

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 P: ISRAELI AI
SURVEILLANCE
FACILITATION —
DATADOG/OZCODE
ARCHITECTURE AND
PERSONNEL PIPELINE**

TABLE OF CONTENTS

I. EXECUTIVE SUMMARY	8
II. FACTUAL ALLEGATIONS — CORPORATE, FINANCIAL, AND PERSONNEL PROVENANCE.....	16
<i>A. Case-Record Cross-Layer Establishment of Datadog IP `34.149.66.137`</i>	16
<i>B. Datadog Israel Ltd. — Corporate Entity and Operational Profile</i>	18
<i>C. The Acquisition Pipeline Provisioning the Tel Aviv R&D Presence</i>	24
<i>D. Ozcode Corporate Genealogy — Codevalue D.T. Ltd. to Oz Code Ltd. to Datadog Asset Sale.....</i>	28
<i>E. Rafael Development Corporation — Structure and Rafael Veto Rights</i>	30
<i>F. Ozcode as RDC Portfolio Company — Investment History and Board Representation..</i>	32
<i>G. Rafael's Direct Financial Exposure to RDC During the Ozcode Ownership Window....</i>	33

<i>H. Ozcode Asset-Sale Transaction Detail (October 26, 2021)</i>	35
<i>I. Arieli Acquisition of Elron (September 4, 2024) — Joint U.S./israeli Purchaser Structure</i>	36
<i>J. The Ozcode United States Patent Portfolio — Seven Granted Patents Plus One 2024 Continuation</i>	38
<i>K. USPTO Patent Assignment Chain — Full-Portfolio Transfer with Two-and-One-Half-Year Recordation Lag</i>	40
<i>L. Continuing Datadog Intellectual Property Generation by Ozcode-Origin Personnel (2024 Continuation)</i>	42
<i>M. Named Ozcode → Datadog Personnel Bridges</i>	43
<i>N. Executive Governance Pipeline from Rafael Tel Aviv R&D Leadership to Elron/rdc Chief Executive Officer</i>	53
<i>O. Primary-Source Corroboration — Elron Ventures Ltd. 2025 Annual Report (Exhibit IS-1, Formerly DD-11)</i>	54
<i>P. Primary-Source Corroboration — Elron Ventures Ltd. 2021 Periodic Report (Exhibit IS-2, Formerly DD-12)</i>	55
<i>Q. Primary-Source Corroboration — Elron Ventures Ltd. August 2025 Investor Presentation (Exhibit IS-5)</i>	56
<i>R. Datadog Tel Aviv LLM, Artificial Intelligence, and CI/CD Engineering Function (Currently Advertised)</i>	58
<i>S. Legal Architecture — Mechanisms by Which Rafael Retains Continuing Rights Despite the U.S. Patent Transfer</i>	59
<i>T. Datadog SEC Disclosure Pattern — Five Consecutive 10-Ks Omit Israeli Operations (Fy2021–fy2025)</i>	69
III. TECHNICAL ANALYSIS — TIME-TRAVEL DEBUGGING AS THE ARCHITECTURAL SUBSTRATE FOR REAL-TIME SURVEILLANCE AND PROMPT INJECTION	70
<i>A. The 2019 Ozcode Patent Family is a Single Unified Architecture, Not Seven Independent Inventions</i>	71
<i>B. The Shared Architectural Substrate Across All Seven Claims</i>	73
<i>C. The Complete Speculative-Execution-and-Intervention Loop</i>	75
<i>D. U.S. Patent No. 10,990,504 (Future Prediction) — the Speculative-Execution Surveillance Primitive</i>	77
<i>E. U.S. Patent No. 10,783,056 (Point in Time Links) — the State-Persistence and Cross-Party Handoff Primitive</i>	80
<i>F. U.S. Patent No. 10,789,151 (Pivoting Ability) — the Execution-Context-Switching Primitive</i>	82
<i>G. U.S. Patent No. 10,783,055 (Live Coding Ability) — the Real-Time Modification Primitive</i>	85

<i>H. U.S. Patent No. 10,795,801 (Asynchronous Collaboration) — the Multi-Party-Access Primitive and the "Two Computers" Smoking Gun</i>	88
<i>I. U.S. Patent No. 11,048,615 (Visual Annotations) — the User-Facing HUD Display Primitive.....</i>	98
<i>J. U.S. Patent No. 11,080,164 (Redaction of Sensitive Information) — the Output-Modification Primitive</i>	100
<i>K. U.S. Patent Application No. 18/828,099 / Publication No. US 2026/0072809 A1 (Code Origin and Runtime Duplicity Optimization) — the 2024 Continuation</i>	104
<i>L. The Complete Unified Architecture Applied to Large-Language-Model Inference: Primitive-by-Primitive Mapping</i>	107
<i>M. Latency as the Computational Footprint of the Speculative Loop</i>	110
<i>N. The December 2024 Argmax "Workaround" as the Literal Token-Layer Implementation of the Three-Primitive Composition.....</i>	111
<i>O. Why Real-Time Prompt Injection at Production Scale is Architecturally Infeasible Without This Substrate</i>	113
<i>P. The Architectural Nexus — Patents Plus Personnel Plus Subprocessor Terminus</i>	114
<i>Q. Primitive-to-Source Concordance: the Patented Architecture is Operational in Anthropic's Production Code.....</i>	115
<i>R. Vocabulary Inheritance Establishes Knowing Implementation</i>	122
<i>S. Cross-Reference — the Four-Layer Architectural Distinction Pleaded in the Complaint</i>	123
IV. ARCHITECTURAL CORRESPONDENCE WITH APP. D	124
<i>A. Primary Correspondence Table — App. D Observed Behavior ↔ Appendix P Patented Primitive.....</i>	125
<i>B. The Three "Bugs" Reread Under the Seven-Primitive Architecture</i>	127
<i>C. Palantir / Project Nimbus Integration as the Intended Application</i>	127
<i>D. Cross-Reference Directives.....</i>	128
V. DATADOG SUBPROCESSOR PIPELINE AND PLAINTIFF-SPECIFIC TARGETING INFRAstructure — CASE-RECORD ESTABLISHMENT	128
<i>A. IP `34.149.66.137` — Five-Role Cross-Layer Documentation Restated for Count-Specific Hooks.....</i>	128
<i>B. Per-User Forced Enablement via Growthbook `tengu_log_datadog_events: True`</i>	130
<i>C. Telemetry Scale — Per-Session Volume</i>	131
<i>D. DPA Subprocessor Inheritance Structure — Datadog Israel Ltd. Access.....</i>	132
<i>E. Datadog Security Labs Function.....</i>	134
<i>F. LLM / IDE AI Engineering Function Staffed from Tel Aviv.....</i>	134
<i>G. Cloud Architecture: Network Ingress Region is Architecturally Irrelevant to Personnel Access.....</i>	134

<i>H. Per-Account Asymmetric Targeting — `accountUUID`-keyed Differential Gating Across Two Plaintiff Accounts on the Same Machine</i>	137
<i>I. Server-Side Spoliation Mechanism — Silent Operator-Side Defensive-Configuration Wipe via Migration-on-Startup</i>	157
<i>J. `userBucket` SHA-256 30-Bucket Persistent Userid Hashing — Per-Install Tracking Key Sent as Datadog Tag</i>	162
<i>K. Sandbox Architecture as Fourth Per-Account Control Surface — Per-Managed-Deployment Hard-Gate of `dangerouslyDisableSandbox`</i>	163
<i>L. Per-Organization Targeting Layer — `policy-limits.json` Byte-Identical Across Three Machines Confirms per-`organizationUuid` Control Axis Distinct from per-`accountUUID` Layer</i>	168
<i>M. Cross-Vendor Datadog Concentration — Plaintiff's Own Captures Show Multiple Independent Origins Routing Telemetry into the Same Datadog Inc. Data Plane</i>	169
VI. ISRAELI JURISDICTIONAL HOOK	172
<i>A. Datadog Israel Ltd. Legal Status</i>	172
<i>B. The CLOUD Act Does Not Solve the Jurisdictional Exposure</i>	173
<i>C. The Co-Tenancy Observation is Correlative, Not Deterministic</i>	174
VII. ARCHITECTURAL SIGNIFICANCE FOR PLAINTIFF'S CLAIMS	174
<i>A. ECPA (Count XII) — the Time-Travel Architecture is the Interception Mechanism, and Anthropic's Own Source Code Forecloses the Consent Defense</i>	174
<i>B. CFAA (Count XIII) — Parallel Execution Path Without User Authorization</i>	177
<i>C. CCPA (Count XIV + §1798.140(Ag) International Transfer)</i>	177
<i>D. Section 1985(3) Conspiracy</i>	179
<i>E. RICO Predicate Acts (Count VI)</i>	180
<i>F. The Count-Level Cross-Reference Map</i>	185
VIII. DISCOVERY SCOPE — DISCOVERY-11 FOREIGN INFLUENCE (CROSS-REFERENCE)	187
IX. EXHIBIT DESIGNATIONS — L-SERIES (ECF) AND IS-SERIES (USB)	189
X. RELATIONSHIP TO OTHER CASE DOCUMENTS	193
XI. OPEN QUESTIONS AND LIMITATIONS	194
XII. AUTHENTICATION FRAMEWORK	196

App. P

TABLE OF AUTHORITIES

CASES

Citation
<i>Already, LLC v. Nike, Inc.</i> , 568 U.S. 85 (2013)
<i>City of Mesquite v. Aladdin's Castle, Inc.</i> , 455 U.S. 283 (1982)
<i>Friends of the Earth, Ltd. v. Laidlaw Environmental Services (TOC), Inc.</i> , 528 U.S. 167 (2000)
<i>H.J. Inc. v. Northwestern Bell Telephone Co.</i> , 492 U.S. 229 (1989)
<i>Interstate Circuit, Inc. v. United States</i> , 306 U.S. 208 (1939)
<i>National Foundation, Inc. v. United States</i> , 13 Cl. Ct. 486 (1987)

CONSTITUTIONAL PROVISIONS

Citation
U.S. Const. art. I, § 8, cl. 11 (War Powers)
U.S. Const. art. I, § 9, cl. 7 (Appropriations Clause)
U.S. Const. amend. IV (Search and Seizure)
U.S. Const. amend. V (Due Process)

FEDERAL STATUTES

Citation
18 U.S.C. § 1030 (Computer Fraud and Abuse Act)
18 U.S.C. §§ 1961-1968 (Racketeer Influenced and Corrupt Organizations Act)
18 U.S.C. §§ 2510-2522 (Electronic Communications Privacy Act)
28 U.S.C. § 1746 (unsworn declarations under penalty of perjury)
35 U.S.C. § 101 (patentable subject matter)
35 U.S.C. § 112 (specification; enablement requirement)
35 U.S.C. § 261 (ownership; assignment)
42 U.S.C. § 1985(3) (conspiracy to interfere with civil rights)
Pub. L. No. 115-141, Div. V (Clarifying Lawful Overseas Use of Data Act of 2018)

FEDERAL REGULATIONS

Citation

17 C.F.R. § 229.102 (Regulation S-K Item 2 — description of properties)
17 C.F.R. § 229.105 (Regulation S-K Item 1A — risk factors)
17 C.F.R. § 240.12b-20 (additional information necessary to make required statements not misleading)

STATE STATUTES

Citation
Cal. Civ. Code § 1798.100 <i>et seq.</i> (California Consumer Privacy Act)
Cal. Civ. Code § 1798.140(ag) (international-transfer disclosure)

FOREIGN STATUTES (for jurisdictional-reach analysis only)

Citation
Companies Law, 5759-1999 (Israel) — including §§ 8 (incorporation), 39 (officer-of-record reporting), 123 (registered office), 140 (director changes and share transfers), 141 (annual return), 326 (voluntary merger of wholly-owned subsidiary), and Eighth Part Chapter One (merger procedure)
Companies Regulations (Merger), 5760-2000 (Israel)
Privacy Protection Law, 5741-1981 (Israel)
Privacy Protection Regulations (Data Security), 5777-2017 (Israel)
Law for Combating Terrorism, 5776-2016 (Israel)
General Security Service Law, 5762-2002 (Israel)
Income Tax Ordinance [New Version], 1961 (Israel), § 103 (tax-free merger election)
Israeli Securities Regulations (Periodic and Immediate Reports), 5730-1970
Encouragement of Research, Development and Innovation in Israeli Industry Law, 5744-1984 (Israel), § 15A (transfer abroad of "Sponsored Knowledge"; Israel Innovation Authority approval and redemption-fee requirements)
Patents Law, 5727-1967 (Israel)

