

Identity Depersonalization

For devs and operators · ~12 min read

Every AI account you sign into is building a profile of you. Some of that profile is the stuff you already know about — your email address, your billing details, the prompts you typed. Most of it is not. A modern AI provider sits on top of a stack that collects account metadata, device telemetry, network signals, behavioral timing, and the linguistic shape of everything you write. If you ever need to operate an AI account that is not publicly tied to your real identity — for research, red-teaming, safety work, sensitive journalism, or any other lawful reason — you have to understand what you are up against. "Using a different email" is not a persona. This lesson is about what an actual persona looks like, and why most of them fail on the fingerprints their author did not know they were leaving.

The fingerprint inventory

Start by taking an honest inventory of everything an AI account accumulates. It falls into five layers, and an analyst correlating two accounts to the same human will try to match on every one of them.

Account metadata includes the billing name, billing address, card BIN, card last-four, and the history of payment methods the account has ever used. It includes the email address, any recovery email, any phone number attached for SMS verification, and the full set of OAuth connections the account has ever authorized. It includes the account's creation timestamp, its first-login IP, and the provenance chain of how the account was created (signup flow, referral code, invite token). None of this is visible to you in the UI, but all of it lives in the provider's database and in its fraud and abuse systems.

Device signals include the user-agent string, the operating system version, the browser version, the screen resolution and color depth, the list of installed fonts, the audio-context fingerprint, the WebGL renderer string identifying your GPU, the canvas fingerprint, the hardware concurrency (CPU cores), the device memory hint, and the timezone offset. In a native app or a desktop client, device signals additionally include

the machine's hostname hash, installed extensions, locale settings, and the exact build number of the client.

Network signals include every IP address the account has ever logged in from, the ASN and geolocation of each, and the pattern of which networks you use when. They include TLS fingerprints (JA3/JA4) that identify your client stack independent of the user-agent. They include DNS patterns from any bundled telemetry, and they include latency-timing patterns that leak geographic proximity even when the IP is hidden.

Behavioral signals include the timing between keystrokes, the rhythm of your session starts and stops, the cadence of your prompts per hour, the days and hours when you are typically active, the speed at which you read responses before sending the next message, the rate at which you copy versus retype, and the mouse-movement fingerprint in web clients. None of this is information you consciously chose to share, but all of it is logged somewhere.

Linguistic signals are the most underestimated of the five. They include your vocabulary, your sentence length distribution, your preferred idioms, your punctuation habits, your typos, your capitalization style, your preferred phrasings for common requests, the technical terms you treat as known versus explained, and the specific way you structure a multi-step request. Linguistic signals travel across accounts effortlessly — you take them with you — and they are the hardest to scrub.

Writing-style as a fingerprint

Stylometry is the statistical study of writing style as an identifier, and it is very old — it has been used to attribute anonymous pamphlets to specific authors for more than a century. What is new is that modern language models have made stylometry cheap, fast, and extremely accurate. An LLM can be prompted to compare two short passages and judge, with surprising confidence, whether they were written by the same author. Even simple classical features — average sentence length, type-token ratio, function-word frequency — give a reliable signal over a few hundred tokens. Once you are producing thousands of prompts under a persona, the corpus is more than enough to re-identify you.

The features that leak most strongly are the ones you do not notice yourself using. Idioms you picked up in school. The way you open a request ("hey can you..." versus "Please generate..."). Whether you use Oxford commas. Whether you use contractions.

Whether you write out numbers under ten. Whether you end prompts with "ta" or "thanks" or nothing. Whether you ask for "skeletons" or "stubs" or "scaffolds" or "templates." Whether you abbreviate "implementation" to "impl." Whether you capitalize acronyms consistently. Every one of those is a bit of information, and a few dozen bits is enough to single out one person in a population of a million.

Remember

Your prose is a signature.

The practical consequence is that a persona needs a deliberate voice, and that voice has to be different from your own, and it has to be *consistent* within the persona. Drifting back to your natural voice for even a single session can re-identify you, because stylometric classifiers do not need much text to flip. The best defense is to write down, explicitly, what the persona's voice looks like — vocabulary choices, sentence shape, common phrasings — and refer to it before every session.

Tool and telemetry leaks

Stylometry is one half of the problem. The other half is that your tool chain itself is a fingerprint, and most of it is telemetry you never see. IDE plugins phone home with installation IDs. Analytics SDKs report usage events keyed to a device ID. Crash reporters upload stack traces with machine identifiers. Auto-update pings carry a persistent installation UUID. Font lists, audio context, and WebGL renderer strings together produce a canvas fingerprint that is stable across browser sessions on the same machine. None of these were designed to deanonymize you, but together they narrow the field enough that a single additional signal — a matching IP, a matching OAuth identity, a matching idiom — can close the identification.

