

A-06

Security and Access Policy

Least access, no production writes without written approval, AI providers only with consent, and deletion at the end.

Axdale. Software engineering by Michael Lip, Warsaw, Poland (CET).

Website <https://www.axdale.com>. Code portfolio on GitHub at <https://github.com/theluckystrike>.

Version 1.0. 23 October 2026.

This policy says how I handle your access, your code and your data, and it applies to every engagement under the Axdale Engagement Agreement. If this policy and the Agreement conflict, the Agreement applies. If your own security policy is stricter, send it to me before day 1. I follow it.

1. Principles

1. I ask for the least access that does the job.
2. I make no write to production without written approval for that specific action.
3. Secrets never travel in plain text.
4. AI model providers see your code only with your written consent.
5. When the work ends, I remove my access and delete your code and data.

2. Least privilege

1. Each Statement of Work lists every access I need, the access level, who grants it and when it ends.
2. I use a named personal account for each system. I never use a shared login.
3. The default is write access to branches only, with your main branch protected. I don't need admin rights on a repository, a cloud account or an organization.

4. The default environment is a test or staging environment with test data. I don't need production data.
5. Read-only access is the default for anything outside the repository, such as logs, dashboards or billing exports.
6. If I find that I have more access than the Statement of Work lists, I tell you in writing and ask you to reduce it.
7. You can remove any of my access at any time, with no notice.

3. Production

1. In the free first Week there are no production writes at all.
2. In a Paid Week, a production write needs written approval from your Decision Maker for that specific action. A production write includes a deploy, a database migration, a change to production data and a change to production configuration.
3. Each approval names the action, the time window and the rollback step.
4. I record each approved production action in the daily update, with the result.
5. If a production action goes wrong, I stop, roll back where the approval allows it, and tell you in writing at once.

4. Secrets

1. You share secrets only through your secret manager or a one-time encrypted link that expires.
2. Secrets never go in email, chat, issues, pull requests, code, commit history or AI prompts.
3. I store secrets I receive only in an encrypted password manager or in local environment files that git ignores.
4. I ask for test keys with spend caps where the provider supports them.
5. If a secret is exposed, I tell you within 24 hours and ask you to rotate it. I don't try to hide or rewrite the exposure.
6. At the end of the engagement I delete every secret you gave me, and I ask you to rotate any that I held.

5. AI model providers

1. AI model providers see your code or data only with your written consent. You give consent by ticking providers in the Schedule of the Agreement or in a Statement of Work.
2. If you tick no provider, no AI model provider sees your code or data. I still deliver the work.

3. My usual providers are Anthropic (Claude and Claude Code) and OpenAI (Codex and the OpenAI API). I use them only through paid accounts in my own name.
4. Where a provider offers a setting that stops it from using my inputs to train its models, I turn that setting off for every account I use on client work.
5. No other provider sees your code or data without your separate written consent. This includes free hosted models, other model vendors and AI features in third-party tools.
6. I never put secrets or personal data into an AI prompt.
7. I review every change that an AI tool helped to write before it reaches a pull request. I say in the pull request description when an AI tool helped.
8. Copies held by a provider follow that provider's own retention terms. I can't delete them. That's one more reason why providers see your code only with your consent.
9. If you withdraw consent for a provider, I stop using it on your work from that moment.

6. Personal data

1. I avoid personal data. Test data should be anonymized before I get it.
2. If the work needs personal data, you and I sign a data processing agreement before I get access. That agreement then controls how I handle the data.
3. I don't copy personal data to my devices unless the data processing agreement allows it.
4. Logs that the work creates must not contain email addresses, phone numbers or other direct identifiers. I add tests for this where the work writes logs.

7. Devices and accounts

1. I work on a computer with full disk encryption, an automatic screen lock and current security updates.
2. Every account that holds access to your systems uses multi-factor authentication where the system supports it.
3. I keep your work in a folder that is separate from other clients' work.
4. I don't use public computers or shared devices for your work.

8. Code handling

1. Your code stays in your repository or in a private fork that you control.
2. I don't copy your code into public repositories, public gists, public issue trackers or other clients' projects.
3. I don't add a dependency with a known critical vulnerability. I list each new dependency and its license in the pull request.
4. I add no strong copyleft dependency (for example GPL or AGPL) without your written approval.

- 5. I don't add telemetry, tracking or remote access code.

9. Security incidents

- 1. A security incident is any event where your code, data, secrets or access may have reached someone who shouldn't have it.
- 2. I tell you in writing within 24 hours after I become aware of an incident.
- 3. The notice says what happened, what may be affected, what I have done and what I suggest you do.
- 4. I help you contain the incident within the hours of the current Week, at no extra charge if I caused it.

10. Third parties

- 1. I use no subcontractors on your work without your written consent.
- 2. Every person who sees your code under a consent you give is bound by duties of secrecy at least as strict as the Agreement.

11. When the work ends

Within 5 Working Days after the engagement ends, I do these steps and confirm each one to you in writing.

- 1. I remove or hand back every access listed in the Statements of Work.
- 2. I delete local clones of your repositories, private forks that I own, branches I created that you didn't merge, and local environment files.
- 3. I delete secrets, test keys and exports of your data from my devices and accounts.
- 4. I keep AI tool session logs that contain your code or data only in an encrypted archive that I alone can open. I delete them when you ask in writing.
- 5. I keep only the documents the Agreement allows, such as invoices and the signed Agreement, for as long as the law requires.

Your written deletion confirmation lists each item with the date it was done.

12. Your checklist

Please give

Please don't give

Write access to branches in named repositories

Admin rights to the organization or cloud account

A test or staging environment

Production database access

Please give	Please don't give
Test keys with spend caps	Production secrets
Anonymized test data	Real customer personal data, unless a data processing agreement is signed
A list of approved AI providers	Secrets in email or chat
Your own security policy, if you have one	Shared logins

13. Questions

Send any security question to michael@axdale.com. I answer in writing within 1 Working Day.

Axdale. Michael Lip, independent engineer, Warsaw, Poland (CET). michael@axdale.com. This document is plain English and isn't legal advice.

Axdale, Michael Lip. Warsaw, Poland (CET). michael@axdale.com. www.axdale.com. Code record github.com/theluckystrike. Version 1.0. 23 October 2026.