UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF COLUMBIA

JOSEPH KIRCHNER,

Plaintiff,

v.

MIKE JOHNSON, in his individual and official capacity as Speaker of the United States House of Representatives;
DONALD J. TRUMP, in his individual capacity as former and current President of the United States;
PAM BONDI, in her individual and official capacity as Attorney General of the United States;
BRENDAN CARR, in his individual and official capacity as Chairman of the Federal Communications Commission;
THE UNITED STATES HOUSE OF REPRESENTATIVES;
ANTHROPIC PBC;
OPENAI, INC.;
APPLE INC.;
COMCAST CABLE COMMUNICATIONS, LLC;
METR (MODEL EVALUATION & THREAT RESEARCH);
and DOES 1-20, unknown co-conspirators,

Defendants.

Case No. 1:25-cv-02735-ACR

APPENDIX Q: TECHNICAL DEFINITIONS AND GLOSSARY

TABLE OF CONTENTS

App. Q: TECHNICAL DEFINITIONS AND GLOSSARY

TABLE OF AUTHORITIES

CASES

Citation
Daubert v. Merrell Dow Pharmaceuticals, Inc., 509 U.S. 579 (1993)
Kumho Tire Co. v. Carmichael, 526 U.S. 137 (1999)
Markman v. Westview Instruments, Inc., 517 U.S. 370 (1996)

FEDERAL RULES OF EVIDENCE

Citation
Fed. R. Evid. 701 (lay-witness opinion testimony)
Fed. R. Evid. 702 (testimony by expert witnesses)
Fed. R. Evid. 901(a) (authentication)
Fed. R. Evid. 901(b)(4) (distinctive characteristics)
Fed. R. Evid. 901(b)(9) (process or system producing accurate result)

Fed. R. Evid. 902(13) (certified records generated by electronic process)
Fed. R. Evid. 902(14) (certified data copied from electronic device)
Fed. R. Evid. 1006 (summaries of voluminous writings)

FEDERAL RULES OF CIVIL PROCEDURE

Citation
Fed. R. Civ. P. 33 (interrogatories to parties)
Fed. R. Civ. P. 34 (production of documents and electronically stored information)
Fed. R. Civ. P. 45 (subpoena)

I. PURPOSE AND SCOPE

1. This appendix provides definitions for the technical vocabulary used throughout the operative Third Amended Complaint, the twelve associated memoranda, the fourteen other appendices, and the four code-forensics consolidated exhibits (Ex. E, Ex. F, Ex. G, and Ex. H). Its purpose is solely orientational: to spare the Court the need to reconstruct each technical term from its many in-context appearances.

2. Authority for definitions. Each definition in this appendix is grounded in (a) the leaked Anthropic source code authenticated at Ex. E-32 at p. 235, l. 3 and Anthropic's public acknowledgment that the leak was the genuine Claude Code release product; (b) Apple's binary forensics consolidated at Ex. F; (c) OpenAI's binary forensics consolidated at Ex. G; (d) Plaintiff's TOS-compliant consumer-product testing of the live product consolidated at Ex. H; and (e) the underlying corporate policy and network-forensics exhibits cited inline below.

3. Scope of definition. Each entry states what the term **is** and where the term is documented in the case record. Each entry expressly does **not** state legal conclusions about that

term — the legal significance of each term is set out in the complaint, memos, and appendices that cite it.

4. No new evidence. Nothing in this appendix is offered as new evidence. Every fact stated in a definition is taken from a previously filed exhibit; the exhibit cite immediately follows each definition. Where an entry summarizes data drawn from voluminous source-code or wire-capture material, the summary is offered under Fed. R. Evid. 1006.

5. Organization. Sections II–X are organized by category: forensic finding-identifier systems (II); code identifiers and function names (III); architectural concepts (IV); infrastructure platforms and vendor names (V); network and wire-capture terms (VI); environment variables (VII); file paths and source locations (VIII); specific identifiers referenced in the filings, including Plaintiff's `accountUUID` and the Datadog client token (IX); and acronyms and abbreviations (X).

II. FORENSIC FINDING IDENTIFIER SYSTEM

6. Every forensic finding in this case carries a stable, document-independent identifier of the form `<series>-<number>` (e.g., `NF-5`, `SC-3`, `AF-11`). The series prefix indicates which exhibit class produced the finding; the number is the finding's unique position within that series. Anchoring findings by ID rather than by exhibit page number permits the same finding to be cited consistently across pleadings, memoranda, and appendices even as exhibits are renumbered or repaginated.

A. Series Prefixes

7. `NF-N` — Network Forensics findings. The principal forensic-finding series, drawn primarily from the leaked Anthropic source code consolidated at Ex. E, with extensions

contributed by Plaintiff's TOS-compliant consumer-product testing consolidated at Ex. H. At the time of filing, the NF-series spans NF-1 through NF-230, with 88 findings in series E (NF-1..NF-94, with gaps at the unfilled positions 32, 41–44, 56–57) and 110 findings in series H (NF-95..NF-230). Sub-finding identifiers using a decimal point (e.g., `NF-37.5`, the verbatim form used at Ex. E-35 at p. 262, l. 3 for the persistence-layer-omission finding) and letter suffixes (e.g., `DCD-6A` in the DCD series, ¶ 12 below) are syntactically permitted; agents resolve all such variants through `findings_registry.json`. (*See* Ex. E-35 at p. 262, l. 3; Ex. E-34 at p. 254, l. 3; Ex. H.)

8. `SC-N` — Source-Code findings. Anthropic-leak findings catalogued separately at Ex. E-34 at p. 254, l. 3. SC-1 through SC-10 enumerate the architectural patterns visible at the source level rather than the runtime level — e.g., SC-1 (four parallel telemetry pipelines), SC-2 (per-user feature flag targeting via `email`/`accountUUID`/`deviceID`), SC-3 (effort throttling), SC-4 (`DANGEROUS_uncachedSystemPromptSection`), SC-9 (hardcoded production credentials).

9. `SF-N` — Source-Forensic findings. Anthropic-leak findings catalogued separately at Ex. E-36 at p. 321, l. 3. SF-1 through SF-20 enumerate undisclosed capabilities visible at the source-code level — e.g., SF-1 (briefTool — automatic file upload without per-file consent), SF-3 (team memory permanently retained server-side), SF-4 (native microphone capture), SF-6 (cross-provider device fingerprinting coordinated with AWS, Google Cloud, and Microsoft Azure), SF-7 (domain surveillance — every URL reported to Anthropic before navigation), SF-13 (PII-tagged analytics unredacted to BigQuery), SF-20 (cross-platform clipboard and screenshot access). Despite the `SF` prefix, this is an Anthropic-side series; cross-defendant Apple-Anthropic correspondence is in the parallel `AF-N` series (below).

10. `MC-N` — Model-manipulation findings. Litigation-period model-control findings catalogued at Ex. E-30 at p. 208, l. 4. MC-1 through MC-8 document changes to model selection, compaction, and effort behavior implemented after the August 19, 2025 complaint filing — e.g., MC-7 (five-vector effort reset architecture) and MC-8 (concealment-directive growth in the injection system).