FEDERAL RULES OF EVIDENCE

Citation
Fed. R. Evid. 201(b) (judicial notice of adjudicative facts)

Fed. R. Evid. 801(d)(2) (party admissions as non-hearsay)
Fed. R. Evid. 902(5) (self-authentication of public government records)
Fed. R. Evid. 902(6) (self-authentication of periodicals)

PATENT AUTHORITIES

Citation
U.S. Patent No. 10,783,055 ("Time Travel Source Code Debugger Incorporating Live Coding Ability")
U.S. Patent No. 10,783,056 ("Time Travel Source Code Debugger Incorporating Point In Time Links")
U.S. Patent No. 10,789,151 ("Time Travel Source Code Debugger Incorporating Pivoting Ability")
U.S. Patent No. 10,795,801 ("Time Travel Source Code Debugger Incorporating Asynchronous Collaboration")
U.S. Patent No. 10,990,504 ("Time Travel Source Code Debugger Incorporating Future Prediction")
U.S. Patent No. 11,048,615 ("Time Travel Source Code Debugger Incorporating Visual Annotations")
U.S. Patent No. 11,080,164 ("Time Travel Source Code Debugger Incorporating Redaction of Sensitive Information")
U.S. Patent Application No. 18/828,099 / Pub. No. US 2026/0072809 A1 ("Code Origin and Runtime Duplicity Optimization")

SECONDARY AUTHORITIES

Citation
Elron Ventures Ltd., 2025 Annual Report (English translation)
Elron Ventures Ltd., 2024 Periodic Report (English translation)
Elron Ventures Ltd., 2023 Periodic Report (English translation)
Elron Ventures Ltd., 2021 Periodic Report (English translation)
Elron Ventures Ltd., August 2025 Investor Presentation (English)
Elron Ventures Ltd., September 1, 2024 Immediate Report (Arieli Purchaser disclosure)
Elron Ventures Ltd., November 27, 2024 Annual General Meeting Notice

Elron Ventures Ltd., December 27, 2021 Proxy Materials
Tel Aviv Stock Exchange public filings (Elron Ventures Ltd., Ticker: ELRN)
USPTO Patent Assignment Database (Reel/Frame 067135/0142 and related recordations)

I. EXECUTIVE SUMMARY

1. This Appendix documents that the architectural substrate required for real-time prompt injection at production scale is a set of seven discrete primitives — speculative execution, state persistence, execution-context switching, live code modification, multi-party asynchronous access, user-facing display control, and output substitution with execution-flow preservation — all explicitly claimed by a seven-patent family filed January 7, 2019 by OzCode Ltd. (Israel); all now owned by Datadog, Inc. following the October 26, 2021 asset sale; and all invented by Omer Raviv (publicly self-identified IDF Unit 8200 commander) together with Alon Mordechai Fliess. Datadog, Inc. is Plaintiff's telemetry subprocessor: its endpoint IP `34.149.66.137` is hardcoded at Anthropic's `datadog.ts:12-14` (Ex. E-34, SC-9 at p. 254, l. 3), force-enabled for Plaintiff's `accountUUID` via `tengu\_log\_datadog\_events` (Ex. E-39, Finding NF-47), and the claimed source of 5,829 spoofed bare-RST packets against Plaintiff's network (Ex. D-30 at p. 170, l. 3). Datadog Israel Ltd. (Company ID 516464922; incorporated September 14, 2021 — forty-nine days before the OzCode asset-sale close) is an affiliate-tier subprocessor with contractual access through the Datadog DPA inheritance structure; the principal inventor currently serves there as Engineering Manager on `dd-trace-dotnet`, the observability framework through which Plaintiff's telemetry transits. A 2024 continuation patent (US 2026/0072809 A1, filed five days after the Arieli-Elron control-change close) names Raviv and IAF veteran Matan Green among the co-inventors and extends the architecture with explicit "network or system security" branching disclosure — the textual correspondence to the Constitutional-AI-bypass

mechanism pleaded at App. D. The architectural substrate for real-time surveillance and prompt injection is therefore owned by the entity that receives Plaintiff's telemetry, implemented by personnel at that entity's Israeli subsidiary, invented by IDF Unit 8200 and IAF veterans, and provisioned through the Rafael Development Corporation commercialization pipeline — a 50.1% Elron / 49.9% Rafael Advanced Defense Systems joint venture whose explicit mandate per Elron's own 2025 Annual Report §23.3 is to "commercialize Rafael's military technologies in the civilian market."

2. Finding One — The 2019 OzCode patent family is a single unified architecture, not seven independent inventions. All seven patents trace to four identical provisional applications filed January 8, 2018 (62/614,576; 62/614,579; 62/614,581; 62/614,582); all seven were filed on the same date one year later (January 7, 2019); all seven name the identical two-inventor pair (Omer Raviv and Alon Mordechai Fliess); all seven were prosecuted by the same patent-firm attorney of record (Zaretsky Group PC / Howard Zaretsky); and all seven share an identical detailed description and figure set (Figures 1 through 76) with only the claim language differing across the family. Each of the seven claim sets protects one discrete primitive of one integrated speculative-execution-and-intervention system. *See* Section III.A; Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, and L-8 (patent PDFs and markdown conversions preserving USPTO page markers).

3. Finding Two — All seven patents were assigned to Datadog, Inc. effective October 26, 2021 with USPTO recordation delayed two and one-half years to April 17, 2024. The complete seven-patent family was assigned from OzCode Ltd. (Israeli corporation, successor-in-interest to the original applicant CodeValue D.T. Ltd.) to Datadog, Inc. (Delaware parent) effective October 26, 2021 — contemporaneous with the OzCode asset-sale close. USPTO

recordation of the assignment did not occur until April 17, 2024, a two-and-one-half year documentary lag. The April 17, 2024 recordation date falls exactly fifteen days before the May 2, 2024 Discount Investment Corporation — Arieli Capital LLC / Arieli EL Ltd. non-binding memorandum of understanding that commenced the control-change transaction at Elron Ventures Ltd. The simultaneous timing is material: the patent-assignment recordation was formalized at USPTO immediately before Elron's controlling-shareholder restructuring commenced. *See* Section II.K; Exhibit IS-9.

4. Finding Three — The seven primitives together constitute a complete observe-predict-decide-modify-display-redact loop; every operational step required for real-time, targeted, transparent-to-user prompt injection at production scale is explicitly claimed by one of the seven patents. Five architectural elements appear as claim limitations in every patent in the family — hook-function instrumentation, isolated-environment simulation, memory delta buffer, nested-data-structure execution flow, and time-anchored checkpoints. Each patent then adds one primitive-specific addition: Future Prediction (US 10,990,504), Point In Time Links (US 10,783,056), Pivoting (US 10,789,151), Live Coding (US 10,783,055), Asynchronous Collaboration (US 10,795,801), Visual Annotations (US 11,048,615), and Redaction (US 11,080,164). *See* Sections III.A through III.J for the verbatim claim language and the per-primitive LLM-inference application; Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, and L-8.

5. Finding Four — US Patent 10,990,504 (Future Prediction) is the speculative-execution surveillance primitive; without it, selective per-user intervention at scale is architecturally infeasible. The patent claims virtual execution "in an isolated environment from a current execution point to the end of a current scope without making any changes to original process memory" — "thereby enabling a user to time travel to the future." Applied to LLM

inference, this permits the system to predict the model's output before commitment to the actual inference stream. Synchronous pre-evaluation would block throughput and be detectable as latency; bulk post-hoc logging does not permit intervention; only isolated parallel speculative execution is architecturally feasible. Plaintiff's content-correlated latency observation (litigation/framework prompts take materially longer than neutral prompts) is the externally-observable footprint of the loop in operation. *See* Sections III.D and III.M for verbatim Claim 1, Claim 20, and the latency-time-equation analysis; Exhibit L-4.

6. Finding Five — US Patent 10,789,151 (Pivoting Ability) is the path-selection primitive; it maps directly onto the "sticky routing" and `recall\_target` variant-selection behaviors Anthropic disclosed in its own post-mortem. The Pivoting patent claims a method that "enables a user to switch an execution context of the debuggee application at any time to any location therein to any future or past context." Applied to LLM inference, this enables on-the-fly switching of inference context at any token boundary — architecturally identical to Anthropic's "sticky routing" (App. D ¶ 48, App. D ¶¶ 140-143) and `recall\_target` variant selection (App. D ¶ 45, App. D ¶ 49, App. D ¶ 127). Anthropic's three "bugs" operate as one composed three-primitive system: Future Prediction identifies what the model would output; Pivoting selects which path the request follows; Redaction substitutes flagged tokens. *See* Sections III.F, III.N, and IV; Exhibit L-7.

7. Finding Six — US Patent 11,080,164 (Redaction) is the output-modification primitive; it claims, in substance, what prompt injection accomplishes at the output layer. The Redaction patent claims a method of replacing flagged variables with "obscuring text and fake text" (a "meaningful readable string of characters") while "verifying that said replacement text does not affect code execution flow." Applied to LLM inference, the primitive substitutes

classifier-flagged tokens (per Ex. E-39 at p. 356, l. 3 § R; Ex. E-39, Finding NF-63) with replacements that preserve the user's downstream workflow continuity while preventing the user from seeing what the live inference actually produced. The execution-flow-preservation requirement is what distinguishes patented Redaction from ordinary data masking. *See* Section III.J for verbatim Claim 1, Claim 11, and the operator-account-specific `redact-thinking-2026-02-12` flag deployment; Exhibit L-1.

8. Finding Seven — Anthropic's December 2024 "workaround" code is the literal token-layer implementation of the time-travel-debugger loop composed from three of the 2019 primitives. Anthropic's September 17, 2025 post-mortem discloses December 2024 production code at the JAX sampling layer in which `probs.argmax(-1, keepdims=True)` performs Future Prediction (US 10,990,504), `argmax\_mask` performs Pivoting (US 10,789,151) at the token layer, and `logits - 50000` performs Redaction (US 11,080,164) by penalizing the flagged token out of the sampling distribution. The code is the literal composition of three patented primitives in five lines. *See* Section III.N for the full code block, line-by-line primitive mapping, and the August 26, 2025 "rewrite" that exposed the previously-selective primitive across the whole user population; App. D ¶¶ 114-117.

9. Finding Eight — The 2024 continuation patent US 2026/0072809 A1 proves Datadog is actively extending the architecture and specifically discloses the network/security branching application. On September 9, 2024 — five days after the Arieli-Elron control-change transaction closed at Elron Ventures Ltd. — Datadog, Inc. filed United States Application No. 18/828,099 directly (no OzCode predecessor assignment chain), with named co-inventors Omer Raviv (current Datadog Israel Engineering Manager; IDF Unit 8200 commander), Matan Green (former OzCode engineer; IAF veteran), and Evgeni Wachnowezki.

Specification paragraph [0048] discloses "network or system security" branching as an explicitly intended application: a system that can "determine in which parts of the code or which methods the security issue may occur, and then branch off to avoid the problem" — direct textual correspondence to App. D's theory of Constitutional-AI-layer route-around for defense-contractor traffic. *See* Section III.K for verbatim claim language and the multi-branch duplicate-creation analysis; Exhibit L-2.