For a dev-facing persona, the riskiest telemetry sources are often the ones you trust the most: the IDE, the terminal emulator, the package manager, the code host CLI, and any AI-assistant plugin. Each of these was written by a company that has its own legitimate interest in knowing who is using the product, and each of them writes telemetry that the AI provider can correlate to. The safest assumption is that any tool on

your persona machine is reporting to someone, and the less overlap there is between the persona's tool set and your real tool set, the weaker the correlation.

OAuth and the linking problem

OAuth is the most efficient deanonymization primitive on the modern web. When you click "Log in with GitHub" on a persona account, you have just asked GitHub to attest to the persona provider that the persona and your real GitHub identity are the same person. The provider stores that attestation forever. GitHub stores it too. Any future investigator who subpoenas either side can pivot from the persona to the real identity in one hop, and they do not even have to correlate anything — the link is explicit and cryptographically signed.

The temptation to OAuth a persona is always some flavor of convenience: "it's just for login," "I'll unlink it later," "I only need it for the API scope." None of those survives contact with reality. Unlinking an OAuth grant does not retroactively delete the server-side attestation. Scopes are only the ceiling on what the grant can do, not the floor — the linking fact is there regardless of scope. And "just for login" means you gave the persona provider a provider-attested binding between the persona and your real identity in exchange for saving thirty seconds on a signup form. That is never worth it.

Designing a clean persona

A persona that can actually survive scrutiny is built from the bottom up, in a specific order, and each layer is independent of the previous one. The order matters: if you skip a layer or reuse something from your real identity, every layer above it is compromised.

Device. Either a dedicated physical machine or a dedicated virtual machine that has never been signed into a personal account. If it is a VM, it should be provisioned fresh, with its own locale and timezone settings, and it should never share a clipboard, a filesystem, or a network with the host's personal state. The device must have its own distinct set of installed software — if your real machine runs Cursor, Raycast, and 1Password, the persona machine should not.

Network egress. The persona should not share an IP address with your real identity. A clean approach is to route persona traffic through a stable egress that is not your home or office network and is not used for any other account. Avoid free VPNs, which are

correlation-heavy. Avoid rotating through a dozen IPs, which trips fraud systems and gets the persona flagged independently.

Account creation. Fresh email address at a provider you do not use for anything else, created from the persona device on the persona network. Fresh phone number for SMS verification — never your real number, never a number your real identity has ever touched. No OAuth links, ever. Password set manually, stored in a separate vault that contains nothing else.

Payment. Fresh payment trail — prepaid card bought for cash, virtual card issued under a different identity, or any funding source that does not share a billing name, address, or BIN country with your real cards. Reusing a personal card for even a single subscription unifies the two identities in the provider's billing database forever.

Voice. Write down the persona's voice explicitly: opening phrase, closing phrase, punctuation style, contractions or not, typical request shape, typical technical vocabulary. Refer to it before every session. Re-read the most recent persona prompts before writing new ones, to keep the voice consistent. Do not drift toward your natural voice even once.

Cadence. Decide in advance which hours the persona is active, how many prompts per session is typical, and how long the persona typically spends between sessions. Behavioral timing is one of the signals you cannot edit after the fact; the only way to control it is to be deliberate about it from the beginning.

Cross-contamination

Cross-contamination is what happens when a persona and your real identity become linked in a provider's system — by OAuth, by billing, by stylometry, by recommendation-engine correlation, or by a mistake you made once and did not notice. You usually find out about it through symptoms rather than notifications. The persona account starts receiving recommendations that match your real interests. "People you may know" surfaces your real contacts. The persona's suggested content drifts toward your real reading history. An account warning mentions details that only make sense if the provider already knows who you are.

When you see any of these symptoms, the persona is burned. The only correct response is to destroy it and start over. Do not try to rehabilitate a contaminated

persona. Rehabilitating a persona means continuing to operate a known-linked account, which is worse than not having a persona at all — the linkage gives the provider a point of correlation for everything the persona does from that moment forward. Delete the account, discard the device or reimage the VM, rotate the payment method, and build a new persona from scratch with the lessons of the old one. The cost of burning a persona is always lower than the cost of operating a linked one.

Rules to remember

- Your account knows more than you typed: billing, IP, device, OAuth, prompt history, and prose style.
- One persona = one browser profile = one device (or dedicated VM) = one payment trail.
- Never use "log in with <real-account>" on a persona.
- Your writing style is a fingerprint. Persona = persona voice, consistently.
- If contamination is suspected, burn the persona. Do not rehabilitate it.