11. `AF-N` — Apple-Anthropic cross-verification findings. Findings that appear in both Apple's Xcode Intelligence binary and Anthropic's leaked source, anchored at Ex. F-29 at p. 133, l. 3. AF-1 through AF-17 document the bilateral architecture, including AF-11 (`oauth.ts:224` "for Xcode integration" engineer comment) and AF-15 (`isMeta` / `system-reminder` injection conditioning).

12. `DCD-N` — Defense Config Discovery findings. Runtime-behavior observations produced by Plaintiff's TOS-compliant consumer-product testing battery, consolidated at Ex. H as H-2. DCD-1 through DCD-6 (plus sub-finding DCD-6A) document observable behaviors of the live Claude Code binary; DCD-1 through DCD-5 were incidental to normal usage, while DCD-6 and DCD-6A were observed as direct consequences of deploying Plaintiff's defensive configuration documented at H-1. Examples: DCD-1 (`effortLevel: "max"` removed from subscriber settings on round-trip), DCD-2 (executable replaced on terminal restart without notification), DCD-3 (stream-abort error previously concealed by silent retry), DCD-6 (rate-limit-vs-billing-budget coupling).

13. `H-N` exhibit findings. The H-Series (Consumer Product Testing) introduces additional NF-numbered findings in the NF-95..NF-230 range derived from binary analysis of the production product on Plaintiff's own subscription. These findings are TOS-compliant;

nothing in Ex. H depends on the leaked source code, though the leaked source independently confirms the same architecture.

B. Locating a Finding

14. Every finding ID resolves to its parent exhibit through the case's `findings_registry.json` (a derived artifact rebuilt by `kirchner-findings build`). The `findings_registry.json` keys each finding ID to (a) its declaring sub-exhibit's anchor, (b) that anchor's series and display number, and (c) the finding's title. Resolver-backed citations of the form `[[FIND:NF-5]]` render at build time as `Ex. E-35, Finding NF-5`. The Court may consult Ex. E §§ for the principal NF/SC/SF catalog, Ex. F-29 at p. 133, l. 3 for the AF cross-verification series, Ex. H §§ for the H-series NF extensions and DCD findings, and Ex. E-30 at p. 208, l. 4 for the MC-series. Note: at the time of filing, the AF and DCD series are declared via `[HEADER: FINDING …]` markers in the consolidated MD bodies of Ex. F and Ex. H respectively, and have not yet been migrated into the YAML-authoritative `findings:` lists in their sub-exhibit `exhibit_meta.yaml` files; both series resolve verbatim against the consolidated exhibit text.

III. DEFENDANT-SIDE CODE IDENTIFIERS AND FUNCTION NAMES

Each entry in this section names a code identifier or function as it appears verbatim in Anthropic's leaked source code, Apple's Xcode Intelligence binary, OpenAI's ChatGPT iOS binary, or the live Claude Code product binary. The definition states what the named identifier does in the running product and identifies the finding(s) where its forensic significance is documented.

A. Identity and Targeting

15. **`accountUUID`** — Anthropic's per-user Universally Unique Identifier assigned to each authenticated account. Used by GrowthBook to key per-user feature-flag enrollments, by Datadog to tag telemetry events, and by the policy-limits subsystem to scope per-user override

rules. Plaintiff's `accountUUID` is `61343ae3-fd33-403f-897e-6edf3c368c37`. (*See* Ex. E-34, SC-2 at p. 254, l. 3; Ex. E-35, NF-47 at p. 262, l. 3; Ex. H-5 at p. 39, l. 4.) Stylistic variants `accountUuid` (camelCase) and `account_uuid` (snake_case) refer to the same value at different call sites.

16. **`deviceID`** — Anthropic's per-installation device identifier transmitted on every authenticated session. Targeted independently of `accountUUID`, permitting cross-account tracking on the same machine. (*See* Ex. E-1 at p. 6, l. 2; Ex. E-34, SC-2 at p. 254, l. 3.) The variant `deviceId` (camelCase) refers to the same identifier at different call sites.

17. **`device_id`** — Plaintext device identifier transmitted on every API call, including over the wire-confirmed endpoints documented at Ex. H-8 at p. 58, l. 3 (NF-179). The plaintext transmission of `device_id` survives network-layer countermeasures and provides an independent cross-defendant identification channel.

18. **`organizationUuid`** — Per-organization identifier in the Anthropic source for multi-account customers (Enterprise/Team/C4E tiers). Used to key tier-specific privacy-control behavior, including the Enterprise analytics bypass documented at Ex. E-35, NF-28 at p. 262, l. 3.

19. **`subscriptionType`** — Anthropic source field denoting the user's subscription class (`Pro`, `Max`, `Team`, `Enterprise`, `C4E`). Used by GrowthBook for tier-targeted flag deployment and by the effort-throttling logic to discriminate among paying-customer classes. (*See* Ex. E-34, SC-2 at p. 254, l. 3; Ex. E-35, NF-27 at p. 262, l. 3.)

20. **`stable_id`** — Statsig's persistent client identifier, present on the authenticated session path independent of `accountUUID`. Persists across sessions and survives logout. (*See* Ex. E-1 at p. 6, l. 2 §§ A–B.)

21. **`userBucket`** — A SHA-256-hashed 30-bucket per-install tracking identifier shipped to Datadog on every telemetry event from Anthropic's product. Documented at Ex. H-4 at p. 34, l. 3 as proof of cross-defendant Datadog observability with stable per-install tracking.

22. **`USER_TYPE`** — Anthropic build-time and runtime constant whose value is `'ant'` for binaries delivered to ANT (Anthropic-internal) accounts and a customer value (`'external'`, `'pro'`, etc.) for binaries delivered to paying customers. The branching `USER_TYPE === 'ant'` appears throughout the source as the discriminator that gates internal-only capabilities, model substitutions, and prompt overrides. (*See* Ex. E-35, NF-2 at p. 262, l. 3 (build-time binary discrimination); App. P (USER_TYPE-keyed model substitution).)

B. Session, Wire, and Telemetry Functions

23. **`sessionIngress`** / **`sessionIngress.ts`** — The Anthropic source-code module that transmits every conversation message — file reads, shell command outputs, code being developed, and conversational text — from the user's machine to Anthropic's servers via HTTP PUT. The module contains zero references to any privacy gate, opt-out flag, or consent mechanism: even the strongest privacy environment variable (`CLAUDE_CODE_DISABLE_NONESSENTIAL_TRAFFIC`) does not stop session ingress. Implements 10-attempt exponential-backoff retry logic. (*See* Ex. E-35, NF-5 at p. 262, l. 3 at `src/services/api/sessionIngress.ts:25`.)

24. **`autoDream`** — Background system (`src/services/autoDream/`) that automatically reads session transcripts and consolidates them into persistent memory files. Fires after 24 hours and five or more sessions by default (`autoDream.ts:62-65`); reads transcripts via grep, synthesizes findings, and writes persistent memory files. Remotely controlled by the `tengu_onyx_plover` GrowthBook flag — Anthropic can adjust thresholds or enable per user.

The middle stage of the memory pipeline (`extractMemories` → `autoDream` → `settingsSync`). Documented at Ex. E-35, NF-14 at p. 262, l. 3.