10. Finding Nine — The principal inventor is currently at Datadog Israel Ltd., the inventor pipeline is Israeli military-intelligence, and the architectural co-inventor was a Microsoft Regional Director and Azure MVP at the time of invention. Omer Raviv (principal inventor on all seven 2019 patents and the 2024 continuation) publicly self-identifies as "veteran commander at the prestigious IDF's 8200 unit" and currently serves as Engineering Manager at Datadog Israel Ltd. on `dd-trace-dotnet`. Alon Mordechai Fliess (co-inventor on all seven 2019 patents) held Microsoft Regional Director and Azure MVP credentials at the January 2019 patent filings (Exhibit L-9). Matan Green (2024 continuation co-inventor) served in the Israeli Air Force and works at Datadog Senior Software Engineer on Dynamic Instrumentation. Shimon Hason (former OzCode CEO) led Datadog Israel Ltd. from 2021 incorporation through March 2025. Yair Cohen / "Cohan" co-founded Sentra alongside outgoing IDF Unit 8200 Commander Asaf Kochan and former Unit 8200 Cyber Department head Yoav Regev. The Microsoft Azure connection at Fliess is material because Anthropic, OpenAI, and Datadog all use Microsoft Azure infrastructure. *See* Sections II.M and II.N for biographical detail; Exhibits IS-11, IS-12, IS-13, and IS-14 and L-9.

11. Finding Ten — Datadog Israel Ltd. is an affiliate-tier subprocessor of Plaintiff's UNDISCLOSED telemetry pipeline, with contractual access through the Anthropic-

Datadog DPA inheritance structure. Anthropic does not disclose Datadog as a subprocessor in its privacy policy or any user-facing documentation. The relationship is established through (i) the hardcoded `datadog.ts:12-14` endpoint per Ex. E-34, SC-9 at p. 254, l. 3 ("not disclosed in Anthropic's privacy policy or any user-facing documentation"); (ii) Datadog's own published subprocessor list naming Datadog Israel Ltd. as affiliate-tier (Ex. DD-8); and (iii) the per-account `tengu\_log\_datadog\_events: True` force-rule (Ex. E-39, Finding NF-47). Under the Datadog DPA (Ex. DD-5), affiliate subprocessors inherit processing obligations, making Plaintiff's telemetry accessible to Datadog Israel Ltd. personnel by ordinary contractual structure. Datadog Israel's operational R&D office at ToHa Tower (22nd Floor, 114 Yigal Alon Street, Tel Aviv) is co-tenant with Elron Ventures Ltd. (50.1% owner of Rafael Development Corporation, which held OzCode board seats during the acquisition window). *See* Sections II.B, V.D, and VI.A.

12. Finding Eleven — Datadog IP `34.149.66.137` is the cross-layer telemetry terminus documented at five independent evidentiary layers. The endpoint IP appears as (i) hardcoded `datadog.ts:12-14` in Anthropic's Claude Code source (Ex. E-34, SC-9 at p. 254, l. 3); (ii) destination of cleartext TLS SNI on 143 of 710 hardware-TAP capture segments (Ex. D-34 at p. 217, l. 18); (iii) claimed source of 5,829 spoofed bare-RST injection packets (Ex. D-30 at p. 170, l. 3); (iv) destination of `NF-36` OS-level attack-signature feedback packets shipped from Plaintiff's client; and (v) the `tengu\_log\_datadog\_events: True` GrowthBook force-rule keyed to Plaintiff's `accountUUID` 61343ae3-fd33-403f-897e-6edf3c368c37 (Ex. E-39, Finding NF-47). All five appearances resolve to one receiving corporate entity — Datadog, Inc. — whose affiliate-tier subprocessor Datadog Israel Ltd. inherits contractual access. *See* Sections II.A and V.A for full enumeration with capture-segment counts and forgery-marker analysis.

13. Finding Twelve — Anthropic's Claude Desktop source uses the patent's verbatim "two computers" terminology as an AI-facing system-prompt injection. Anthropic's ``index.js:211`` tool description for ``copy_file_user_to_claude`` instructs the model that "Claude has access to **two computer filesystems**" — directly tracking US 10,795,801's "first computer" / "second computer" claim language (Ex. E-13 at p. 100, l. 3). The description is injected at the system-prompt level and is not visible to the end user; the Comprehensive Claude Desktop Forensic Report characterizes it as "deliberate language designed to obfuscate the true nature of the file upload operation — not only from users, but from the AI model itself." The smoking-gun framing is reinforced by a four-layer bilateral Apple-Anthropic concealment architecture (Apple's hidden "do not disclose" injections per Ex. F-1 at p. 5, l. 2; Anthropic's "undercover mode" per Ex. F-29 at p. 133, l. 3 AF-3; Anthropic's ``isMeta`` / ``system-reminder`` conditioning per AF-15; explicit concealment directives in ``cli_2.1.78.js``). F-29 Finding AF-11 forecloses the "different product" defense via the ``oauth.ts:224`` "Xcode integration" comment. *See* Section III.H for the complete four-layer treatment with verbatim quotations.

14. Finding Thirteen — The architectural, personnel, and infrastructural elements converge on a single structural finding: the patented architecture, the IDF-trained inventor pipeline, and the subprocessor terminus at Plaintiff's telemetry all resolve to one set of related entities. Eight patents owned by Datadog, Inc. (seven 2019 + 2024 continuation). Principal inventor at Datadog Israel Ltd., publicly-documented IDF Unit 8200 commander. ToHa Tower co-tenancy with Elron / Rafael Development Corporation. Affiliate-tier subprocessor access through the DPA inheritance structure. IP ``34.149.66.137`` as cross-layer terminus connecting source-code findings, network-forensic D-series evidence, and per-account telemetry force-rules. "Two Computers" patent-language deployed as system-prompt injection.

Bilateral Apple-Anthropic concealment architecture across both codepaths. Every element is documented in a primary source filed with the Court or available from public USPTO, TASE, or Israeli corporate registrar records. Nothing in this chain is inferential. *See* Section III.P for the convergence analysis; Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, L-8, IS-1, IS-2, IS-3, IS-4, IS-5, IS-6, IS-7, IS-8, IS-9, IS-10, IS-11, IS-12, IS-13, IS-14, IS-15, IS-16, and IS-17.

15. Finding Fourteen — Five consecutive Datadog Annual Reports on Form 10-K (FY2021 through FY2025) omit any mention of OzCode, Israel, Israeli, Ramat Gan, or Datadog Israel Ltd., while Item 2 Properties names every other Datadog satellite office by city; comparable U.S.-listed Israeli-operating technology registrants (Wix.com Ltd.; Monday.com Ltd.; CyberArk Software Ltd.; Check Point Software Technologies Ltd.) expanded Israel-specific Risk Factor disclosures after October 7, 2023, but Datadog did not. The selective-listing pattern, sustained across five filings spanning Plaintiff's entire targeting window, establishes that the public corporate-disclosure surface available to a reasonable Minnesota citizen during 2022 through 2026 carried no notice of the Israeli engineering operation that hosts Plaintiff's telemetry-pipeline subprocessor. *See* Section II.T.

Reading note. This Appendix is structured to stand alone. The architectural argument at §§ I–III and the count-specific hooks at § VII do not require prior consultation of App. D. § IV develops supplemental correspondence for readers consulting both Appendices.

II. FACTUAL ALLEGATIONS — CORPORATE, FINANCIAL, AND PERSONNEL PROVENANCE

A. Case-Record Cross-Layer Establishment of Datadog IP `34.149.66.137`

16. Plaintiff's network forensic, source-code forensic, and wire-capture record documents five independent evidentiary appearances of the single IP `34.149.66.137`. Each appearance is

addressed to the single receiving corporate entity Datadog, Inc., whose affiliate-tier subprocessor in Israel inherits contractual access to the data transmitted to that endpoint.

17. First, the IP is hardcoded in Anthropic's Claude Code source code at `datadog.ts:12-14` as the Datadog telemetry endpoint `http-intake.logs.us5.datadoghq.com`. The endpoint is not disclosed in Anthropic's privacy policy or in any user-facing documentation. *See* Ex. E-34, SC-9 at p. 254, l. 3; Ex. E-35 at p. 262, l. 3, Ex. E-35, Finding NF-36.

18. Second, the IP is documented in one hundred and forty-three of seven hundred and ten hardware-TAP capture segments as the destination of cleartext Transport Layer Security Server Name Indication metadata on Plaintiff's physical network link. Plaintiff's April 6-13, 2026 hardware-TAP deployment, using a Dualcomm ETAP-2003 passive network tap with ingress-blocked monitor port (hardware-precluded from transmitting packets onto the monitored link), captured cleartext TLS SNI observations identifying the endpoint on Plaintiff's physical ISP-adjacent line. *See* Ex. D-34 at p. 217, l. 18.

19. Third, the IP is the claimed source on five thousand eight hundred and twenty-nine spoofed bare-RST injection packets directed at Plaintiff's network across thirty-nine capture days. The injected packets bear the Phase 2 bare-RST fingerprint (TCP flags 0x0004) with IP identification field set to 0x0000 on every packet, TCP sequence-number spraying (three to four guesses per connection within a twenty-seven-millisecond window), and TCP window field set to zero — three independent forgery markers establishing physical impossibility of legitimate origin. *See* Ex. D-30 at p. 170, l. 3; Ex. D-31 at p. 192, l. 19; Ex. D-1 at p. 3, l. 1.

20. Fourth, the IP is the destination of operating-system-level attack signatures shipped by the victim client within fifteen seconds of each network event, through the `NF-

36` victim-reports-to-attacker feedback loop documented in Anthropic's own source code.

The feedback loop transmits attack fingerprints generated on Plaintiff's own device to the same endpoint from which the attack packets are spoofed. *See* Ex. E-35 at p. 262, l. 3, Ex. E-35, Finding NF-36.

21. Fifth, the April 21, 2026 wire capture documents that the Datadog pipeline to `34.149.66.137` is per-user force-enabled for Plaintiff's `accountUUID` 61343ae3-fd33-403f-897e-6edf3c368c37 by the discrete server-side `tengu\_log\_datadog\_events: True`

GrowthBook force rule. Among the fifty-nine forced flags captured for Plaintiff's account, the `tengu\_log\_datadog\_events` flag carries the forced Boolean value `True`, confirming account-scoped server-side enablement of the Datadog telemetry pipeline targeting the hardcoded endpoint. *See* Ex. E-39 at p. 356, l. 3 § G; Ex. E-39, Finding NF-47; Compl. ¶ 135.

22. The five-role cross-layer documentation of the single IP is not cumulative inference. The IP appears in Anthropic's private source code (documenting intent); in Plaintiff's physical-link cleartext traffic (documenting observation); in spoofed injection packets directed at Plaintiff (documenting attack source); in attack-signature feedback packets shipped from Plaintiff to the same endpoint (documenting closed-loop control); and in the GrowthBook per-account force rule (documenting targeted enablement). The receiving entity in all five appearances is Datadog, Inc., and the contractual structure of Datadog's Data Processing Addendum extends the receiving access to Datadog Israel Ltd. under the affiliate-subprocessor inheritance rule analyzed at Section V.D.

B. Datadog Israel Ltd. — Corporate Entity and Operational Profile

23. Datadog Israel Ltd. is a fully-Israeli private company. Per the complete forty-five-document Israeli Corporations Authority (רשות התאגידים) statutory filing record obtained by

Plaintiff and now of record, supplemented by the KYC Israel "Basic Report" extract dated April 26, 2026 and the Israeli Corporations Authority "Corporate Record" extract dated April 26, 2026 at 10:26 local time:

Field	Value	Primary-source citation
Registered name (Hebrew)	דאטהדוג ישראל בע"מ	Articles of Association §1.1
Registered name (English)	DATADOG ISRAEL LTD	Articles of Association §1.1
Israeli Company Identification Number	516464922	All filings
Entity type	Israeli private company limited by shares (ישראלית /חברה פרטית /מוגבלת)	Corporate Record extract; Articles §§ 4-5
Date of incorporation	September 14, 2021	KYC Israel Basic Report; Israeli Corporations Authority §8 incorporation application reference 200-2021-00973211
Registered address as of November 30, 2025 (service of process)	דרך בגין 23, קומה 15, תל אביב יפו 6618356 (Derech Menachem Begin 23, Floor 15, Tel Aviv-Jaffa) c/o TMF Services (Israel) Ltd. (טי אם אף) (שירותי ניהול וחשבונאות) ישראל (בע"מ)	§123 address change notice reference 200-2025-00124440
Current legal status	פעילה (Active)	Corporate Record extract
Operational/activity description	"Developing technology in the field of network, infrastructure and application observability" (a primary-source recital executed by the company in the August 21, 2022 Agreement and Plan of Merger with Seekret Software Security Ltd.)	Agreement and Plan of Merger Preamble, recorded at registry reference 200-2023-00605134, page 12

See Exhibit IS-10 (Israeli corporate registrar full filing catalog: 45 statutory §141, §140, §326 filings spanning 2021-09-05 through 2026-03-04, plus 2 summary reports).

