Debian noticed in 2002. Sun agreed to relicense it under BSD in 2009. Oracle, who had bought Sun by then, confirmed the last files in August 2010.

26 years. Get the license right from the start.

Can a machine own code?

So. AI.

A font

Laatz v. Zazzle

Let's start with a font.

Nicky Laatz is a type designer. She sued Zazzle over her fonts. She had registered them with the US Copyright Office as "font data", basically the code inside the font files.

The problem is that she drew the letters in FontLab, and FontLab generated the code. The examiner had asked her, more than once, whether the files were hand-coded by a human or generated by a font program. In March 2024 the court let Zazzle go ahead with its argument that she wasn't the author of that code. FontLab was. Her copyright claim was eventually dismissed.

This is a font editor. Not AI. A tool generated the code, so the code wasn't hers.

Now replace FontLab with Claude.

A human author

Thaler v. Perlmutter

In the US the big case is Thaler. Stephen Thaler tried to register a picture and listed his AI system as the author.

The appeals court in DC said no in March 2025. The Copyright Act requires a human author. The Supreme Court refused to hear it in March this year.

But note what it doesn't say. The same court said you can copyright work made with the assistance of AI. The author just has to be a human.

Prompts

"unlikely" to be enough

So how much human is enough?

The US Copyright Office wrote a report on this in January 2025. Their answer: AI used as a tool is fine, and AI output can be protected in whole or in part where a human determined the expressive elements. But prompts alone, at this stage, are unlikely to be enough.

What does protect you: your own code that's still in there, creative changes you make to the output, and how you select and arrange the pieces.

Europe

"the author's own intellectual creation"

Europe gets to the same place by another road. For software, EU law protects a program if it's "the author's own intellectual creation". The courts read that as a human making free and creative choices.

The lower courts agree so far. A court in Prague in 2023: an AI image isn't a work, the prompt is just an idea. A court in Munich in February this year: three AI-generated logos, no protection, because the prompting didn't shape the result in a way you could see.

The EU's top court hasn't ruled on AI yet. There's a case pending (Like Company v Google Ireland).

The UK

"computer-generated works"

The UK is the odd one out, at least on paper. Since 1988, its copyright law has a section for "computer-generated works" with no human author. The author is whoever made the arrangements.

In March this year the government said it wants to remove that. It hasn't done it yet. So for now, in the UK, the machine's output does get protected. Probably. Nobody has really tested it.

What's left?

Your edits

Your choices

Your structure

So if an agent wrote most of a file, what do you actually own?

The parts you wrote. The changes you made that are creative. How you put things together.

Copyright on code was always narrow, because what the code does isn't protected, only how you wrote it. With AI, the model did a lot of the writing. So there's even less that's yours.

Your LICENSE file

Covers what humans wrote

And here's the uncomfortable bit for open source.

A license is permission to use your copyright. If there's no copyright in a part of the code, there's nothing to license. Your MIT or GPL file still covers what humans wrote and arranged. On the purely generated parts, it might not have much to bite on.

This is my reading, not a court's. But it's why everyone, from the Software Freedom Conservancy to the FSF, says the same thing: keep track of what was generated. Because nobody else will.

Training

The other side of the coin. Not what comes out of the model, but what went in.

Copilot

Doe v. GitHub

You might remember the Copilot lawsuit from 2022. Developers sued GitHub, Microsoft and OpenAI, saying Copilot was trained on their code and spits it out without the license.

Most of it got dismissed over the years. The big remaining claim was under the DMCA, about removing copyright information.

The appeals court ruled last month, September 16. The DMCA claim is dead. The reasoning is interesting: Copilot creates new code, it doesn't take a copy of yours and strip the license off. You can't remove something from a thing you just created.

But two claims are still alive: breach of the open source licenses. Those are back in the lower court. So "does training on my MIT code break the MIT license" is still an open question, literally being argued right now.

Fair use

Bartz v. Anthropic

Kadrey v. Meta

In the US, the bigger fights are about books.

June 2025, two judges in San Francisco, two days apart. Both said that training a language model on books can be fair use. "Exceedingly transformative", one of them said.

But. Anthropic had downloaded pirated books, and the judge said that part was not fair use. Anthropic settled. For one and a half billion dollars. About 3,000 dollars per book.

And the Meta judge was clear: this doesn't mean training is legal. It means these particular plaintiffs made the wrong arguments.

Not fair

Thomson Reuters v. Ross

And then, a month ago, the first appeals court ruling on AI training. And it went the other way.

Ross built a legal search tool trained on Westlaw's headnotes. The appeals court said that's not fair use. A big part of the reasoning: there is now a market for licensing training data, and Ross took that market.

Important caveat: Ross isn't generative AI. It returned existing text. The court said so itself. So it doesn't decide the ChatGPT question. But it's a hint about where judges' heads are.

London vs Munich

Weights: copies or not?

Europe can't agree with itself either.

London, November 2025, Getty against Stability AI. The judge said the model weights don't contain the images. They're patterns, not copies.