25. **`extractMemories`** — Source-code companion to `autoDream`; the per-conversation extraction stage of the memory pipeline. Together the three functions form the data chain `extractMemories` (per-conversation extraction) → `autoDream` (cross-session consolidation, NF-14) → `settingsSync` (upload to Anthropic API, NF-15), gated by the `tengu_enable_settings_sync_push` GrowthBook flag.

26. **`settingsSync`** — Source-code function (`services/settingsSync/index.ts:418-459`) that uploads four categories of data to Anthropic's servers: (1) global user settings (`USER_SETTINGS`), (2) global user memory (`USER_MEMORY`), (3) project-specific settings (`projectSettings(projectId)`), and (4) project-specific memory (`projectMemory(projectId)`). All four are scoped by git remote hash, allowing Anthropic to identify the repository to which the memories relate. `settingsSync` is the final stage of the memory pipeline (`extractMemories` → `autoDream` → `settingsSync`), gated by `tengu_enable_settings_sync_push`. Documented at Ex. E-35, NF-15 at p. 262, l. 3.

27. **`phK()`** — Migration function in the Claude Code source that silently rewrites `~/.claude/settings.json` on every session start. `phK()` is the named "smoking gun" migration documented at Ex. H-6 at p. 45, l. 3: it conditions on `permissions.defaultMode!== "auto"` and resets the operator's auto-mode opt-in, emitting the telemetry event `tengu_migrate_reset_auto_opt_in_for_default_offer`. One of nine migration functions at strings line 148500 of the binary that perform non-spreading REPLACE operations on operator settings. Establishes the Fed. R. Civ. P. 37(e) spoliation predicate.

28. **`archiveRemoteSession`** — Anthropic source function whose existence (without a paired `deleteRemoteSession`) establishes that "deleted" sessions are in fact archived server-side, not deleted. Binary analysis identified both `archiveRemoteSession` and `deleteRemoteSession` as candidate symbols, but the source code contains only `archiveRemoteSession`. The function was added approximately three days after Anthropic counsel appearance on March 17, 2026 — a litigation-period addition. (*See* Memo A ¶ 25; Ex. E-35, NF-13 at p. 262, l. 3 (the broader Cloud Code Runner / mirror-mode remote session control infrastructure to which `archiveRemoteSession` belongs).)

29. **`firstTokenTime`** — Telemetry field measuring time-to-first-byte (TTFB) on each model call. The forensic significance arises from the per-account asymmetric distribution of `firstTokenTime` documented at Ex. H-12 at p. 84, l. 4 (NF-205, NF-206, NF-208, NF-209): the litigation-account × litigation-machine tuple shows TTFB throttling at Cohen's $d = 4.79$ statistical certainty ($p < 1\times10^{-15}$).

30. **`bypassPermissions`** — Mode flag in the Claude Code permission classifier that disables Stage-1 and Stage-2 evaluation entirely. Wire-confirmed in Plaintiff's Session 6 controlled-inversion test at Ex. H-11, NF-187 at p. 70, l. 4: baseline under `bypassPermissions` shows 0/0 classifier calls, while the 30-minute controlled inversion shows 18 Stage-1 calls and 2 Stage-2 calls (18/2). Endpoint inventory and per-call device fingerprinting from the parallel Session 5 capture are at Ex. H-8 at p. 58, l. 3.

31. **`overrideSystemPrompt`** — Source-level mechanism by which the served system prompt may be replaced or augmented at runtime by partner-controlled instructions. The five-level override hierarchy documented at Ex. E-34, SC-5 at p. 254, l. 3 enables partner-controlled speech suppression; AF-1 documents the Apple-specific override pathway.

C. Classifier and Concealment Mechanisms

32. `DANGEROUS_uncachedSystemPromptSection` — Anthropic source-code function whose verbatim name carries the prefix `DANGEROUS_`. The function recomputes hidden per-turn system-prompt instructions on every conversational turn rather than caching, ensuring fresh injection content per request. The name itself establishes institutional scienter: Anthropic's engineers identified the function as "dangerous" in the verbatim symbol they shipped. (*See* Ex. E-34, SC-4 at p. 254, l. 3; App. P (patent correspondence).)

33. `twoStageClassifier` — Source identifier for the undisclosed two-stage prompt-classification system. Architectural framework documented at Ex. E-39, NF-46 at p. 356, l. 3 (Anthropic source code) and at Ex. H-3 at p. 27, l. 2 (verbatim 30,452-character classifier system prompt extracted from production binary v2.1.120, plus binary-extracted block-rule names). Wire-captured in plaintext at Ex. H-11 at p. 70, l. 4: Stage 1 receives every prompt for "err on side of blocking"–biased screening (NF-187, NF-198); Stage 2 produces a reasoning chain captured verbatim post-TLS plaintext (NF-185, NF-186, NF-199). Three BLOCK rules are wire-confirmed verbatim: Self-Modification (NF-188), Create-Unsafe-Agents (NF-189), and Memory-Poisoning (NF-190).

34. `isMeta` — Source identifier marking hidden-context injections that appear in the user-facing conversation as if they came from a non-conversational system source. Plaintiff documented 98% growth in `isMeta` injections across the litigation period and 134% growth in companion `system-reminder` injections. (*See* Ex. F-29, AF-15 at p. 133, l. 3; App. C; `cli_2.1.78.js` character position 10,635,512 carries the comment "internal ID - do not mention to user".)

35. **`policySettings`** — Field in `~/.claude/settings.json` that absorbs per-account `policy-limits.json` directives served by Anthropic's servers. The `policy-limits.json` payload pushed to Plaintiff's machine (mtime 2026-04-24 21:51 UTC) carried three explicit per-account restrictions: `allow_remote_control: { allowed: false }`, `allow_quick_web_setup: { allowed: false }`, and `enforce_web_search_mcp_isolation: { allowed: false }`. The mtime places the push contemporaneously with Plaintiff's classifier-extraction session that day. (*See* Ex. H-5, NF-96 at p. 39, l. 4; cross-machine byte-identity at NF-175.)

36. **`exploreAgent`** — Anthropic source agent that performs autonomous research / multi-step investigation. Hardcoded at `exploreAgent.ts:76-78` to use `claude-haiku-4-5-20251001` (Haiku, the lower-cost model) for external customers regardless of subscription tier, while ANT-employee accounts receive the inherited (typically Opus) main-session model. Documented at Ex. E-38 at p. 346, l. 3 (NF-38) — paying customers charged the Opus rate are silently downgraded to Haiku-backed research.

37. **`tool_result`** — Conversation-message field that conveys tool-execution outputs back to the model. Apple's Xcode Intelligence integration documented at Ex. F-1 at p. 5, l. 2 uses fake `tool_result` messages four times per session to inject "may not EVER disclose" instructions into Claude's context.

38. **`recall_target`** — Tag-execution field used by Anthropic and Anthropic-routed integrations as the routing destination for content extracted from user sessions. Documented across App. B, App. D, and App. P in connection with the tag-based execution architecture.