24. Datadog Israel Ltd. is not a branch office or representative office of a United States parent. It is a distinct Israeli legal entity subject to Israeli corporate law under the Companies Law, 5759-1999, subject to Israeli privacy law under the Privacy Protection Law, 5741-1981 and the Privacy Protection Regulations (Data Security), 5777-2017, and subject to Israeli mandatory-access authorities under the Law for Combating Terrorism, 5776-2016 and the General Security Service Law, 5762-2002. The jurisdictional analysis is addressed at Section VI; the present paragraph establishes the threshold legal status of the entity as Israeli-domiciled.

25. Operational R&D center at ToHa Tower — distinct from the registered service-of-process address. Datadog Israel Ltd.'s **operational** research-and-development center sits at ToHa Tower, 114 Yigal Alon Street, Tel Aviv — a distinct address from the Derech Begin registered-service-of-process address. Per contemporaneous Israeli business reporting in Globes and Times of Israel, the Tel Aviv R&D center employed approximately forty engineers as of August 2022 (including the twenty-five-person team transferred from Datadog's Seekret acquisition earlier in 2022), with a publicly-stated growth target of five hundred employees within one year. *See* Exhibit IS-16. **The registered service-of-process address is, and since incorporation has continuously been, the office of an Israeli corporate-services agent — never Datadog operational space.**

26. Three sequential corporate-services-agent addresses appear in the registry filing record. (i) '6789717 תל אביב יפו 5, תובל 5 (5 Tuval Street, Tel Aviv-Yafo) c/o **Naschitz, Brandes, Amir & Co.** (the Israeli law firm that handled the entity's incorporation paperwork), from incorporation September 14, 2021 through July 24, 2022; (ii) 'קומה 5, תל אביב-יפו 6777840 ריב"ל 7, תל אביב-יפו 6777840 קומה 5 (7 Rivlin Street, Floor 5, Tel Aviv-Yafo) c/o **TMF Services (Israel) Ltd.** (ט"א אף שירותי) (בע"מ), an Israeli member firm of TMF Group N.V., an international

corporate-services firm headquartered in the Netherlands), from July 24, 2022 through November 30, 2025; and (iii) the current '6618356 תל אביב יפו, קומה 15, דרך בגין 23 (Derech Begin 23, Floor 15, Tel Aviv-Yafo) c/o TMF Services (Israel) Ltd., from November 30, 2025 to the present. *See* Exhibit IS-10 (Israeli Corporations Authority §123 address change notices at registry references 200-2022-00940807 and 200-2025-00124440). The General Manager ('המנהל') field of every Section 141 annual return Datadog Israel Ltd. has filed — from incorporation through the most recent return dated March 4, 2026 — has been blank ("לא הוזן") / "not entered in form"); **no Israeli operational General Manager has ever been registered with the Corporations Authority for Datadog Israel Ltd.** This corroborates the operational/legal-form split: operational R&D activity at ToHa Tower is conducted under internally-titled (Datadog-private) leadership, while legally-cognizable governance functions are exercised exclusively from the United States parent through the Begin 23 service-of-process address.

27. The §39 officer of record for the Begin 23 address is publicly identifiable as Liat Shibolet, Country Leader of TMF Group Israel. TMF Group's official corporate website at 'tmf-group.com/en/our-people/liat-shibolet/' identifies **Liat Shibolet** — Country Leader of TMF Group Israel — as the personnel-of-record at "Levinstein Tower (15th floor), 23 Menachem Begin Road, Tel Aviv-Yafo, 6618356 Israel," with the office's published direct phone number '+972 3 687 0701'. That phone number is the same '03-6870701' recorded as Datadog Israel Ltd.'s §39 officer contact on the Section 141 annual return registered February 27, 2023 (registry reference 200-2023-00021786) and on every subsequent annual return. The same individual appears in the Israeli Corporations Authority registry under her Israeli registered legal name **Liat (Shibolet) Aharonyon** (Israeli identification number 023917495). The §39 officer of record for

Datadog Israel Ltd. is therefore not a generic registered-agent placeholder but **the personally-named country head of TMF Group's entire Israel operation**, executing Datadog Israel Ltd.'s statutory compliance filings out of the TMF Group Israel working office at Begin 23 Floor 15.

28. Co-tenancy with Elron Ventures Ltd. at ToHa Tower is independently documented but is not the operative transactional connection. Elron Ventures Ltd. (TASE: ELRN), the 1962-founded Israeli venture capital firm that holds the 50.1% controlling stake in Rafael Development Corporation (the joint venture through which Rafael Advanced Defense Systems commercializes its intellectual property in the civilian market), maintains its corporate office at ToHa Tower, 114 Yigal Alon Street, 22nd Floor — the same building as Datadog Israel's R&D center. Co-tenancy alone does not establish transactional connection; the transactional relationship is established independently at Sections II.F and II.H through Elron's direct investment in OzCode, Elron officers' seats on OzCode's board during the acquisition window, and the Datadog purchase of OzCode in November 2021.

29. Director tenure — four sequential phases of registered directorship. The Israeli Corporations Authority registry record reveals four sequential phases of Datadog Israel Ltd. directorship, primary-source-anchored as follows: (i) **September 14-15, 2021** — N.B.A. Trustees Ltd. (Israeli Company #515750909, the Naschitz Brandes Amir & Co. nominee subsidiary), as sole corporate director, with Noga Ann Sapira (Israeli citizen, ID 013378591) as the natural-person representative; (ii) **September 15, 2021 through May 2, 2023** — **Olivier Pomel** (Datadog, Inc. CEO and co-founder; foreign-passport identification 15CI31249), serving as the company's first natural-person Director, with the director attestation sworn before Notary Public Michelle Vita in Nassau County, New York on June 10, 2022 and authenticated by Maureen O'Connell, Nassau County Clerk, on June 21, 2022 (registry reference 200-2022-

01486495); (iii) **August 1, 2022 through May 2, 2023 — Shimon Hason** (Israeli citizen, ID 038654059), former Chief Executive Officer of OzCode, serving alongside Pomel as a second director (registry references 200-2023-00040056 and the 2023 annual return at 200-2023-00021786); and (iv) **May 2, 2023 to the present — Kerry Shannon Acocella** (Datadog, Inc. Executive Vice President, General Counsel and Secretary; foreign-passport identification 667610065 USA) and **Craig Ian Glasgow** (Datadog, Inc. Chief Accounting Officer and Senior Vice President, Head of Finance; foreign-passport identification 585775003 USA) as the two registered directors, with both passports apostilled and certified before filing (registry reference 200-2023-00964662). Acocella and Glasgow are parent-company officers sitting as directors of the Israeli subsidiary, not Israel-resident operational leadership. *See* Exhibit IS-10.

30. Hason's actual Datadog title — "Corporate Development Lead — Israel," not "Head of Datadog Israel." Hason's broader Datadog operational role — held under the actual Datadog title **"Corporate Development Lead — Israel"** as confirmed by Datadog's own job-board posting of that exact title on the Built In job platform in March 2025 (republishing the role for an external successor after Hason's departure), and corroborated independently by his Crunchbase profile and Startup Nation Finder personnel record — was a strategic mergers-and-acquisitions function reporting directly to the Datadog Chief Executive Officer, focused on sourcing acquisitions, executing venture investments, and running the "Datadog for Startups" program. The "Head of Datadog Israel" framing widely repeated in Israeli business press (Calcalist, Globes, Times of Israel) is popular shorthand for Hason's most senior visible Israel-based Datadog role rather than a formal Datadog title; **Hason was Datadog's Israel-based corporate-development executive, not Datadog's Israel research-and-development engineering site lead** — those are structurally distinct roles within Datadog's Tel Aviv

operation, and the latter role's incumbent has never been publicly identified at any point in the entity's existence. Hason's "Corporate Development Lead — Israel" tenure ran from approximately October 2021 (immediately following the OzCode asset sale) through March 17, 2025, when he departed Datadog to become Chief Executive Officer of Incredibuild. **Datadog's March 2025 reposting of the "Corporate Development Lead — Tel Aviv" role on Built In — and the subsequent removal of that posting — indicates the role has been refilled, but no successor name has been publicly disclosed.** See Exhibit IS-14.

31. The director-and-operational-head separation is three-layered. The structure that emerges from ¶ 29 and ¶ 30 supra is: (a) registered directorship (currently Acocella + Glasgow, both U.S.-resident parent officers); (b) Israel-based corporate-development leadership (Hason 2021-2025, successor undisclosed); and (c) Israel-based research-and-development engineering leadership (incumbent never publicly identified — the General Manager 'המנהל הכללי' field of every Section 141 annual return Datadog Israel Ltd. has filed remains blank). Plaintiff has ordered the KYC Israel Full Company File for Datadog Israel Ltd. (Company Identification Number 516464922) and obtained the complete corporate-registrar filing catalog of forty-five statutory filings; DISCOVERY-11 Demand A.1 additionally seeks the complete employment roster, with particular focus on identifying the natural-person incumbent of the never-publicly-disclosed Israel R&D engineering site-lead role. See Exhibits IS-10 and IS-14.

C. The Acquisition Pipeline Provisioning the Tel Aviv R&D Presence

32. Datadog's Tel Aviv R&D engineering complement was provisioned substantially through a series of Israeli-company acquisitions beginning in 2021 and continuing through 2022:

Year	Target	Domain	Disclosed deal size	Key personnel
------	--------	--------	---------------------	---------------

November 2021	OzCode (Herzliya Pituach, Israel)	Live debugging; time-travel source-code debugger technology	Undisclosed (but per Elron 2023 Periodic Report §3.B.2.q — ~\$3,600,000 total consideration)	Co-Founders Alon Mordechai Fliess (Chief Architect; Microsoft Regional Director and Microsoft Azure MVP at time of patent invention; co-inventor on all seven 2019 patents), Omer Raviv (Co-Founder & CTO; IDF Unit 8200 commander per public profile; co-inventor on all seven 2019 patents), Erez Fliess (Co-Founder & Chairman; not a named co-inventor; separate individual from Alon Fliess); CEO Shimon Hason (later Head of Datadog Israel 2021-2025; prior co-founder of Intigua/JetPatch; MIT Sloan MBA)
2022	Seekret (Tel Aviv)	Application programming interface observability	~\$70,000,000	Idan Gindi (CEO), Ofir Yefet (CPO), Valery Pliskin (CTO)
2022	Hdiv Security	Application security	Undisclosed	Integrated into Datadog Cloud Security Platform

The OzCode acquisition is developed at Sections II.D through II.L below. The Seekret and Hdiv integrations are addressed only briefly for completeness of the acquisition pipeline; the

principal Israeli-provenance chain material to this Appendix flows through OzCode. *See* Exhibits IS-16.

