TECH TIP — INTERMEDIATE

GitHub Privacy, Policy, and Ethics for Lawyers

What your bar rules actually require you to think about before you push

From Nolan Law Firm — nemolegal.com/tools

This Is Not Optional Reading

Lawyers have confidentiality obligations that apply to every tool they use — including GitHub. ABA Model Rule 1.6 and its state equivalents require competent handling of client information. ABA Formal Opinion 477 (2017) and Formal Opinion 512 (2024) establish that using cloud-based and AI tools requires understanding how those tools handle data. GitHub is a cloud tool. These rules apply.

> **Bottom line:** Before you push anything to GitHub, ask: does this file contain client information? If yes, it does not go on GitHub. Period.

What GitHub Does With Your Code

GitHub (owned by Microsoft) stores your repositories on their servers. For private repos, they do not share your code publicly. For public repos, everything is visible to the entire internet, indexed by search engines, and archived by third parties. Once something is public on GitHub, assume it is permanent — even if you delete it later, it may have been cached or forked.

Repo Type	Who Can See It	Key Risk
Public	Anyone on the internet	Secrets, client data, confidential info exposed permanently
Private	Only you and collaborators	Still on Microsoft servers — vendor risk applies
Organization	Members of the org	Access control depends on org settings

What Never Goes on GitHub

This list is not exhaustive. When in doubt, leave it out.

```
# NEVER commit these to any GitHub repo (public OR private):


Client names, case numbers, or matter details

Any document containing client information

API keys, passwords, or access tokens (see .env guide)
```

```
Database connection strings with credentials

Court filing content or discovery materials

Engagement letters or fee agreements with client data

Anything covered by attorney-client privilege

HIPAA-covered health information

Social Security numbers, dates of birth, financial data
```

The .gitignore Defense

The best protection is preventing accidental commits. A well-configured `.gitignore` file tells Git to never track certain files or folders. Set this up before your first commit.

```
# .gitignore for a law practice project:

.env

*.env

secrets/

client-data/

uploads/

*.docx # if Word docs might contain client info

*.pdf # if PDFs might contain client info

config_local.py

credentials.json

*.key

*.pem
```

GitHub's Terms of Service — What Lawyers Should Know

You own your code. GitHub's ToS states that you retain ownership of content you post. GitHub gets a license to host and display it, nothing more.

Public repos grant broad licenses to others. When you publish code publicly, others can view, fork, and reference it. If you want to control how others use your tools, add a LICENSE file. Without one, GitHub's defaults apply — which provide less protection than you might assume.

GitHub scans for secrets. GitHub runs secret scanning on public repos and will alert you (and sometimes the affected service) if it detects API keys or credentials. This is a safety net, not a reason to be careless.

Law enforcement requests. GitHub complies with valid legal process. Private repos are not immune to subpoenas. Do not store anything on GitHub that you wouldn't store with any other cloud provider subject to US law.

Choosing Public vs. Private — The Lawyer's Framework

What It Is	Public or Private?	Why
General automation scripts with no client data	Either	Safe either way; public builds your reputation
Document templates (blank, no client info)	Either	Consider public if you want to share with bar
Practice management integrations	Private	Config and structure may reveal client workflow
Intake or portal code	Private	Architecture reveals how client data flows
Anything with test data resembling real client data	Private until scrub first	Anonymize before committing
Tools you want to open-source for the community	Public + add LICENSE	Pick an appropriate license

If You Accidentally Push Sensitive Data

Step 1: Delete the file and push the deletion — but this alone is NOT enough. The file still exists in the commit history.

Step 2: Use git filter-repo (the current recommended tool) or BFG Repo Cleaner to scrub the file from the entire history. Then force-push.

Step 3: If the repo is public, assume the data was seen. Notify affected clients per your jurisdiction's breach notification requirements.

Step 4: Rotate every credential that was exposed immediately.

The Missouri Rules of Professional Conduct do not have a specific provision on GitHub, but Rules 1.1 (competence), 1.6 (confidentiality), and 5.3 (supervision of non-lawyers) all apply to how you manage technology in your practice. The ABA's Formal Opinion 512 (2024) is the current authoritative guidance on AI and cloud tools — its principles extend to GitHub.

> **Using AI to Help With This** You don't have to fully understand this to use it. Here are prompts that work:

```
"Review my .gitignore file for a law practice automation project and identify any categories of sensitive data I might be missing: [paste current .gitignore]."
```

```
"I want to open-source a document automation tool I built. What license should I use, what should I scrub before making it public, and what should the README say?"
```

```
"Draft a one-paragraph GitHub policy for my law firm covering what can be committed to public repos, what requires private repos, and what cannot go on GitHub at all."
```