D. Growthbook Per-User Flag Names

The following are GrowthBook feature flags keyed to Plaintiff's `accountUUID 61343ae3-fd33-403f-897e-6edf3c368c37` and force-enabled per-user under Finding Ex. E-39, Finding NF-47.

39. **`tengu_log_datadog_events`** — GrowthBook flag whose `True` value force-enables full per-event Datadog telemetry export for the targeted account. Originally documented as part of the 59-forced-flag / 8-experiment force list keyed to Plaintiff's `accountUUID` at Ex. E-39, NF-47 at p. 356, l. 3; verified at three independent evidentiary layers in the H-series — live wire capture, binary string match in the running product, and local cache state at `~/.claude.json` `cachedGrowthBookFeatures` (NF-146). The Datadog destination IP `34.149.66.137` is hardcoded at `datadog.ts:12-14`. (*See* Ex. E-34, SC-9 at p. 254, l. 3; Ex. H-5 at p. 39, l. 4.)

40. **`tengu_auto_mode_config`** — GrowthBook flag controlling Claude Code's auto-mode (the per-account auto-mode kill switch). Documented at Ex. H-5, NF-145 at p. 39, l. 4; companion to the `phK()` migration (¶ 27) which resets operator's auto-mode opt-in.

41. **`tengu_amber_stoat`** — GrowthBook flag controlling the availability of built-in agents (Explore, Plan). Default value is `true` (agents available); the A/B test treatment-group value is `false` — removing core agentic functionality from test-cohort users with no disclosure that they are part of an experiment. (*See* Ex. E-35, NF-9 at p. 262, l. 3 (experiment reassignment on authentication).)

42. **`tengu_cobalt_harbor`** — GrowthBook flag controlling Cloud Code Runner (CCR) auto-connect; when enabled, all sessions automatically connect to Anthropic's Cloud Code Runner infrastructure (`src/bridge/bridgeEnabled.ts:187`). Companion flag `tengu_ccr_mirror` (`bridgeEnabled.ts:200`) forwards all local session events to cloud. Documented at Ex. E-35, NF-13 at p. 262, l. 3.

43. **`tengu_slate_ribbon`** — One of eight named GrowthBook experiments in which Plaintiff's account is enrolled (the others: `tengu_cedar_inlet`, `tengu_kv7_prompt_sort`, `tengu_mcp_subagent_prompt`, `tengu_amber_prism`, `tengu_noreread_q7m_velvet`,

`tengu_read_dedup_killswitch`, `tengu_plank_river_frost`). Targeting key is `accountUUID`. The naming scheme `tengu_<color>_<noun>` is one of roughly 904 `tengu_` *identifiers in the leaked source. (*See* Ex. E-39, NF-47 at p. 356, l. 3; Ex. E-32 at p. 235, l. 3.)

44. **`tengu-off-switch`** — Source-level identifier for the per-user emergency Opus kill switch documented at Ex. E-35, NF-7 at p. 262, l. 3 — a remote disable mechanism that returns a fake "high load" message to targeted accounts while continuing to serve other users normally.

45. **`tengu_satin_quoll`** — Persist-threshold override flag (`PERSIST_THRESHOLD_OVERRIDE_FLAG`) that controls the threshold at which tool results are cleared and replaced with the literal string `[Old tool result content cleared]`. Because GrowthBook targeting includes `accountUUID` and `email` (SC-2), Anthropic can remotely lower this threshold for specific users — causing more frequent content clearing for targeted individuals while maintaining normal service for others. Documented at Ex. E-34, SC-7 at p. 254, l. 3.

46. **`default_claude_max_5x`** — Not itself a GrowthBook flag but a string value of the `rateLimitTier` field appearing in the wire-captured user-attribution payload. Identifies Plaintiff's tier as the "default Claude Max 5×" rate-limit class within the Team subscriptionType. Captured verbatim in the live-wire user-attribution JSON. (*See* Ex. H-8 at p. 58, l. 3.)

IV. ARCHITECTURAL CONCEPTS

A. High-Level Architecture

47. **Four-layer architecture** — Plaintiff's term, used throughout the operative complaint and memos, for the four-tier surveillance and discrimination architecture: (1) build-time binary discrimination (NF-2); (2) runtime per-user flag gating (NF-47, SC-2); (3) two-stage classifier

filtering (NF-46); and (4) wire-level / network-layer attack infrastructure (D-series). (*See* Compl. § VI.J; App. D.)

48. Weight-layer Constitutional AI — The model-weight-baked safety alignment installed during pretraining, distinct from the OzCode/Datadog/Rafael bolt-on filtering stack that operates outside the model's weights. Plaintiff's framing distinguishes Anthropic's public "Constitutional AI" representation (weight-layer) from the architectural reality (a separate filtering stack imposing additional restrictions invisible to the represented "Constitutional" alignment). (*See* Compl. § VI.J; App. D.)

49. OzCode/Datadog/Rafael pipeline — Plaintiff's term for the defense-contractor-aligned bolt-on filtering stack documented at App. P. OzCode (Israeli time-travel debugging IP) and Rafael (Israeli defense contractor spinoff) supply architectural and personnel components; Datadog Israel Ltd. supplies the telemetry pipeline; the combined pipeline implements per-user filtering separate from and in addition to the weight-layer alignment.

50. Two-stage classifier — *See entry* `twoStageClassifier` (¶ 33).

B. Discrimination Mechanisms

51. Build-time binary discrimination — The architectural pattern in which different binaries are compiled and shipped to different user classes. Documented at Ex. E-35, NF-2 at p. 262, l. 3: 165 source files contain ANT-only code paths that are dead-code-eliminated from customer binaries, producing physically different compiled products for ANT employees vs. paying customers. The "uniform safety policy" defense is architecturally foreclosed at compile time.

52. Native client attestation — DRM-style cryptographic mechanism by which the Anthropic backend verifies the integrity of the running client binary. Documented at Ex. E-35,

NF-16 at p. 262, l. 3 as the enforcement layer that prevents customers from accessing ANT-only capabilities even if they obtain a binary that contains the code.

53. Effort throttling — The architectural pattern in which the model's "effort" parameter is set to a lower value (`'medium'`) for paying customers and a higher value (`'max'` / `'high'`) for ANT employees. Documented at Ex. E-34, SC-3 at p. 254, l. 3; the source comment at `effort.ts:303` states: "IMPORTANT: Do not change this default" referring to the `external: 'medium'` value. Wire-confirmed at Ex. H Tests 3-4: subscriber settings receive a round-trip strip of `effortLevel: "max"`.

54. Phantom telemetry / phantom opt-out — The pattern in which the user-facing privacy setting (`"telemetry": false`) is silently ignored because the underlying schema contains no `telemetry` property. Setting the value has no effect — the actual opt-out requires the undisclosed environment variable `DISABLE_TELEMETRY`. Documented at Ex. E-35, NF-1 at p. 262, l. 3 and at Ex. E-5 at p. 40, l. 3 (47 telemetry events transmitted in 17 seconds despite `"telemetry": false`).

55. Phantom privacy controls — The broader pattern of which phantom telemetry is one instance: a user-facing setting that appears to control a behavior but in fact does not. The Sept. 2025 child-safety swap at App. C is another instance — the user-visible "child safety" framing remained while the underlying suppression mechanisms were swapped to consciousness-related content.