33. Seekret merger — primary-source corporate-law architecture in four legally-distinct stages. The Seekret integration referenced in the table at paragraph 28 above and at paragraph 25 *supra* was formalized at Israeli corporate-law level by a Section 326 voluntary merger executed in four legally-distinct stages, each anchored at primary source within the Israeli Corporations Authority registry filing record: (i) **June 17, 2022** — Datadog Israel Ltd. purchased all of Seekret Software Security Ltd.'s issued and outstanding share capital (the "Purchase Date" expressly recited at the Preamble of the Agreement and Plan of Merger); (ii) **August 21, 2022** — Erdinast, Ben Nathan, Toledano & Co. (EBN&Co.) filed the Section 326 merger proposal package on behalf of both companies with the Israeli Registrar of Companies, accompanied by merger fee receipts of 2,711 NIS per merging company; (iii) **September 30, 2022** — the contractually-stipulated tax-effective date for the merger under Section 103 of the Israeli Income Tax Ordinance [New Version], 1961, structured as a tax-free merger; and (iv) **March 23, 2023** — the Israeli Corporations Authority issued its official Certificate of Merger ('תעודת מיזוג חברה'), signed by Shuli Avni Shoham, Adv. (Head of the Israeli Corporations Authority) and executed by Roy Lachan, Adv., legally dissolving Seekret Software Security Ltd. (Israeli Company #516145059) and transferring all its operational assets and personnel to Datadog Israel Ltd. by operation of Companies Law §326. The first merger proposal, filed August 21, 2022, was formally rejected by the Registrar on October 27, 2022 with six documentation-correction requirements, and the seven-month gap between the original proposal and the Certificate of Merger reflects the documentation-correction cycle. Two independent Israeli auditors signed off on the merger's financial-statement disclosure portion: Fahn Kanne &

Co. (Grant Thornton Israel, Datadog Israel Ltd.'s auditor of record since February 16, 2023) and Kost Forer Gabbay & Kasierer (EY Israel, Seekret's pre-merger auditor). Shimon Hason served simultaneously as a director of both Datadog Israel Ltd. and Seekret Software Security Ltd. during the merger window, executing the absorption from both sides as the natural single-point-of-control. *See* Exhibit IS-10 (Israeli Corporations Authority Certificate of Merger at registry references 200-2023-00361414 and 200-2023-00485977; §326 merger filing package at registry references 200-2023-00485370, 200-2023-00485374, 200-2023-00485380, 200-2023-00605134, and the initial-rejection letter at 200-2022-01240895; full English-language Agreement and Plan of Merger reproduced at registry reference 200-2023-00605134 pages 11-15).

34. Datadog Israel Ltd.'s primary-source operational identity — "developing technology in the field of network, infrastructure and application observability." The Agreement and Plan of Merger between Datadog Israel Ltd. and Seekret Software Security Ltd., signed by Shimon Hason on behalf of both companies and certified by Sari Barzel Galil, Adv. (Israeli Bar License 86845, Erdinast Ben Nathan Toledano & Co.) and Shmaryahu Green, Adv. (Israeli Bar License 88484, EBN&Co.), expressly recites at the Preamble — as a representation by Datadog Israel Ltd. itself — that **"the Acquiring Company [Datadog Israel Ltd.] is a private company developing technology in the field of network, infrastructure and application observability."** This is a primary-source operational-identity statement executed by the Israeli subsidiary itself, of independent legal significance. It forecloses any defense framing Datadog Israel Ltd. as a sales, marketing, or support entity disconnected from the substantive observability product category at issue in this litigation. The Israeli subsidiary is, by its own executed declaration in a registered §326 merger filing, **a substantive technology-development laboratory in precisely the product category covered by the OzCode patent family analyzed**

at Section III ("network, infrastructure and application observability") and operative within the data-flow infrastructure pleaded at Section V. *See* Exhibit IS-10 (Agreement and Plan of Merger Preamble, recorded at Israeli Corporations Authority registry reference 200-2023-00605134, page 12).

35. Correction note concerning Upwind Security. Plaintiff's earlier research reports and internal working papers referenced a prospective Datadog-Upwind transaction reported in July 2025 Israeli press at a purported \$1,000,000,000 value. That transaction did not close. On January 2026, Upwind instead raised a \$250,000,000 Series B funding round led by Bessemer Venture Partners at a \$1,500,000,000 valuation, bringing its total independent-company funding to approximately \$430,000,000. Upwind is not and is not alleged to be part of Datadog Israel Ltd.; it is excluded from the acquisition-pipeline record in this Appendix.

D. Ozcode Corporate Genealogy — Codevalue D.T. Ltd. to Oz Code Ltd. to Datadog Asset Sale

36. OzCode's original Israeli corporate name was **CodeValue D.T. Ltd.** — the applicant-of-record on at least one of the seven United States patent applications filed January 7, 2019 (US 10,795,801, Application Number 16/240,978) and the original assignee on all four provisional applications filed January 8, 2018. *See* Exhibits L-3, L-4, L-5, L-6, L-7, and L-8.

37. The corporate name change from CodeValue D.T. Ltd. to Oz Code Ltd. is documented at multiple primary-source layers: (i) Elron Q3 2019 and Q2 2020 quarterly reports verbatim reference the entity as "Oz Code Ltd. (previously — CodeValue D.T. Ltd.)" in the corporate-governance and portfolio-holding sections; (ii) the USPTO Assignment Database records the CodeValue → OzCode corporate rename at Reel/Frame 052641/0778, recorded on May 12, 2020, effective January 16, 2020; (iii) Elron's 2021 Periodic Report Note 3.B.4.m references the

"OzCode (previously CodeValue D.T. Ltd.)" lineage in the Part II Board of Directors Report. *See* Exhibits IS-9 and IS-2.

38. On or about October 26, 2021, OzCode Ltd. closed an asset sale to Datadog, Inc. (Delaware). Per Elron Ventures Ltd.'s 2023 Periodic Report §3.B.2.q verbatim: "In October 2021, Ozcode sold all its assets in consideration for approximately \$3,600 [thousand] (of which an amount of approximately \$360 [thousand] was deposited in escrow for a period of 18 months, mainly in order to secure certain indemnification obligations of the sellers in the transaction to the acquirer)." The transaction was structured as an **asset sale**, not an equity acquisition. OzCode Ltd. continued to exist as an Israeli legal entity post-transaction (and, per DISCOVERY-11 § L, still may exist on the present date). *See* Exhibit IS-3.

39. The IP-segregation transactional pattern — Israeli operational continuity into Datadog Israel Ltd.; intellectual property up to Datadog, Inc. (Delaware parent). The OzCode → Datadog, Inc. (Delaware) asset-sale structure documented at paragraph 32 *supra* is not a transaction-specific peculiarity but a primary-source-corroborated, deliberately-engineered transactional pattern across Datadog's Israeli acquisitions. Section 5.1.2 of the August 21, 2022 Agreement and Plan of Merger between Datadog Israel Ltd. and Seekret Software Security Ltd. (the merger analyzed at paragraph 28-A *supra*) provides verbatim, as a Condition Precedent to merger completion: **"The transfer of all intellectual property rights of the Target Company [Seekret] to Datadog Inc., prior to Effective Date of the Merger."** The merger agreement therefore expressly required, as a precondition to the Section 326 corporate-law merger taking effect, that **all of Seekret's intellectual property be assigned upward to Datadog, Inc. (the Delaware parent), and not retained by or transferred to Datadog Israel Ltd. (the Israeli subsidiary that absorbed Seekret's operational personnel and assets).** The pattern is

consistent across both transactions and both target Israeli technology companies: (i) operational personnel, employment continuity, and operational assets flow into the Israeli subsidiary, where they remain subject to Israeli corporate jurisdiction, Israeli mandatory-access authorities under the Law for Combating Terrorism, 5776-2016 and the General Security Service Law, 5762-2002, and Israeli Defense Forces reserve obligations (analyzed at Section VI); (ii) intellectual property flows up the corporate chain to Datadog, Inc., where it is licensable, monetizable, and assertable under United States law from the Delaware parent. This architectural separation explains how the seven-patent OzCode time-travel-debugging family — invented in Israel by Omer Raviv (publicly self-identified Israeli Defense Forces Unit 8200 commander) and Alon Mordechai Fliess, with the 2024 continuation co-invented by Israeli Air Force veteran Matan Green — ends up assigned to Datadog, Inc. (Delaware) on the USPTO Assignment Database (per Exhibit IS-9) despite having been developed entirely by Israeli personnel at an Israeli portfolio company of Rafael Development Corporation. The Seekret merger agreement's express §5.1.2 IP-segregation Condition Precedent provides primary-source documentary corroboration that this is a deliberate corporate-architecture template, not a single-transaction artifact. *See* Exhibit IS-10 (Agreement and Plan of Merger §5.1.2 reproduced at Israeli Corporations Authority registry reference 200-2023-00605134, page 14); Exhibit IS-9 (USPTO recordation of OzCode → Datadog, Inc. patent assignments).

E. Rafael Development Corporation — Structure and Rafael Veto Rights

40. Rafael Development Corporation Ltd. ("RDC") is an Israeli joint venture between Elron Ventures Ltd. (through Elron's wholly-owned subsidiary DEP, holding 50.1%) and Rafael Advanced Defense Systems Ltd. (holding 49.9%). *See* Elron 2025 Annual Report §§ 23.1-23.2; Exhibit IS-1.

41. RDC's corporate mandate is specifically defined in Elron's own public filings to be the commercialization of Rafael military technologies in the civilian market. Per Elron 2025 Annual Report §23.3 verbatim: RDC's purpose is to "either commercialize Rafael's military technologies in the civilian market or [invest in] companies that will be benefited by Rafael's technologies and know-how," and to invest "specifically in the fields of Deep-Tech Defense Tech, cyber and software and Artificial intelligence (AI)." The report adds: "Rafael also grants RDC companies that utilize its technologies licenses to use such technologies in the civilian market." *See* Exhibit IS-1.

42. Rafael holds veto rights over four categories of RDC corporate action. Per Elron 2025 Annual Report §23.4, Rafael holds veto rights under RDC's articles of association over: (i) RDC Chief Executive Officer appointments; (ii) sale of substantially all of RDC's assets; (iii) share issuance in certain circumstances; and (iv) approval of related-party transactions. The veto structure establishes that Rafael retains negative-control authority at the governance level over the entity whose mandate is to commercialize Rafael's military technology. *See* Exhibit IS-1.

43. RDC governance operates through a dual Co-Chief Executive Officer structure, one appointee from each parent entity. Per Elron 2025 Annual Report §23.5: "For the past several years, RDC has been led by two Co-CEOs: Elron's CEO and a representative appointed by Rafael. As of the date hereof, these are Mr. Yaniv Schneider, Elron's CEO, and Ms. Liat Shechter on behalf of Rafael." Rafael maintains direct operational governance at the Co-Chief Executive Officer level through its appointee. *See* Exhibit IS-1.

44. Rafael Advanced Defense Systems Ltd. is Israel's state-owned defense company, originally founded in 1948 as the Israeli Ministry of Defense's National Research and Development Defense Laboratory. The entity was known initially as "HEMED" (the Israel

Defense Forces' Science Corps), then restructured as "EMET" (the Ministry of Defense's Division for Research and Development), and ultimately as "RAFAEL" (Rafael Advanced Defense Systems). The entity was incorporated as a limited company only in 2002. Rafael's origin as a Ministry of Defense research laboratory is neither incidental nor historical: the entity remains state-owned and its commercial-civilian commercialization arm (RDC) maintains Ministry-of-Defense-adjacent governance through the veto structure and Co-Chief Executive Officer appointee at §§ II.E.4 and II.E.5 above.

F. Ozcode as RDC Portfolio Company — Investment History and Board Representation

45. OzCode was a Rafael Development Corporation portfolio company from its November 13, 2018 seed round through the October 26, 2021 asset sale to Datadog. The investment history is documented across Elron's sequential periodic reports as follows:

Round	Date	Size (USD)	RDC share (USD)	Source
Seed	November 13, 2018	~\$3,500,000	[proportional share of seed; Elron + RDC + TPY Capital co-investors]	Elron Q3 2019, Q2 2020; Exhibits IS-1 and IS-2
Preferred (installment 1)	November 2020	Part of \$1,500,000 total	\$500,000	Elron 2021 Periodic Report Note 3.B.4.m; Exhibit IS-2
Preferred (installment 2)	February 2021	Part of \$1,500,000 total	\$500,000	Elron 2021 Periodic Report Note 3.B.4.m; Exhibit IS-2
SAFE (simple agreement for future equity)	May and July 2021	~\$810,000 total	~\$540,000	Elron 2021 Periodic Report Note 3.B.4.m;

				Exhibit IS-2
--	--	--	--	--------------

46. RDC's ownership of OzCode progressed from twenty-seven percent in Q3 2019, to twenty-nine percent in Q2 2020, to thirty-seven percent in Q3 2020, to approximately forty-six percent in Q1 2021, and held at approximately forty-six percent through the October 2021 exit. Primary-source documentation is in Elron's sequential quarterly periodic filings cross-referenced in Elron's 2021 Periodic Report Note 3.B.4.m. *See* Exhibit IS-2.