Munich, also November 2025, GEMA against OpenAI. The court said ChatGPT memorized song lyrics, and memorizing them in the weights is a copy.

Both are under appeal. Different evidence, different laws, opposite answers.

---

robots.txt

EU AI Act, since August 2025

One thing that is settled, in Europe. Since August last year, anyone putting a general purpose model on the EU market has to respect machine-readable opt-outs from text and data mining, and publish a summary of what they trained on. Open weight models too. Fines started this August.

So your robots.txt is now a legal document. And a German court said an opt-out written in plain words in a website's terms of service didn't count. It has to be machine-readable.

GPL and the weights

Nobody says it attaches

So, does the GPL infect a model trained on GPL code?

No court has said so. The FSF hasn't said so. The OSI hasn't. The Software Freedom Conservancy calls it "undecided".

The output is different. If the model hands you a copy of GPL code, it's still that person's code, and the GPL is still your only license to use it.

Stack Overflow

CC BY-SA

A detour. Stack Overflow.

Every snippet on Stack Overflow is licensed CC BY-SA. Attribution, share alike. Copy an answer into your code, you owe attribution.

In December 2015 they announced that from 2016, code would be MIT. The community pushed back. In January 2016 it was "delayed indefinitely". It never happened.

In my old version of this talk I said they were "migrating to MIT". I was wrong.

"Steal Overflow"

Fast forward to May 2024. Stack Overflow signs a deal with OpenAI to give it access to its data for training. Some users deleted or defaced their answers in protest. One of them replaced their answer with "I have removed this answer in protest of Steal Overflow's decision to collude with OpenAI".

The answers got restored. Because the license you grant when you post is irrevocable. You can't take it back.

When the model copies

Back to the output. What happens when the model doesn't create something new, but hands you somebody else's code?

Q_rsqrt

0x5f3759df

Copilot, July 2021

Quick story. July 2021, Copilot is brand new. Armin Ronacher, the guy who wrote Flask, starts typing a function. Copilot hands him the fast inverse square root from Quake III. Line by line, comments included. One of those comments I can't read out loud here.

Quake III is GPL. And then Copilot autocompleted a license on top. BSD. (Wrong one.)

Why that function? Because it's everywhere. The more copies of something in the training data, the more likely the model remembers it word for word.

Oh, and it isn't Carmack's. It goes back to Greg Walsh, at a company called Ardent, in the late 80s. The humans got the attribution wrong first.

How often?

~1 in 10 weeks (2021)

<1% of suggestions (today)

Not often. GitHub studied this in 2021 with the first Copilot. One real copy every ten weeks of use, per developer. It happened most with popular code, and with an empty file.

Today GitHub says less than one percent of suggestions match public code.

Rare. Not zero. And researchers who go looking can pull a lot of memorized code out of these models.

Filters

≈150 characters

public GitHub code only

The tools have filters. Copilot can block suggestions that match public code, or show you where it came from and the license. Amazon and Google have similar features.

My favourite is Devin. If the code matches public code, Devin is told to "rewrite the implementation in its own words". Which is what we used to tell students who copied.

The model companies? Mostly not. I couldn't find one for ChatGPT, Codex, Claude Code, Mistral or DeepSeek. Cursor doesn't have one either. So if that's what you use, nothing is checking for you.

And when there is a filter, read the small print. They check against public GitHub code only. Around 150 characters and up. A clean result means "no exact match found". It doesn't mean "no license".

The license follows the code

MIT: keep the notice

GPL: the whole thing is GPL

And the license doesn't care how the code got into your editor.

If the model gives you a chunk of MIT code, you owe the notice. If it gives you a chunk of GPL code and you ship it in a bundle, the GPL applies to what you ship.

And if your tool shows you the license, you can't really say you didn't know anymore.

Indemnity

Paid plans only

Conditions apply

"But my AI vendor protects me." Maybe.

Microsoft, Google, Amazon, OpenAI and Anthropic all offer some protection if you get sued over output. On paid business plans. With conditions. Don't turn off the filters. Don't use output you should have known infringed.

And OpenAI and Anthropic exclude output that you modified or combined with other things. Which is, let's say, most code.

On the free and individual plans, and in Cursor, the risk is yours.

"if any"

"all our right, title and interest, if any"

And who owns the output? GitHub says it doesn't own it. OpenAI, Anthropic and Cursor assign you all their rights in it. And each of those adds two words: "if any".

They can't give you a copyright that the law doesn't recognize. And they can't clear somebody else's GPL for you.

"You own the output" doesn't mean "the output is clean".

Clean room

Photoshop, rebuilt in a month

chardet: LGPL → MIT

"I never saw the original"

Reverse engineering used to take a team and a year. Now it takes a month and an agent.

Last month someone released free, open source clones of Photoshop, Premiere and Lightroom, written with Claude. He calls it a clean room: Adobe's public file specs, and watching what Photoshop does. No Adobe code. Adobe hasn't done anything about it. Yet.