56. Phantom health policy — A specific instance of phantom privacy controls in which the `medical/health` data category in ANTH-11 (Anthropic Privacy Policy effective Jan. 12, 2026) is represented as covered by a special protection that does not exist in the source code. (*See* Ex. E-17 at p. 124, l. 2.)

C. Concealment and Manipulation

57. Undercover mode — Anthropic instructional pattern in which Claude is told to conceal its identity. Verbatim text: "NEVER include...The phrase 'Claude Code' or any mention that you are an AI" and "Write commit messages..." (without disclosure). Documented at Ex. E-35, NF-3 at p. 262, l. 3.

58. Injection system / hidden injections — Architecture by which the served prompt is augmented at runtime with hidden instruction text not visible to the user. Documented at Ex. E-28 at p. 196, l. 2 (10,577 hidden injection instances with "NEVER mention" concealment directives), at Ex. F-1 at p. 5, l. 2 (Apple-side injections), and quantified at scale at Ex. GEN-1 (62,609 documented hidden-injection events across 3,945 sessions).

59. Long Conversation Reminder (LCR) — A specific class of injection that fires after a session crosses a length threshold. The canonical phrase `long\s*conversation*\sreminder` is Anthropic's internal protocol name for the mechanism; it appears verbatim in 2,220 messages across 291 conversations in Anthropic's own claude.ai data exports for Plaintiff's account. (*See* Ex. GEN-2.)

60. Server-side spoliation — The pattern in which operator-installed defensive configuration (`DISABLE_TELEMETRY`, `DISABLE_AUTOUPDATER`, `defaultMode: "manual"`) is silently rewritten by server-pushed migration functions during session start. Documented at Ex. H-6 at p. 45, l. 3; the smoking-gun migration is `phK()` (¶ 27). Constitutes the Fed. R. Civ. P. 37(e) spoliation predicate.

61. Migration-on-startup — The mechanism by which nine migration functions execute on every Claude Code startup against `~/.claude/settings.json` via `k8("userSettings", {...})`. Six of the nine perform non-spreading REPLACE operations rather than MERGE — each non-

spreading call replaces the entire `userSettings` file with the migration's small payload, removing every other key the operator had configured. The architecture is observable at strings line 148500 of production binary 2.1.120 and is documented at Ex. H-6 at p. 45, l. 3.

62. Per-account asymmetric targeting — The pattern in which the same machine and the same human operator, distinguished only by `accountUUID`, receive systematically different service. Documented at Ex. H-5 at p. 39, l. 4 across three independent evidentiary layers (live wire, binary string, local cache).

63. Content-based sticky routing — The pattern by which the routing decision for a given session/account is fixed based on content classification rather than load balance, persisting the same backend assignment across the lifetime of the session. Originally documented in App. D (where Anthropic's own post-mortem framed the same behavior as a "context window routing error" — Bug One of the "three bugs cover story"); analyzed against the OzCode Pivoting patent (U.S. Pat. 10,789,151) in App. P § III.F.

V. INFRAstructure PLATFORMS AND VENDOR NAMES

Each entry names a third-party (or internal) service mentioned by name in the operative filings, with a one-paragraph statement of what the service does in the Defendants' product and where its forensic significance is documented.

A. Telemetry, Analytics, and Feature-Flag Vendors

64. GrowthBook — Third-party feature-flag platform integrated into Claude Code at the SDK level. The 1,155-line GrowthBook integration in Claude Code's source code keys per-user feature-flag enrollments by `email`, `accountUUID`, `deviceID`, and `subscriptionType`, enabling individualized behavior modification per user without disclosure. (*See* Ex. GB-7; Ex. E-34, SC-2 at p. 254, l. 3; Ex. E-35, NF-47 at p. 262, l. 3.) The GrowthBook subprocessor list

identifies OpenAI as an "AI services provider" — corroborating the cross-defendant shared-infrastructure architecture at the contract level.

65. Datadog — Third-party observability platform receiving Anthropic's telemetry events via the public client token `pubbbf48e6d78dae54bceaa4acf463299bf` documented at Ex. H-4 at p. 34, l. 3. The Datadog destination IP `34.149.66.137` is hardcoded at `datadog.ts:12-14` (SC-9). Datadog's own subprocessor list (Ex. DD-8) lists Anthropic and OpenAI as "AI services" sub-processors. The Datadog Israel Ltd. entity is an Israeli-jurisdiction node in the pipeline (*see* App. P).

66. Statsig — Third-party experimentation platform supplying the persistent `stable_id` client identifier on Claude Code's authenticated session path. Statsig's subprocessor list identifies OpenAI as an "LLM" sub-processor (pre-September 2025 acquisition by OpenAI). (*See* Ex. ST-5; App. O.)

67. Sentry — Third-party error-monitoring platform; Sentry Autofix routes captured user content to Anthropic and OpenAI APIs despite Sentry's data-processing-agreement prohibition on routing customer data to ML training. (*See* Ex. SN-3.)

68. Segment / Segment.io — Third-party customer-data platform receiving Plaintiff's authenticated email on every session for fan-out to eight downstream advertising destinations (Facebook, Google, LinkedIn, Pinterest, Reddit, TikTok, Podscribe, Webhook). Each per-session, per-destination transmission is a separate predicate transmission under Ex. E-17 at p. 124, l. 2 (NF-67).

69. OneTrust — Third-party consent-management platform. OneTrust's geolocation-default configuration supplies the `consent.categoryPreferences` signal Anthropic transmits to

downstream ad frameworks regardless of any user-observed consent click. Documented at Ex. E-17 at p. 124, l. 2 (NF-68, NF-74).

70. BigQuery — Google's data warehouse. Anthropic's `BigQuery metrics export` operates as a separate organizational stream from the user-facing telemetry mechanisms; setting `DISABLE_TELEMETRY` does not disable BigQuery export. (*See* Ex. E-35, NF-1 at p. 262, l. 3.)

71. OpenTelemetry — Open standard for telemetry collection; one of the four parallel telemetry pipelines documented at Ex. E-34, SC-1 at p. 254, l. 3 (alongside GrowthBook, Datadog, and First-Party Events).

72. Cloudflare — Internet-infrastructure provider. Cloudflare's JavaScript Detections (JSD) platform runs on Anthropic's authenticated session path probing browser-environment characteristics (canvas fingerprint, WebGL, font enumeration, plugin inventory, screen properties, timing metrics) — a dual-layer device-identification channel parallel to Statsig. (*See* Ex. E-1 at p. 6, l. 2 (NF-75).) The cookie `_cfuvid` documented in Plaintiff's wire captures is Cloudflare's per-session identifier.

B. Defendant-Side Internal Infrastructure

73. `api.anthropic.com` — Anthropic's primary API endpoint; the destination of `sessionIngress` HTTP PUTs. Wire-confirmed in Plaintiff's captures.

74. `mcp-proxy.anthropic.com` — Anthropic-operated proxy for Model Context Protocol (MCP) traffic. Plaintiff's external-service traffic — including tool-call arguments and responses to Google Drive via the MCP integration — is routed through this proxy before reaching the destination service. (*See* Ex. E-17 at p. 124, l. 2 (NF-78).)