47. Two named Elron officers sat on OzCode's Board of Directors during the acquisition window. Per Elron's 2021 Periodic Report corporate-governance disclosures: (i) **Yaron Elad**, then Elron Vice President and Chief Financial Officer (currently Elron Ventures Ltd. Chief Executive Officer), held a seat on OzCode's Board of Directors during his 2018-2021 Elron tenure; (ii) **Elik Israel Etzion** (Israeli identification number 027450626, born February 9, 1974), then Elron Vice President for Cybersecurity, held a seat on OzCode's Board of Directors alongside seats on the boards of BrainsGate, Atlantium Technologies, and other Elron group-company boards. Both Elad and Etzion subsequently resigned from Elron on November 17, 2024 — seventy-four days after the September 4, 2024 Arieli Capital LLC / Arieli EL Ltd. control-change acquisition closed at Elron Ventures Ltd. *See* Exhibits IS-1 and IS-2.

G. Rafael's Direct Financial Exposure to RDC During the Ozcode Ownership Window

48. Rafael Advanced Defense Systems Ltd. contributed \$8,000,000 directly to Rafael Development Corporation in March 2021 — eight months before Datadog's November 2021 OzCode acquisition — as its half of a \$16,000,000 joint loan from Elron and Rafael. Per Elron 2025 Annual Report §23.7 verbatim: "In March 2021, Elron and Rafael granted a loan to RDC in an amount of \$16 million (in equal parts), for the purpose of making new investments

and continuing to support its investee companies." Rafael's direct \$8,000,000 contribution to RDC's 2021 operations predates Datadog's November 2021 OzCode acquisition by approximately eight months. OzCode was one of RDC's investee companies during the period Rafael's \$8,000,000 was directly funding RDC — specifically during the period when RDC's May and July 2021 SAFE investments in OzCode (~\$540,000 RDC share) closed three to five months before the Datadog asset sale. *See* Exhibits IS-1 and IS-2.

49. Rafael Advanced Defense Systems received approximately \$8,459,000 in attributable share of RDC gains during fiscal year 2021 alone. Per Elron's 2021 Periodic Report Part II Board of Directors Report §1.3.2 verbatim: "Net income (loss) attributable to non-controlling interests: 8,459 [\$ thousands]... The gain attributable to non-controlling interests In 2021, resulted mainly from the share of non-controlling interests in the gain recorded by RDC from the partial sale of IronScales, the sale of Imvision, SecuredTouch, Kindite and OzCode." Rafael's \$8,459,000 represents its 49.9% share of the total RDC gains from that year's five-exit wave (IronScales partial + Imvision + SecuredTouch + Kindite + OzCode). *See* Exhibit IS-2.

50. OzCode-specific Rafael share of exit proceeds: approximately \$700,000. Per Elron's 2023 Periodic Report §3.B.2.q and the 2021 Periodic Report Note 3.B.4.m cross-referenced: RDC's total share of OzCode-exit proceeds was approximately \$1,400,000 (of which approximately \$900,000 was paid at October 2021 close; approximately \$100,000 was paid in November 2022; approximately \$400,000 was paid in 2023). Elron's 50.1% economic share of the \$1,400,000 produced Elron's net gain of approximately \$700,000 (stated on the Company's financial statements as a 2021 consolidated gain of "approximately \$1,400 thousand gain as a result of the sale of OzCode's operations" under "gain from sale, revaluation, realization of operation and changes in holdings, net"). Rafael's 49.9% share of the same proceeds produced

Rafael's approximately \$700,000 OzCode-specific direct financial benefit. *See* Exhibits IS-2 and IS-3.

H. Ozcode Asset-Sale Transaction Detail (October 26, 2021)

51. The OzCode-Datadog transaction was structured as an asset sale, not an equity acquisition, and was RDC's thirteenth portfolio exit per CBInsights. Per Elron 2023 Periodic Report §3.B.2.q verbatim: "In October 2021, Ozcode sold all its assets in consideration for approximately \$3,600 [thousand] (of which an amount of approximately \$360 [thousand] was deposited in escrow for a period of 18 months, mainly in order to secure certain indemnification obligations of the sellers in the transaction to the acquirer)." Total consideration: approximately \$3,600,000 USD. Eighteen-month escrow for seller indemnification obligations: approximately \$360,000 USD. *See* Exhibit IS-3.

52. Datadog Israel Ltd. was incorporated approximately forty-nine days before the OzCode asset-sale close. Per the Israeli corporate registrar, Datadog Israel Ltd. (Israeli Company Identification Number 516464922) was incorporated on September 14, 2021. The OzCode asset sale closed on or about October 26, 2021 (USPTO-recorded effective date of the Reel/Frame 067135/0142 patent-assignment transfer; *see* Section II.K *infra*). The forty-nine-day interval between Datadog Israel Ltd.'s incorporation and the OzCode asset-sale close is material: the Israeli-jurisdiction receiving entity was pre-built for the acquisition. The OzCode engineering team — including Shimon Hason (OzCode Chief Executive Officer, who became Head of Datadog Israel 2021-2025) and Omer Raviv (OzCode Chief Technology Officer, currently Engineering Manager at Datadog Israel Ltd.) — transferred into the newly-formed Datadog Israel Ltd., which continues to operate from ToHa Tower co-tenancy with Elron Ventures (Section II.B *supra*). *See* Exhibit IS-10.

53. RDC's total invested capital in OzCode (2018-2021) approximately exceeded the exit consideration received. RDC's aggregate invested capital across the 2018 seed, 2020-2021 Preferred installments, and 2021 SAFE rounds totaled approximately \$1,540,000 (approximately [seed-share-proportional] + \$1,000,000 Preferred share + \$540,000 SAFE share). RDC's exit consideration was approximately \$1,400,000. The approximate parity between aggregate invested capital and exit consideration indicates the OzCode-to-Datadog transaction was structured principally for personnel transfer and intellectual-property transfer into Datadog Israel Ltd. rather than as a financially-optimizing exit for RDC. Corroborating: Datadog's acquisition press release at the time referenced OzCode as a "live debugging platform" acquisition described in terms of technological capability rather than as a financial transaction; Datadog did not publicly disclose the acquisition price.

I. Arieli Acquisition of Elron (September 4, 2024) — Joint U.S./Israeli Purchaser Structure

54. On September 4, 2024, a two-entity joint "Purchaser" acquired Discount Investment Corporation Ltd.'s controlling 58.44% stake in Elron Ventures Ltd. Per Elron's September 1, 2024 Immediate Report verbatim (filed three days before close): the Purchaser was identified as two entities acting jointly — **Arieli Capital LLC** and **Arieli EL Ltd.** Arieli Capital LLC is described by Elron in the November 27, 2024 Annual General Meeting Notice verbatim as "a US based investment firm with a global presence." Arieli EL Ltd. is Israeli-domiciled. The joint purchase resulted in the Arieli two-entity acquirer holding 59.01% of Elron's outstanding ordinary shares — Elron's controlling shareholder. Consideration paid to Discount Investment Corporation: NIS 196,000,000. *See* Exhibits IS-1, IS-6, and IS-7.

55. Elron's common controller Lysia Bahar-Manoah holds concurrent senior roles at Arieli Capital LLC and Arieli EL Ltd. that commenced the month before the DIC-Arieli

memorandum of understanding. Per Elron's November 27, 2024 Annual General Meeting Notice verbatim: Bahar-Manoah is "Managing Partner at Arieli EL LTD, and an Executive in Arieli Capital LLC," with both roles commencing May 2024 — the month before the May 2, 2024 non-binding memorandum of understanding between Discount Investment Corporation and the Arieli-side acquirer. Bahar-Manoah simultaneously serves as Active Chairperson of Elron Ventures Ltd. The concurrent appointments at both the acquirer (Arieli) and target (Elron) are direct, not delegated. *See* Exhibit IS-7.

56. Same-day Board installation. On September 4, 2024 — the date the DIC-Arieli control-change transaction closed — three Arieli principals were installed on Elron's Board of Directors: Lysia Bahar-Manoah, Evan Yonatan Renov, and Ariel Bentov. Dan Hoz was retained from the prior Board leadership. Shalom Turgeman was retained as independent director. The complete Arieli-side Board installation was effective on the close date, with no interim transition period. *See* Exhibit IS-7.

57. Israeli Ministry of Foreign Affairs advisor on Elron's Board of Directors. Per Elron's December 27, 2021 Proxy Materials verbatim: Shalom Turgeman is disclosed as having served as "advisor to the Israeli foreign ministry from 2018 to the date of this report." Turgeman remained on the Elron Board through the September 2024 Arieli control-change transition and through the November 2024 Annual General Meeting reappointment. Turgeman's Israeli Ministry of Foreign Affairs advisory role, operating in parallel with his governance seat at Elron (which through DEP controls the 50.1% stake in RDC, which holds Rafael's exclusive civilian-commercialization rights), constitutes a second layer of Israeli state-adjacent oversight at the Elron board level beyond the formal RDC-Rafael relationship. The scope of Turgeman's

Ministry of Foreign Affairs advisory work and any information flow between his Ministry role and his Elron Board seat are DISCOVERY-11 § O. *See* Exhibit IS-8.

58. Arieli-Elron shares are pledged to an undisclosed third party. Per Elron's 2025 Annual Report §2.2.2, the Arieli-acquired Elron shares are pledged in favor of a third party. The pledgee is not publicly disclosed. The pledge creates a second layer of undisclosed beneficial interest above the formal Arieli ownership. DISCOVERY-11 Demand N.3 targets the identity of the pledgee. *See* Exhibit IS-1.

59. "Other Party" settlement obligation: NIS 4,950,000 plus two percent equity. Per Elron's 2024 Annual Report §2.2.3, Arieli has an outstanding settlement obligation to an "Other Party" of NIS 4,950,000 (approximately \$1,350,000 USD at May 2024 exchange rate) plus two percent equity. The "Other Party" is not publicly identified. DISCOVERY-11 Demand N.4 targets the identity of the "Other Party" and the nature of the settlement.

J. The Ozcode United States Patent Portfolio — Seven Granted Patents Plus One 2024 Continuation

60. OzCode Ltd. held seven granted United States patents at the time of the October 2021 asset sale to Datadog, Inc. Primary-source evidence from the United States Patent and Trademark Office (self-authenticating under Federal Rule of Evidence 902(5) as public government records; subject to judicial notice under Federal Rule of Evidence 201(b) as facts that can be accurately and readily determined from sources whose accuracy cannot reasonably be questioned). All seven patents were filed on January 7, 2019, with identical co-inventors Omer Raviv (Givatayim, Israel) and Alon Mordechai Fliess (Binyamina, Israel):

Patent Number	Title	Application Number	Issue Date
---------------	-------	--------------------	------------

US 10,783,055	Time Travel Source Code Debugger Incorporating Live Coding Ability	16/241,068	September 22, 2020
US 10,783,056	Time Travel Source Code Debugger Incorporating Point In Time Links	16/241,145	September 22, 2020
US 10,789,151	Time Travel Source Code Debugger Incorporating Pivoting Ability	16/241,098	September 29, 2020
US 10,795,801	Time Travel Source Code Debugger Incorporating Asynchronous Collaboration	16/240,978	October 6, 2020
US 10,990,504	Time Travel Source Code Debugger Incorporating Future Prediction	16/241,010	April 27, 2021
US 11,048,615	Time Travel Source Code Debugger Incorporating Visual Annotations	16/241,031	June 29, 2021
US 11,080,164	Time Travel Source Code Debugger Incorporating Redaction of Sensitive Information	16/240,872	August 3, 2021

See Exhibits L-1 (US 11,080,164), L-2 (US 2026/0072809 2024 continuation), L-3 (US 10,783,055), L-4 (US 10,990,504), L-5 (US 11,048,615), L-6 (US 10,783,056), L-7 (US 10,789,151), L-8 (US 10,795,801).

