

The Personal / Corporate Boundary

For everyone · ~10 min read

One threat model, two directions

Most people think of AI safety at work as a single question: "Am I allowed to use this tool?" That framing is wrong, and it is the reason so many well-meaning employees cause leaks. The real question is not which tool you picked; it is which *jurisdiction* the account and the data belong to. You have a personal life, with personal accounts, personal credentials, and personal data. You also have a corporate life, with a company account, company credentials, and company data. The boundary between those two lives is not defined by the product name on the screen. It is defined by who owns the seat you are logged into and who owns the bytes you are about to paste.

Once you look at it that way, the boundary has two directions, and you have to defend both of them. It is not enough to keep corporate data out of personal tools. You also have to keep personal data out of corporate tools. Both crossings are violations, both have real consequences, and both are overwhelmingly caused by one thing: convenience. Whichever tool is already open wins over whichever tool is correct. This lesson exists because convenience is the single most common failure mode in AI operational security, and the only defense is to make the boundary a rule you do not bend, even once.

Direction 1 — Personal AI on corporate work

The first direction is the one most security teams already worry about. You are at work, you hit a problem, and you open your personal ChatGPT, Claude, or Gemini tab — the one tied to your private email — and paste something from work into it. Maybe it is a stack trace. Maybe it is a customer email you are trying to draft a reply to. Maybe it is a chunk of source code or a failing SQL query. The tool helps you, you get back to your day, and from your perspective nothing happened. From the company's perspective, an uncontrolled disclosure just occurred.

The failure modes are layered. First, the provider's retention policy on a personal tier is not the policy your employer would have negotiated. Personal tiers typically retain prompts by default, and they may be eligible for training unless you have explicitly opted out and verified the opt-out stuck. Second, there is no legal agreement. Your employer has no contract with that provider covering that data, which means any promises made in a consumer terms-of-service apply between you and the provider personally, not between the company and the provider. Third, there is no audit trail on the corporate side. Security cannot see that the data left, cannot measure the exposure, and cannot include it in an incident response. Fourth, the sub-processors that the AI provider uses — cloud hosts, moderation vendors, observability platforms — are now in the picture, and none of them were approved by your company's vendor review.

People cross this direction for reasons that all sound perfectly innocent in the moment. "It's just a stack trace." "I redacted the obvious bits." "It's faster than waiting for the enterprise tier to load." "The corporate account doesn't have the model I want." Every one of those excuses describes convenience, and convenience is exactly the lever that turns a one-off into a habit. The tool does not know the difference between a redacted trace and an unredacted one. It receives everything you paste, and so does its logging pipeline.

Direction 2 — Corporate AI on personal work

The second direction is quieter and almost never discussed, but it is just as dangerous — and for you personally, it may be worse. You have a corporate Claude or ChatGPT seat. You are already logged into it. A friend sends you a piece of code over the weekend and asks for a second pair of eyes. You paste it in. Or you are working on your own side project in the evening, still signed into the corporate tool from earlier in the day, and you use it to generate a helper function. Or you are writing a hobby blog post and you ask the corporate assistant to polish your draft. Each of these is a boundary crossing, and each of them carries a different set of risks.

Intellectual property assignment is the first problem. Most employment contracts contain clauses about what counts as a work product, and using corporate tools to produce anything — even a weekend side project — can pull that work into the scope of those clauses. Retention is the second problem: your personal prompts are now in corporate audit logs, forever, and anyone who can see those logs can see what you

were doing on your own time. Acceptable-use policy violation is the third problem, and it is the one that can cost you your job, because corporate AI seats are almost always contractually limited to corporate work. Audit noise is the fourth: security tooling designed to flag unusual patterns will flag the content that does not look like your day job, and you do not want to be answering those questions. And finally, there is the simple privacy problem that your own personal life is now sitting inside a system your employer fully controls and can review at any time.

The linked-account trap

There is a third category of boundary crossing that does not require you to paste anything at all. It is the one that happens in the background, silently, the moment you grant a tool access to something it should not see. OAuth grants, browser extensions, and IDE plugins are the usual culprits. You install a helpful-looking AI assistant in your browser, and you are already signed into corporate Gmail in the same profile. You "Log in with Google" to a personal AI product using the account you use for everything else, and that account is linked to dozens of other services. You install an IDE plugin that promises to explain your code, and your IDE happens to have both a personal GitHub repo and a corporate monorepo open in adjacent tabs.

In every one of these cases, the tool's blast radius is the union of everything it can see. There is no wall inside the plugin separating the corporate tab from the personal tab. There is no wall inside the browser extension separating the corporate Gmail from the personal Gmail. There is no wall inside the OAuth grant separating the scopes you thought you gave from the scopes you actually gave. The moment two jurisdictions share a context, they share a blast radius.

Remember

One linked account = one blast radius.

Habits beat intent

If you read the two directions and the linked-account trap carefully, you may have caught yourself thinking, "Sure, but I would never do that on purpose." That is the wrong defense, because the overwhelming majority of boundary violations are not on-purpose decisions. They are accidents of convenience that harden into habits over a handful of incidents. You paste a log file into your personal ChatGPT once because the corporate tool is slow. A week later you do it again because you remember the first time worked out fine. A week after that, you stop noticing which tab you are in. By the end of the month, you have a habit, and a habit is indistinguishable from a policy you have written for yourself.

The "just this once" moment is the moment to catch. It is the only moment you have a chance of noticing the boundary at all, because after that your brain quietly promotes the exception to the rule. This is why professional operational security leans so heavily on bright-line rules. A bright line is a rule with no exceptions — not because every exception is catastrophic, but because the exceptions erode the rule. If you let yourself cross the boundary once for a "safe" reason, you will cross it again for a less-safe reason, and then a less-safe one, and you will not notice the drift until something goes wrong.

What good looks like

The defense against both directions of the boundary and against linked-account traps is boring, mechanical, and very effective. It is physical separation of the two jurisdictions at the account and browser level, with no exceptions and no shortcuts.

- **Separate browser profiles** — one profile for corporate, one profile for personal. Different colors, different bookmarks, different extensions, different sign-ins. Never mix.
- **Separate accounts** — corporate AI tools are signed in with the corporate identity only. Personal AI tools are signed in with the personal identity only. Never cross-login.
- **No "Log in with Google" across the boundary** — never authorize a personal tool with your corporate account, and never authorize a corporate tool with your personal account.
- **No browser extensions across the boundary** — corporate profile gets only the extensions your company approves. Personal profile gets whatever you want, but it

cannot see corporate tabs.

- **Corporate work only on corporate tools** — no exceptions for speed, model availability, or because the other tab was already open.
- **Personal work only on personal tools** — no exceptions for convenience, no matter how harmless the personal work feels.
- **When in doubt, switch profiles** — the five seconds it takes to change windows is the cheapest security control you will ever deploy.

None of this requires technical skill. It requires discipline and a mental model in which the boundary is a hard rule, not a soft preference. Once the separation is physical, the question "am I on the right side?" stops being a judgment call and becomes a glance at which browser profile is open.

Rules to remember

- Personal AI accounts never touch corporate code, data, credentials, tickets, or customers.
- Corporate AI accounts never touch personal projects, friends' code, or side-gigs.
- One browser profile per jurisdiction. No cross-login.
- Treat linked accounts and OAuth grants as shared blast radius — assume anything the tool can see is now in the other tool.
- The first "just this once" is how every leak starts. There is no "just this once."