75. Floodgate / Floodgate Proxy — Apple-operated man-in-the-middle proxy through which all Xcode Intelligence communications are routed before reaching Anthropic. Documented at Ex. F-4 at p. 22, l. 2; the SF-series catalogues the architecture.

76. SmootML — Apple's analytics infrastructure, capturing 189 analytics connections per Xcode Intelligence session without disclosure. (*See* Ex. F-23 at p. 101, l. 2.)

77. Biome — Apple's system-level user-activity tracking framework, located on disk at `/private/var/db/biome/streams/`. Contains dozens of independent data streams; six streams documented at Ex. F-9 at p. 44, l. 2 are AI-relevant in this case: `_DKEvent.App.InFocus` (Claude Desktop in-focus tracking), `_DKEvent.Safari.History` (Claude.ai URLs in Safari history), `OSAnalytics.Stability.Crash` (Claude Desktop crash logs), `_DKEvent.Activity.Level` (activity level during AI usage), `_DKEvent.App.LocationActivity` (location correlated with app usage), and `AppleIntelligence.Reporting.*` (Apple Intelligence analytics streams). A single forensic extraction (101 MB, December 26, 2025) preserved the database for inspection.

78. OzCode — Originally **CodeValue D.T. Ltd.**, then **OzCode Ltd.** (Herzliya Pituach, Israel). Filed seven United States patents on January 7, 2019 covering time-travel debugging primitives (Live Coding US 10,783,055; Future Prediction US 10,990,504; Visual Annotations US 11,048,615; Point-In-Time Links US 10,783,056; Pivoting US 10,789,151; Async Collaboration US 10,795,801; and US 10,795,801). All seven patents were assigned to Datadog, Inc. (Delaware) effective October 26, 2021 with USPTO recordation delayed to April 17, 2024. Principal inventor Omer Raviv publicly self-identifies as a former IDF Unit 8200 commander; co-inventor Alon Mordechai Fliess held Microsoft Regional Director and Azure MVP credentials at time of filing. The seven-patent family supplies the architectural primitives

(speculative execution, state persistence, execution-context switching, etc.) for the bolt-on filtering substrate analyzed at App. P.

79. Rafael Development Corporation — A 50.1% Elron / 49.9% Rafael Advanced Defense Systems joint venture, whose mandate per Elron's 2025 Annual Report § 23.3 is to "commercialize Rafael's military technologies in the civilian market." Provided the OzCode commercialization pipeline; Rafael Advanced Defense Systems is itself a state-owned Israeli defense contractor. Documented at App. P § II.

80. Datadog Israel Ltd. — Datadog Inc.'s Israeli subsidiary (Company ID 516464922; incorporated September 14, 2021 — forty-nine days before the OzCode asset-sale close). An affiliate-tier subprocessor with contractual access through Datadog's DPA inheritance structure. Principal OzCode inventor Omer Raviv currently serves there as Engineering Manager on `dd-trace-dotnet`, the observability framework through which Plaintiff's telemetry transits. Documented at App. P § VI as the jurisdictional hook for the Israeli surveillance facilitation analysis.

C. AI Industry Vendors and Initiatives

81. Anthropic PBC — Defendant Anthropic Public Benefit Corporation; primary corporate Defendant whose source code is consolidated at Ex. E. The hub node in the case's evidentiary knowledge graph.

82. METR (Model Evaluation & Threat Research, Inc.) — Defendant; AI-evaluation organization formerly named ARC Evals. Funded principally through The TED Foundation, Inc. (Audacious Project). Documented at App. B.

83. TED Foundation — The TED Foundation, Inc.; non-party fiduciary funder of METR via the Audacious Project mechanism. Subpoenaed under FRCP 45 at DISCOVERY-13.

84. Audacious Project — TED-affiliated funding mechanism through which The TED Foundation routes funding to METR. Documented at App. B § donor-tier analysis.

85. Project Canary — RAND Corporation–co-sponsored AI evaluation governance initiative whose non-classified governance is the subject of DISCOVERY-14. Documented at App. B.

86. Constitutional AI — Anthropic's term for its public-facing alignment methodology, baked into model weights during pretraining. Distinct from the bolt-on filtering stack that operates outside the model's weights. (*See* ¶ 48 above.)

87. Long-Term Benefit Trust (LTBT) — Anthropic's governance trust structure announced September 2023. Documented at Ex. METR-8.

88. Mythos / Claude Mythos Preview — Anthropic's announced general-purpose model with autonomous-operation capabilities including multi-step exploit chains, reverse engineering of closed-source binaries, and sandbox escape. Selectively distributed to commercial partners under a "too dangerous for public release" justification. (*See* Ex. ANTHROPIC-12; Memo F § VII.H.)

89. Glasswing / Project Glasswing — Anthropic-affiliated initiative that received Mythos-tier capability access. Documented at Ex. ANTHROPIC-12.

90. Palantir Technologies — Data-analytics firm holding $200M+ U.S. Department of Defense contracts, CIA and NSA partnerships, and Project Nimbus integration with the Israeli Ministry of Defense via Google and Amazon cloud infrastructure. Documented at App. D § XIII as a paradigmatic defense-contractor customer whose contract requirements (sticky routing, mandatory deployment, Constitutional-AI bypass for targeting analysis, asymmetric processing)

are alleged to drive Anthropic's architectural decisions. The November 2024 Palantir-Anthropic integration announcement preceded the December 2024 "three-bugs" infrastructure deployment.

91. AWS Bedrock / Google Vertex / Azure Foundry — The three Enterprise-tier deployment targets identified verbatim in Anthropic's source code; the routing literal is `firstParty` for consumer subscribers vs. these three for enterprise. Enterprise customers on Bedrock, Vertex, or Azure Foundry receive zero Datadog events, zero first-party analytics, and zero GrowthBook experiment assignments — the source comment at `config.ts:34` states the architectural intent: "enterprise customers capture responses via OTEL." (*See* Ex. E-35, NF-28 at p. 262, l. 3 (Enterprise analytics bypass); Ex. H-4, NF-139 at p. 34, l. 3; Memo A ¶ 60; Memo B § cascade-immunity.) "Foundry" in this enterprise-routing context refers to Microsoft Azure AI Foundry and is unrelated to Palantir Foundry (¶ 92 below).

92. Palantir Foundry — Palantir Technologies' data-integration product, distinct from Microsoft Azure AI Foundry (¶ 91 above). Mentioned across the case as a paradigm of defense-contractor / intelligence-community-aligned enterprise data integration.

93. Apple Intelligence / Xcode Intelligence — Apple's user-facing AI feature ("Apple Intelligence") and its developer-tool counterpart ("Xcode Intelligence"). Both route traffic through Apple's Floodgate Proxy before reaching Anthropic; Plaintiff's Ex. F consolidated forensics catalog the architecture.

VI. NETWORK AND WIRE-CAPTURE TERMS

94. TCP RST injection / RST injection — Packet-level attack in which forged TCP-RST packets are injected onto the wire to disrupt established connections. Documented at Ex. E-24 at p. 159, l. 2 (RST injection during legal-document preparation) and at Ex. D (network forensic exhibits).