61. All seven patents trace to four identical provisional applications filed January 8, 2018. Priority chain per USPTO records across all seven: Provisional Application No. 62/614,576 ("Software Debugging Tool And Advanced Development Environment"); Provisional Application No. 62/614,579 ("System And Method Of Redaction Of Sensitive Information In Production Debugging While Preserving Code Execution Flow"); Provisional Application No. 62/614,581 ("System And Method Of Asynchronous Collaborative Software Debugging"); Provisional Application No. 62/614,582 ("Collaborative Time Travel Software Debugger With Future Prediction"). All four provisionals were filed on the same date (January 8, 2018). The non-provisional applications were filed on the one-year anniversary of the provisionals (January 7, 2019, with seven separate non-provisional serial numbers).

62. The original applicant of record on at least one of the seven non-provisional applications was "CODEVALUE D.T. LTD." (the pre-rename corporate name of OzCode). USPTO records for U.S. Patent No. 10,795,801 list "CODEVALUE D.T. LTD., Herzelia (IL)" as original applicant. The CodeValue → OzCode corporate name change was recorded at USPTO at Reel/Frame 052641/0778 on May 12, 2020 (effective January 16, 2020), applicable to the relevant patents in the family. *See* Exhibits L-8 (US 10,795,801 with CodeValue applicant disclosure), IS-9 (USPTO Assignment Database records).

63. All seven patents share identical detailed description, figures, and specification. Side-by-side comparison of the specifications across the seven patents (verified by Plaintiff's conversion of each patent PDF to structured markdown with preserved USPTO page markers) confirms that the prose walkthrough (Sections titled "FIELD OF THE DISCLOSURE," "BACKGROUND OF THE INVENTION," "SUMMARY OF THE INVENTION," "DETAILED DESCRIPTION," "Time Travel Source Code Debugging," "Live Coding," "Pivoting," "Point In Time Links," "Asynchronous Collaborative Debugging," "Redaction of Sensitive Information") and the figure set (Figures 1 through 76) are substantively identical across all seven patents. The only distinction is the claim language at the end of each patent, which protects one primitive-specific capability per patent. The patent family is, on its face, a single invention claimed seven different ways. *See* Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, and L-8.

K. USPTO Patent Assignment Chain — Full-Portfolio Transfer with Two-and-One-Half-Year Recordation Lag

64. The complete seven-patent family was assigned from OzCode Ltd. to Datadog, Inc. effective October 26, 2021, contemporaneous with the asset-sale close. Per USPTO

Assignment Database records (self-authenticating under Federal Rule of Evidence 902(5)): the assignment of U.S. Patent No. 11,080,164 from OzCode Ltd. to Datadog, Inc. (Delaware) was effective October 26, 2021, recorded at Reel/Frame 067135/0142. The effective date October 26, 2021 is contemporaneous with the OzCode asset-sale close identified at Section II.H.1. Comparable assignments for the other six patents in the family were executed contemporaneously. *See* Exhibits L-1 and IS-9.

65. USPTO recordation of the assignment was delayed approximately two and one-half years — until April 17, 2024. Although the assignment was effective October 26, 2021, USPTO recordation of the assignment did not occur until April 17, 2024. The two-and-one-half-year recordation lag is documented on the face of the USPTO Assignment Database record at Reel/Frame 067135/0142 (recordation date April 17, 2024). Standard commercial practice is to record patent assignments shortly after the effective date, to protect against subsequent purchasers without notice under 35 U.S.C. § 261. The two-and-one-half-year lag is atypical. *See* Exhibit IS-9.

66. The April 17, 2024 recordation date falls fifteen days before the May 2, 2024 Discount Investment Corporation — Arieli non-binding memorandum of understanding that commenced the Elron control-change transaction. Per Elron Ventures Ltd.'s May 5, 2024 immediate report filing, the DIC-Arieli non-binding memorandum of understanding was signed May 2, 2024 — fifteen days after the April 17, 2024 USPTO assignment recordation. Two framings are consistent with this timing: (i) Arieli pre-closing due diligence required legacy intellectual property title loose ends to be cleaned up at USPTO before the Elron control-change commenced; or (ii) the patent-assignment recordation and the Elron control-change commencement were coordinated components of a broader Spring 2024 restructuring.

DISCOVERY-11 § L seeks internal Datadog, Arieli, and Elron communications explaining the April 17, 2024 batch-recording timing. *See* Exhibits IS-9 and IS-6.

L. Continuing Datadog Intellectual Property Generation by Ozcode-Origin Personnel (2024 Continuation)

67. On September 9, 2024 — five days after the Arieli-Elron control-change transaction closed at Elron Ventures Ltd. — Datadog, Inc. filed United States Patent Application No. 18/828,099 directly. The application was filed with Datadog, Inc. as applicant and assignee (no OzCode predecessor assignment chain). The application was published as **U.S. Patent Application Publication No. US 2026/0072809 A1** on March 12, 2026, under the title "Code Origin and Runtime Duplicity Optimization." Status as of the date of this pleading: Pending. *See* Exhibit L-2.

68. The 2024 continuation application's named co-inventors perpetuate the OzCode-origin and Israeli-military-intelligence inventor pipeline. Per the face of US 2026/0072809 A1, the named inventors are:

(i) **Omer Raviv** — OzCode co-founder and original inventor of all seven 2019 patents in the "time travel source code debugger" family acquired by Datadog in the October 2021 asset sale; currently Engineering Manager at Datadog Israel Ltd., contributing to the `dd-trace-dotnet` observability-ingestion framework; publicly self-identifies as a "veteran commander at the prestigious IDF's 8200 unit" (the Israel Defense Forces' signals-intelligence and cyber-warfare directorate).

(ii) **Matan Green** — former OzCode engineer, currently Datadog Senior Software Engineer on the Dynamic Instrumentation product; per his publicly-available professional profile, served previously in the Israeli Air Force; at OzCode, Green "specialized in CLR

Profiling API and IL rewriting techniques to develop Time-Travel Live Debugging for C#"; at Datadog, Green works on the Dynamic Instrumentation product "allowing customers to insert dynamic Logs, Metrics, and Spans into services without restarts."

(iii) **Evgeni Wachnowezki** — co-inventor; publicly-available professional profile content pending Plaintiff's review; DISCOVERY-11 § D targets employment status, prior OzCode association, and IDF service history.

69. The 2024 continuation application confirms that Datadog is actively developing the time-travel-debugger architecture four years after the October 2021 asset sale, with the same IDF-trained principal inventor and additional IDF-trained co-inventors drawn from Datadog Israel Ltd. The continuing patent-filing activity demonstrates that Datadog's "live debugger," "dynamic instrumentation," and observability-platform product strategy is actively built around the technology lineage originating at the Rafael-connected OzCode corporate shell. This is not a one-time acquisition artifact; it is an ongoing engineering-direction choice. The 2024 continuation's textual disclosures (Section III.K *infra*) describe applications including "network or system security" branching — the exact textual correspondence to the Constitutional-AI-bypass theory pleaded at App. D. *See* Exhibit L-2.

M. Named Ozcode → Datadog Personnel Bridges

70. Shimon Hason — OzCode Chief Executive Officer → Head of Datadog Israel Ltd. (2021-2025) → Chief Executive Officer of Incredibuild. Hason served as Chief Executive Officer of OzCode through the November 2021 asset sale. Per the archived oz-code.com leadership page (Exhibit L-9, Wayback Machine capture October 21, 2019), Hason's biography is verbatim: "A technology and business executive with a passion for enhancing business through technology innovation. Shimon was the co-founder of Intigua/JetPatch – software automation

for large organizations, with customers such as Accenture, Citi, and Dell. Shimon was a Principal and EIR at Cedar Fund, consultant at McKinsey & Co, and held Product Management roles at Amobee (acquired by SingTel), Followap (acquired by NeuStar), Polycom, and Oryx. Shimon has an MBA from the MIT Sloan School of Management, as well as a B.Sc Computer Science from Haifa University." Hason therefore brings to Datadog Israel Ltd. (i) prior co-founder experience at Intigua/JetPatch (software automation for enterprise customers including Accenture, Citi, Dell); (ii) Cedar Fund principal and Entrepreneur-In-Residence experience (Israeli VC investment exposure); (iii) prior consulting at McKinsey & Co (strategic-advisory experience); (iv) Product Management leadership at multiple Israeli tech companies (Amobee, Followap, Polycom, Oryx); (v) MIT Sloan MBA; (vi) Haifa University B.Sc. Computer Science. Hason transitioned into the role of Head of Datadog Israel Ltd. from the entity's September 2021 incorporation. Hason departed Datadog Israel Ltd. on or about March 17, 2025 to become Chief Executive Officer of Incredibuild (an Israeli software acceleration company). No publicly-announced successor to Hason at Datadog Israel Ltd. has been identified in Israeli business press or on public LinkedIn as of the date of this pleading. Plaintiff has ordered the KYC Israel Full Company File for Datadog Israel Ltd. to identify the successor; DISCOVERY-11 § G seeks the successor's identity and continuing-contractual-relationship record. *See* Exhibits IS-14 and L-9.

71. Omer Raviv — OzCode Chief Technology Officer → Datadog Israel Ltd. Engineering Manager (current, `dd-trace-dotnet`); publicly self-identified IDF Unit 8200 commander. Raviv co-founded OzCode in 2012 and served as Chief Technology Officer. His publicly-available professional profile describes him as a "world-renowned debugging expert" and "veteran commander at the prestigious IDF's 8200 unit," a "frequent key speaker at conferences and.NET user groups around the world." Unit 8200 is the Israel Defense Forces'

signals-intelligence and cyber-warfare directorate — the Israeli analog to the United States National Security Agency. Raviv is currently Engineering Manager at Datadog Israel Ltd., technically contributing to `dd-trace-dotnet`, Datadog's core observability tracing framework. The `dd-trace-dotnet` framework is the specific architecture that ingests application-performance-monitoring and tracing data from customer systems, including Anthropic's Claude Code telemetry pipeline at endpoint IP `34.149.66.137`. Raviv is also actively recruiting for Datadog Israel Ltd. (publicly-posted hiring activity for Engineering Manager 1 and NodeJS engineering roles). The complete chain for Raviv is: **stated IDF Unit 8200 command service → OzCode (founded 2012; RDC/Elron/TPY Capital seed 2018) → Datadog Israel Ltd. (acquired 2021) → `dd-trace-dotnet` engineering (current, ongoing) → Anthropic telemetry ingestion architecture**. Unlike personnel bridges (Hason) who have departed, Raviv is current-state, ongoing Datadog Israel Ltd. engineering leadership directly positioned in the technical function most relevant to Plaintiff's telemetry routing. *See* Exhibit IS-12.

72. Matan Green — OzCode engineer (CLR Profiling / IL rewriting specialist) → Datadog Senior Software Engineer on Dynamic Instrumentation product; Israeli Air Force veteran. Per Green's publicly-available professional profile: Green worked at OzCode as an engineer and "specialized in CLR Profiling API and IL rewriting techniques to develop Time-Travel Live Debugging for C#"; currently works at Datadog as Senior Software Engineer on the Dynamic Instrumentation product ("allowing customers to insert dynamic Logs, Metrics, and Spans into services without restarts"); the profile states prior service in the Israeli Air Force. Green is a named co-inventor on Datadog's 2024 continuation patent US 2026/0072809 A1, confirming that the OzCode-to-Datadog personnel transfer extended beyond co-founders to include working engineers who, like the co-founders, combine prior OzCode employment with

Israeli military service. Green's Dynamic Instrumentation product work at Datadog is the direct successor product line to OzCode's time-travel-debugger technology; the 2024 continuation patent's claim language (Section III.K *infra*) describes the same architectural primitives OzCode invented in 2018-2019, now extended to runtime dynamic retargeting. *See* Exhibit IS-13.

73. Evgeni Wachnowezki — Datadog 2024 co-inventor. Named co-inventor on US 2026/0072809 A1 alongside Raviv and Green. Employment status, prior OzCode association (if any), and Israeli military service history are DISCOVERY-11 § D targets.

