

Lesson 02 — The Personal / Corporate Boundary

Where your personal AI life ends and your corporate AI life begins — and why crossing that line in either direction is the most common way leaks happen.

Audience: Everyone

AI OpSec

Foundations

Notes: No technical background required. The whole lesson is a single boundary and five rules.

What you will learn

- **Both directions of the boundary** — personal-into-corporate and corporate-into-personal
 - **Why "convenience" is the #1 failure mode** — the tool that is already open wins over the tool that is correct
 - **How a single linked account blows both sides** — OAuth grants and extensions quietly merge your two lives
-

Two directions, one rule

Personal AI ← Corp data

Company code, tickets, customers pasted into your own account. **Red arrow.**

Corp AI ← Personal data

Friend's code, side projects, hobby work pasted into your company account. **Red arrow.**

Both arrows are equally wrong. The boundary is about *whose jurisdiction owns the data and the account*, not about which tool you like better.

Direction 1 — Personal AI touching corporate work

You open your own ChatGPT, Claude, or Gemini account and paste something from work into it — a stack trace, a ticket, a customer email, a snippet of source.

- The company has **no contract** with that provider
 - The company has **no control** over retention, training, or sub-processors
 - The company has **no audit trail** — security cannot see what left
 - The data now lives in a jurisdiction the company never approved
-

Direction 2 — Corporate AI touching personal work

You use your company Claude or ChatGPT seat for something that has nothing to do with work: a friend's project, your own side-gig, a hobby repo.

- **IP assignment confusion** — contributions made with corp tools can become the company's
 - **Corp retention of personal content** — your weekend code is now in corporate logs forever
 - **Policy violation** — acceptable use policies forbid non-work use
 - **Disciplinary exposure** — audit logs show you doing non-work on corp tools
-

The "just once" fallacy

Every leak starts as a one-off

Nobody plans to build a habit out of pasting corp data into a personal account. It happens once because the tool was open and the clock was running. Then it happens twice. Then it is how you work.

- Habit forms in about **3 incidents**
 - The third time, you no longer even notice you crossed the line
 - "I'll be careful" is not a control
-

Linked accounts = shared blast radius

OAuth grants, browser extensions, and IDE plugins quietly merge your two sides into one context.

Example

Your personal GitHub is logged into a corp IDE extension. Now every prompt and every file the extension reads travels through the same account context — personal credentials, corporate source, one blast radius.

- If it can read both sides, assume both sides leak to the other
 - "Logged in with Google" is a boundary crossing, not a convenience
-

Example A — internal stack trace into personal ChatGPT

A developer hits a weird error at work and pastes the full stack trace into their personal ChatGPT to ask for help.

- **Source code** — file paths and function names from the trace
- **Internal names** — service and module names tied to the roadmap
- **Infra topology** — hostnames, environments, service discovery hints
- **Possibly secrets** — tokens or connection strings that ended up in the trace

All four categories, into an account the company does not own. Direction 1.

Example B — corp Claude helping a friend's side project

A developer's corporate Claude session is already open. A friend asks for help debugging a weekend SaaS. The developer pastes the friend's code in and asks for a review.

- **IP bleed** — contributions suggested by the corp tool may be legally messy
- **Policy violation** — acceptable use covers corp work only
- **Corp retention** — the friend's code is now in corporate audit logs
- **Audit noise** — security sees unknown code flowing through a corp seat

Direction 2 — and it feels harmless, which is exactly the problem.

Example C — personal AI extension in a corp browser

An engineer installs a personal AI browser extension and logs into corp Gmail in the same browser profile.

- Everything the extension can read in corp Gmail now flows to a **personal account provider**
 - There is no contract, no zero-retention, no audit trail
 - The engineer did not "paste" anything — the extension did
 - Direction 1, at scale, and running silently in the background
-

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."