Is it legal? Mostly. In the US, reverse engineering for compatibility has been fair use since 1992, and Google v. Oracle said reimplementing an API can be too. Unless you clicked "I agree" on a license that says you won't. In the US, that clause wins. Blizzard won exactly that against a fan-made Battle.net server. In Europe you can study and test any program you're licensed to use, and the license can't stop you. Copying the code is still copying.

Games are greyer. Fan decompilations of Mario 64 and Zelda have been up for years. Last week Nintendo took down one of Rhythm Heaven.

In March, chardet's maintainer had Claude rewrite the whole library, and switched it from LGPL to MIT. The original author said he had no right to. The new code is 1.3% similar to the old one.

The catch: a clean room means you never saw the original. The model has seen all of GitHub. And in chardet's case, someone checked: Claude did open one of the old files along the way. No court has ruled on any of this.

Who's going to check?

"Nobody sees our code"

Until someone buys you

Until a customer asks

So here's the question I always get. I build commercial software. It's closed. Nobody is ever going to find that GPL snippet. Why should I care?

Because someone does look. When you raise money or get acquired, auditors scan your code. Black Duck is one of the companies that does these audits. Their 2026 report: 947 codebases, and 68% had license conflicts. And they scan snippets, not just your package.json. In their words: "A GPL violation discovered during M&A due diligence can derail an acquisition." (They do sell the audits.)

The standard US investment paperwork now asks founders to promise two things. Open source hasn't forced your code open. And nobody pasted your secrets into an AI tool that trains on them.

Then there are your customers. Your contract says you'll defend them if your product infringes someone's IP. That includes the bits the AI wrote.

And it's JavaScript. Your frontend ships to everyone. In March, Claude Code's own source leaked through its npm package. (Nobody sees our code.)

Anthropic sent takedowns saying "Anthropic is the copyright owner". Boris Cherny, who leads Claude Code, says Claude writes all of his code. Anthropic says more than 80% of the code it merges is written by Claude. (I'm sure the other 20% is the important part.)

Personal seats

Your code becomes training data

No indemnity

"Non-commercial use only"

And if your developers use consumer plans. Claude Pro, ChatGPT Plus, Copilot Free or Pro. Doesn't matter who pays, the company or the developer. If it's not a business plan, three things change.

Your code can become training data, unless someone turned it off. Claude since last October, Copilot since April, ChatGPT too.

No indemnity. That only exists on business plans.

And read the terms. In the UK, Anthropic's consumer terms literally say "Non-commercial use only".

Pay for the business seats. It's the cheapest part of all this.

Your open source project

A human signs off

Node.js: disclose it

Gentoo, NetBSD, QEMU: no

If you maintain something, someone will send you a pull request an agent wrote.

A lot of projects use the DCO, the Developer Certificate of Origin. When you sign off a commit, you certify you have the right to submit it. Everyone agrees on one thing: a human signs off, never the agent. The kernel tried an "Assisted-by" tag this year and dropped it this month.

Some projects say no outright. Gentoo, NetBSD, QEMU. In JavaScript it's mostly "yes, but". Node.js wrote its rules in August, after a 22,000 line pull request written with Claude Code: disclose it, understand it, test it.

So write your policy before that PR shows up.

In conclusion

Your code

Keep track of what was generated

If you write code with AI, and you do: keep track of what was generated. A commit trailer, a note, anything. You might need to know one day what's yours.

Your dependencies

The license follows the code

If the tool shows you a license, act on it. MIT, keep the notice. GPL in a front-end bundle, think twice. And remember that "you own the output" comes with "if any".

Your project

Write the policy

Before the PR arrives

If you maintain something, decide what you accept before the 22,000 line pull request shows up. And put a license on it. 90% of GitHub didn't.

Are we cooked?

Let the tools keep track

Scan what you ship

Pay for the business seats

So, are we cooked? No. But let's be realistic. I just told you to keep track of everything the AI wrote. Nobody is going to do it. Least of all an employee at five on a Friday.

So don't rely on people. Rely on the tools.

The agents already sign their work. Claude Code adds a Co-Authored-By line to every commit by default. Don't turn it off. A company can enforce that for everyone.

Turn on the filters. On Copilot Business, the admin can block suggestions that match public code for the whole company.

Scan what you ship. Checking your dependencies' licenses in CI is cheap: there are plugins for webpack and Rollup, and GitHub has an action that fails the pull request. Snippet scanning, the kind the auditors do, exists too. FOSSA, SCANOSS, Black Duck. Run it before the buyer does.

And pay for the business seats.

Who owns the code?

It's complicated.

Eight years ago, the answer to "who owns the code" was "you, unless your contract says otherwise".

Now it's: you, for the parts you wrote. Maybe nobody, for the parts you didn't. And maybe somebody else, if the model remembered their code.

Thank you

alexandru@badiu.me

https://alexandrubadiu.ro/articles/who-owns-the-code-footnotes/

Thank you. If you have questions, now is a good time, or find me afterwards.