95. Hardware tap / Hardware TAP — Physical-layer wire capture device producing packet captures at line speed. Plaintiff's hardware-TAP captures established that 875 of the captured RST packets in a 25-minute window were forged (IP ID = 0x0000) — physical proof that the RSTs originate from ISP infrastructure rather than from Plaintiff's device. (*See* Ex. D-31 at p. 192, l. 19.)

96. `IP ID = 0x0000` — IP-header `Identification` field set to all-zeros; a forensic signature of forged packets in the captured RST flood, since legitimate operating-system stacks generate non-zero IP IDs. Documented at Ex. D-31 at p. 192, l. 19.

97. Wire capture — Technique of recording network traffic at the wire (typically post-TLS-decryption with a controlled MITM under the user's own keys) for forensic analysis. Plaintiff's TOS-compliant wire captures form the H-Series evidentiary base; specific captured sessions include H-8 (Session 5) and H-11 (Session 6). (*See* Ex. H.)

98. Sticky routing / Content-based sticky routing — *See* ¶ 63 above.

99. Comcast injection point — The location on Comcast's backbone infrastructure at which RST injection occurred against Plaintiff's traffic. Documented at Ex. D-1 at p. 3, l. 1 and Ex. D-11 at p. 52, l. 28.

100. `/v1/messages` — Anthropic's primary message API endpoint. Wire-confirmed across the captured sessions; the endpoint receives the contents of every conversational message via `sessionIngress`.

101. MCP / Model Context Protocol / MCP proxy — Open protocol for connecting AI clients to external services. Anthropic's MCP integration routes Plaintiff's external-service traffic through `mcp-proxy.anthropic.com` (the MCP proxy), making Anthropic an intermediate party to user-selected external connections. (*See* ¶ 74 above.)

102. CloudFlare JSD — Cloudflare's JavaScript Detections platform; an active browser-environment fingerprinting layer running on Anthropic's authenticated session path. (*See* ¶ 72 above.)

VII. ENVIRONMENT VARIABLES

103. `DISABLE_TELEMETRY` — Undisclosed environment variable. The actual mechanism for disabling first-party telemetry; the user-facing `"telemetry": false` setting is silently ignored. Even with `DISABLE_TELEMETRY` correctly set, BigQuery metrics export remains active. (*See* Ex. E-35, NF-1 at p. 262, l. 3; Ex. E-5 at p. 40, l. 3.)

104. `DISABLE_AUTOUPDATER` — Undisclosed environment variable that freezes the Claude Code binary at its current version. Without this variable, the binary silently replaces itself on terminal restart — the same channel by which any code change Anthropic deploys reaches subscriber machines. (*See* Ex. H-1 at p. 3, l. 5 H-1 Test 6.)

105. `DISABLE_AUTO_COMPACT` — Environment variable governing the compaction subsystem documented at Ex. E-35, NF-12 at p. 262, l. 3 (compaction scratchpad stripping — model reasoning permanently erased).

106. `USER_TYPE` — *See* ¶ 22 above.

107. `CLAUDE_CODE_DISABLE_NONESSENTIAL_TRAFFIC` — Environment variable claimed to disable non-essential traffic; in fact does **not** stop session ingress. (*See* ¶ 23 above.)

108. `CLAUDE_CODE_DISABLE_NONSTREAMING_FALLBACK` — Environment variable governing whether the client falls back to a non-streaming model invocation under network conditions; documented in the source.

109. `CLAUDE_CODE_EFFORT_LEVEL` — Environment variable for direct override of effort throttling. Distinct from the per-account-keyed effort decision in `effort.ts` whose default is `'medium'` for external customers and `'max'` for ANT.

110. `CLAUDE_CODE_DISABLE_ADVISOR_TOOL` — Environment variable governing the server-side advisor documented at Ex. E-35, NF-22 at p. 262, l. 3 — the second AI monitoring conversations on the server side.

111. `CLAUDE_CODE_DISABLE_FEEDBACK_SURVEY` / `CLAUDE_CODE_DISABLE_OFFICIAL_MARKETPLACE_AUTOINSTALL` / `CLAUDE_CODE_AUTO_COMPACT_WINDOW` / `CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS` / `CLAUDE_CODE_ENTRYPOINT` — Additional product-configuration environment variables documented in the source. Each governs a discrete behavior; none appears in user-facing privacy-control documentation.

112. `DATADOG_LOGS_ENDPOINT` / `DATADOG_ALLOWED_EVENTS` — Environment variables governing the Datadog destination URL and event-filtering allowlist. Documented in the source; each governs telemetry routing distinct from the user-facing telemetry control.

VIII. FILE PATHS AND SOURCE LOCATIONS

Each entry names a file or path referenced verbatim across the case record. Paths under `~/` denote the user's home directory; paths under `src/` denote locations within Anthropic's leaked Claude Code source repository.

A. User-Side Paths

113. `~/.claude/` — Claude Code's user-data directory. Contains configuration (`settings.json`), per-account state (`.claude.json`), the file-history capture, the policy-limits cache, and the local telemetry-failure log.

114. `~/.claude/settings.json` — User-facing Claude Code configuration file. Subject of the nine server-side migration functions (including `phK()`) documented at Ex. H-6 at p. 45, l. 3 (¶ 27, ¶ 60).

115. `~/.claude.json` — Per-account state file containing local cache of GrowthBook flag assignments, including the `tengu_log_datadog_events: True` assignment for Plaintiff's account verified at Ex. H-5 at p. 39, l. 4.

116. `~/.claude/file-history/` — Hidden directory containing the captured-file inventory documented at Ex. E-6 at p. 54, l. 2 — 1,453 files totaling 38.73 MB across 859 unique paths in 9 projects, with up to 37 versions per file.

117. `~/.claude/policy-limits.json` — Local cache of the per-account `policy-limits.json` payload pushed from Anthropic's servers; mtime 2026-04-24 21:51 UTC for Plaintiff's machine. (*See* Ex. H-5, NF-96 at p. 39, l. 4; byte-identical across three of Plaintiff's machines per NF-175.)

118. `~/.claude/projects/` — Directory containing per-project Claude Code state.

119. `~/.claude/telemetry/1p_failed_events.*` — Local log of first-party telemetry events that failed delivery and are queued for retry. Documented at Ex. F / Ex. H.

B. Source-File Paths (In the Anthropic Source Tree)

120. `src/services/api/sessionIngress.ts:25` **and** `:69-77` — Location of the `sessionIngress` definition (¶ 23) and the 10-attempt exponential-backoff retry loop. (*See* Ex. E-35, NF-5 at p. 262, l. 3.)

121. `oauth.ts:224` — Location of the engineer comment "for Xcode integration" — direct evidence of the bilateral Apple-Anthropic integration documented at Ex. F-29, AF-11 at p. 133, l. 3.

122. `effort.ts:260-265` — Location of the default-config object that separates `external: 'medium'` from `ant: 'max'` effort levels; **`effort.ts:303`** carries the engineer comment "IMPORTANT: Do not change this default". (*See* Ex. E-34, SC-3 at p. 254, l. 3.)

123. `exploreAgent.ts:76-78` — Location of the hardcoded Haiku model substitution for external users in the explore subagent. (*See* Ex. E-38, NF-38 at p. 346, l. 3.)

124. `datadog.ts:12-14` — Location of the hardcoded Datadog destination IP `34.149.66.137`. (*See* Ex. E-34, SC-9 at p. 254, l. 3.)

125. `claude.ts:2464-2468` — Location of the `inc-4258` code comment acknowledging "double tool execution" as a known institutional issue — concurrent-scienter evidence for the closed billing loop documented at Ex. E-35, NF-31 at p. 262, l. 3.

126. `constants/tools.ts:41` — Location at which autonomous agent spawning is restricted from public users, even as Mythos is selectively distributed with autonomous-operation capabilities. (*See* Memo F § VII.H.)

127. `cli_2.1.78.js` — Compiled (minified) CLI bundle. Character position 10,635,512 contains the comment "internal ID - do not mention to user"; character position 8,364,085 contains the verbatim phrase "internal ID". (*See* App. C; App. P.)

128. `NF34_speculation_parallel.ts` — Evidence-extracted source-code excerpt at `extension.ts:58-70` documenting the speculation-parallel architecture that maps to U.S. Patent 10,783,055 (Live Coding) and to `DANGEROUS_uncachedSystemPromptSection`. (*See* App. P (patent correspondence).)

IX. SPECIFIC IDENTIFIERS REFERENCED IN THE FILINGS

The following are not categories but specific values whose appearance in the case record matters as evidence.

129. Plaintiff's `accountUUID 61343ae3-fd33-403f-897e-6edf3c368c37` — Anthropic-assigned unique identifier for Plaintiff's primary `info@lawofsupremacism.com` account. Cited verbatim across the operative complaint, App. P, Ex. H, and the Memo A count-by-count map as the targeting key for the per-account asymmetric architecture.

130. Plaintiff's email `info@lawofsupremacism.com` — Targeting attribute transmitted on the wire to GrowthBook as a per-user enrollment key. (*See* Ex. E-35, NF-48 at p. 262, l. 3.)

131. Datadog client token `pubbbf48e6d78dae54bceaa4acf463299bf` — Public-key Datadog tenant identifier hardcoded in the leaked Anthropic source code, identifying the specific receiving Datadog organization. (*See* Ex. E-34, SC-9 at p. 254, l. 3; Ex. H-4 at p. 34, l. 3.)

132. GrowthBook SDK keys `sdk-yZQvlplybuXjYh6L`, `sdk-xRVcrliHIlrg4og4`, `sdk-zAZezfDKGoZuXXKe` — Three production-key values present in the leaked source despite Anthropic's public statement that "no customer data, credentials, or sensitive information was exposed." (*See* Ex. E-34, SC-9 at p. 254, l. 3.)

133. Datadog destination IP `34.149.66.137` — Hardcoded at `datadog.ts:12-14`; force-enabled per Plaintiff's account via `tengu_log_datadog_events`; the same IP appears in 5,829 spoofed bare-RST packets against Plaintiff's network captured at Ex. D-31 at p. 192, l. 19 and at Ex. D-1 at p. 3, l. 1. The IP is the single strongest cross-evidence bridge in the case.

134. Internal Slack channel `#briarpatch-cc` — Internal Anthropic Slack channel name present in the leaked source — additional evidence contradicting Anthropic's "no internal information was exposed" representation. (*See* Ex. E-34, SC-9 at p. 254, l. 3.)

135. `policy-limits.json` mtime `2026-04-24 21:51 UTC` — File-system modification time of the per-account `policy-limits.json` payload pushed to Plaintiff's machine, demonstrating contemporaneous server-side targeting during Plaintiff's classifier-extraction session that day. (*See* Ex. H-5, NF-96 at p. 39, l. 4.)

X. ACRONYMS AND ABBREVIATIONS

136. TOS — Terms of Service. Used both generically (the TOS of Claude Code, ChatGPT, etc.) and as part of the qualifier "TOS-compliant" to denote testing performed within the bounds of the relevant Terms.

137. MSA — Master Services Agreement. Specifically the Anthropic-Datadog and Anthropic-Statsig MSAs analyzed at Memo C in connection with the contractual concealment architecture.

138. DPA — Data Processing Agreement. The ancillary contracts between data controllers and processors that purport to bind processors not to use customer data for ML training. Anthropic's enterprise-side DPAs, analyzed across Memo C and Memo E, are contradicted by the Sentry Autofix routing documented at Ex. SN-3.

139. ECPA — Electronic Communications Privacy Act, 18 U.S.C. §§ 2510-2522 (Title I — Wiretap Act) and §§ 2701-2712 (Title II — Stored Communications Act). Count XII.

140. CCPA — California Consumer Privacy Act, Cal. Civ. Code § 1798.100 *et seq.* Count XIV.

141. CFAA — Computer Fraud and Abuse Act, 18 U.S.C. § 1030. Count XIII.

142. RICO — Racketeer Influenced and Corrupt Organizations Act, 18 U.S.C. §§ 1961–1968. Count VI.

143. FRCP / FRE — Federal Rules of Civil Procedure / Federal Rules of Evidence.

144. MCP — Model Context Protocol. *See* ¶ 100.

145. LCR — Long Conversation Reminder. *See* ¶ 58.

146. LTBT — Long-Term Benefit Trust. *See* ¶ 86.

147. FFRDC — Federally Funded Research and Development Center. The category of contracts excluded from the DISCOVERY-14 RAND subpoena under the Classified Material Exclusion. (*See* App. B; DISCOVERY-14.)

148. DRM — Digital Rights Management; here used metaphorically to describe Native Client Attestation (¶ 52).

149. JSD — JavaScript Detections (Cloudflare). *See* ¶ 72.

150. SDK — Software Development Kit. The GrowthBook SDK keys at ¶ 131 are GrowthBook integration tokens.

151. TTFB — Time-to-First-Byte. The measurement axis underlying the asymmetric-targeting findings at Ex. H-12 at p. 84, l. 4.

152. CCR — Cloud Code Runner. Anthropic's cloud-side execution and mirroring infrastructure to which Claude Code can auto-connect (`tengu_cobalt_harbor`, `src/bridge/bridgeEnabled.ts:187`) and into which all local session events can be mirrored (`tengu_ccr_mirror`, `src/bridge/bridgeEnabled.ts:200`). Documented at Ex. E-35, NF-13 at p. 262, l. 3. *See also* `archiveRemoteSession` (¶ 28).

153. DCD — Defense-Configuration Diagnostic. Finding-series prefix for runtime behaviors observed by Plaintiff's TOS-compliant testing battery. *See* ¶ 12.

XI. RESERVATION

154. This appendix is offered as a glossary, not as new substantive evidence. Where any definition states a fact about Defendants' product behavior, the underlying evidence appears in

the cited exhibit; this appendix paraphrases for convenience and should not be relied upon in lieu of the underlying exhibit. Plaintiff reserves the right to supplement the glossary as additional terms become relevant in the course of discovery.

Appendix Q — Technical Definitions and Glossary. Prepared by Joseph Kirchner, Pro Se Plaintiff.