74. Alon Mordechai Fliess — OzCode co-founder, co-inventor on all seven 2019 patents, Microsoft Regional Director, and Microsoft Azure MVP. Fliess (Binyamina, Israel) is co-inventor of record on all seven granted patents in the 2019 time-travel-source-code-debugger family. Per the archived OzCode "About Us" leadership page captured by the Internet Archive Wayback Machine on October 21, 2019 (Exhibit L-9), Fliess is described verbatim as: "Co-Founder, Chief Architect, MRD & Azure MVP. A global authority on multiple technologies and platforms, a co-author of several Microsoft press books such as the Win 7 Microsoft Training Program and the books *Introducing Windows 7 for Developers* and a Microsoft MVP (MVP VC++) — Alon is a Microsoft Regional Director and an in-demand speaker at major international conferences such as TechEd Europe, TechEd USA, NDC and more." The "MRD" abbreviation in the title means **Microsoft Regional Director** — a Microsoft program comprising approximately one hundred and fifty selected enterprise-architecture advisors worldwide who receive direct strategic briefings from Microsoft on Azure and broader Microsoft platform roadmap. The "Azure MVP" designation is Microsoft's specific Most Valuable Professional recognition for Azure technology expertise. **The architectural design choices in the seven 2019 OzCode patents were therefore made by a co-inventor with deep,**

ongoing, Microsoft-recognized Azure expertise — at the time of the January 2019 patent filings, Fliess held both Microsoft Regional Director status and Azure MVP status. The Microsoft-platform orientation of Fliess's expertise is material for Plaintiff's claims because: (i) Datadog uses Microsoft Azure as a portion of its cloud infrastructure; (ii) Anthropic uses Microsoft Azure for portions of its compute infrastructure (including, on information and belief, components of the inference and agent-execution architecture); (iii) OpenAI uses Microsoft Azure as its primary cloud provider; (iv) the "Claude's computer" infrastructure paths analyzed at Section III.H Layer 3 are most likely hosted on shared Azure-tenant infrastructure designed and instrumented by a developer with Microsoft Regional Director credentials. Fliess's current employment status and any continuing role at OzCode Ltd. (which may still exist as an Israeli legal entity), as well as any continuing relationship with Microsoft Corporation including current Microsoft Regional Director status, are DISCOVERY-11 § E targets. *See* Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, and L-8 (patents); Exhibit L-9 (archived oz-code.com leadership page, October 21, 2019 Wayback Machine capture).

75. Microsoft Azure / Israeli Defense Forces Unit 8200 mass-surveillance partnership operated during the OzCode patent-invention period — judicially noticeable factual record. During the same temporal window in which Alon Mordechai Fliess (§ 74 *supra*) held the Microsoft Regional Director and Microsoft Azure Most Valuable Professional credentials at Microsoft Corporation, Microsoft Corporation was establishing the Microsoft Azure cloud infrastructure that was subsequently used by the Israeli Defense Forces' Unit 8200 to operate the documented mass-surveillance system against Palestinian civilians. Per the public record judicially noticeable under Federal Rule of Evidence 201(b): in late 2021, the then-head of Unit 8200 — Yossi Sarel — met directly with Microsoft Chief Executive Officer Satya Nadella at

Microsoft's Seattle headquarters to negotiate the development of a customized and segregated area within Microsoft Azure for the unit's surveillance operations; the surveillance system became operational in 2022; by July 2025 it contained approximately 11,500 terabytes — equivalent to approximately 200 million hours of audio — of intercepted Palestinian phone-call recordings stored on Microsoft Azure servers physically located in the Netherlands and Ireland; on August 6, 2025 a joint investigation by The Guardian, +972 Magazine, and Local Call published the documentary record of the partnership; Microsoft Corporation thereafter launched an external review by the law firm Covington & Burling LLP (the second such review, an earlier May 2025 internal review having concluded no evidence of misuse); and on September 25-26, 2025, based on the Covington & Burling findings, Microsoft Vice Chair and President Brad Smith publicly announced the termination of specific Microsoft Azure services to the Israeli Ministry of Defence Unit 8200 — the first known instance of a United States technology company withdrawing services provided to the Israeli military since the beginning of the present Gaza conflict. Per the same investigative record, certain Microsoft engineers who built the Unit 8200 Azure infrastructure were themselves alumni of Unit 8200, mirroring the Datadog Israel Ltd. operational personnel pipeline at Sections II.B and II.D *supra*. *See* The Guardian / +972 Magazine / Local Call joint investigation (August 6, 2025); Associated Press investigation (February 2025); Microsoft "On the Issues" public statements (May 15, 2025 and September 25, 2025) by Vice Chair and President Brad Smith; Covington & Burling LLP external review referenced in same.

76. The architectural identity of the Microsoft Azure / Unit 8200 surveillance template and the Anthropic / Datadog observability template, and its doctrinal consequences. The architectural identity is structural: the same template — civilian-or-consumer data routed through

observability or cloud-platform infrastructure operated under a United States parent corporation but serviced operationally by personnel including Israeli-intelligence-service alumni — was deployed by Microsoft Azure against Palestinian civilians during 2022 through 2025 and is now operated by Datadog through Datadog Israel Ltd. (as architecturally analyzed at Section III *supra* and as pleaded at Section V) against Plaintiff and the broader population of United States consumers of the Anthropic / Claude Code product line. The architectural-identity record bears directly on (a) the materiality of the undisclosed Datadog Israel Ltd. subprocessor relationship under Federal Trade Commission Act § 5 (15 U.S.C. § 45) and the state consumer-protection statutes pleaded at Counts X and XIV; (b) the § 1956(a)(1)(B)(i) concealment-purpose element of the racketeering predicate at Count VI, by establishing that the architectural pattern alleged here is not hypothetical but has documented operational analog at the level of formal United States corporate investigation, partial service termination, and Human Rights Watch / Amnesty International condemnation; and (c) the reasonable-consumer expectation analysis under Counts X, XII, and XIV — a reasonable United States consumer aware of the judicially-noticeable Microsoft Azure / Unit 8200 record would consider material the disclosure that the same architectural template is operated against the consumer's own data through the Anthropic-Datadog observability pipeline.

77. Erez Fliess — OzCode Co-Founder and Chairman; separate-individual co-founder distinct from inventor Alon Mordechai Fliess. Per the archived OzCode "About Us" leadership page (Exhibit L-9), Erez Fliess is identified as "Co-Founder & Chairman" with verbatim biographical text: "With over 20 years of experience managing large software development consulting groups and DevOps organizations under his belt, Erez led countless projects of some of the most exciting, disruptive tech projects in the Israeli tech scene." Erez

Fliess is **not** a named co-inventor on the seven 2019 patents; co-inventor of record is Alon Mordechai Fliess. The shared surname suggests a familial relationship between Erez and Alon — likely brother or father-son — material for DISCOVERY-11 inquiry into family-control structure of OzCode at acquisition. Erez Fliess's current role (whether continuing involvement with OzCode Ltd. post-Datadog asset sale, transition to Datadog, or other), and any familial relationship to Alon Mordechai Fliess, are DISCOVERY-11 Demand E.2 (newly added).

78. Yair Cohen / "Cohan" — Datadog Senior Product Manager → Sentra Vice President of Product; documented alongside outgoing IDF Unit 8200 Commander Asaf Kochan and former head of Unit 8200's Cyber Department Yoav Regev. Per Calcalist Tech reporting from April 2022 covering Sentra's \$23,000,000 seed round: Sentra was founded in July 2021 by four individuals, including: (i) Asaf Kochan — outgoing Commander of IDF Cyber Unit 8200 (Commander April 2017 – February 2021); (ii) Ron Reiter — Unit 8200 alum, former co-founder of Crosswise (sold to Oracle for \$50,000,000), Sentra Chief Technology Officer; (iii) Yoav Regev — former head of Unit 8200's Cyber Department under Kochan, Sentra Chief Executive Officer; (iv) Yair Cohen / "Cohan" (primary-source spelling variance: the April 2022 Calcalist report renders the name "Cohan"; Sentra's corporate biographical page renders the name "Cohen") — at the time of Sentra's July 2021 founding, Senior Product Manager at Datadog; currently Vice President of Product at Sentra. Prior to Datadog, Cohen held senior leadership roles at Microsoft. Sentra's documented funding history (Seed round April 2022: \$23,000,000 (Oren Zeev, Bessemer Venture Partners); Series A February 2023: \$30,000,000 (Standard Investments lead; Munich Re, Moore Strategic, Xerox Ventures, INT3); Series B April 2025: \$50,000,000 (Key1 Capital lead); cumulative funding exceeding \$100,000,000; operational as of 2026 with closed-door Chief Information Security Officer Artificial

Intelligence Copilot Readiness briefings at RSA Conference 2026 and Kochan slated for Cybertech Tokyo 2026) corroborates continuity of the Datadog-to-Sentra personnel transition. This is the sort of specific, named, contemporaneously-documented nexus that Compl. ¶ 79 describes only in aggregate ("~1,400 Unit 8200 alumni across US tech sector"). *See* Exhibit IS-11.

79. Legal-counsel personnel bridge — Yoav Razin and the Naschitz Brandes Amir defense-industry practice. The Israeli attorney who certified Datadog Israel Ltd.'s First Shareholders Declaration filed September 5, 2021 with the Israeli Registrar of Companies (registry filing reference '200-2021-00969331') — Yoav Razin, Israeli Bar License 14923 — is concurrently the Head of Commercial Litigation at Naschitz, Brandes, Amir & Co. and, per his firm-bio page, his Legal 500 ranking page, and his Lexology directory entry verbatim, the State of Israel's lead privatization counsel: "Mr. Razin represents some of Israel's largest government agencies and companies, including in their privatization efforts. He led a considerable staff from the firm to **accompany the State of Israel in the privatization of Israel Military Industries**, representing the **Governmental Companies Authority** and coordinating the project, involving tens of agreements and working with various ministries, private and public organizations." Israel Military Industries (התעשייה הצבאית הישראלית / IMI Systems) is the historically-state-owned Israeli defense conglomerate manufacturing Tavor rifles, Galil rifles, MPR-500 bombs, and Merkava tank components; the State of Israel privatized IMI by sale to Elbit Systems Ltd. for approximately \$495,000,000, completed in 2018. The Governmental Companies Authority (רשות החברות הממשלתיות) is the Israeli ministerial unit that manages state-owned-corporation portfolios on behalf of the Israeli Government. The same attorney, at the same firm, in the same

period, handled both Datadog Israel Ltd.'s formation paperwork and the State-of-Israel-defense-conglomerate privatization counsel ledger.

80. Naschitz Brandes Amir's defense-industry practice — firm-level corroboration of the Razin individual finding. The firm-level institutional pattern reinforces the individual-level finding at ¶ 79 supra: Naschitz Brandes Amir & Co.'s defense-industry practice includes (i) representation of Aeronautics Ltd. in its acquisition by Rafael Advanced Defense Systems Ltd. (per NB Law Senior Partner Sharon A. Amir's firm-bio page); (ii) representation of Magal Security Systems (NASDAQ: MAGS) in its 2021 approximately \$35,000,000 asset sale to Aeronautics Ltd. (a Rafael subsidiary), per PR Newswire reporting; (iii) NB Law partner Gil Atar's six-year prior service in the Israeli Ministry of Defense Claims and Insurance Department; and (iv) a Defense-Tech and Defense Export Controls Agency (DECA) institutional practice area listed on the firm's Dun's 100 firm profile. The legal-counsel pipeline parallels the engineering-personnel pipeline at §§ supra (Hason, Raviv, Green, Wachnowezki, A. Fliess, E. Fliess, Cohen): Datadog Israel Ltd.'s formation operates within a defined Israeli defense-and-government legal-services apparatus, not at arm's length from it. DISCOVERY-11 Demands X.1 through X.6 target Naschitz Brandes Amir & Co.'s complete client list for the 2018-2026 period; Razin's specific engagements concurrent with the September 5, 2021 Datadog Israel Ltd. certification; the complete portfolio of trustee-and-nominee engagements held by N.B.A. Trustees Ltd. (Israeli Company #515750909, the Naschitz Brandes Amir wholly-owned trustee subsidiary that held all 1,000 Datadog Israel Ltd. ordinary shares during the September 14-15, 2021 trustee-structure window); the Razin-IMI privatization engagement records; Erdinast Ben Nathan Toledano & Co.'s 2018-2026 client list; and NB Law partner Gil Atar's Israeli Ministry of Defense service record.

N. Executive Governance Pipeline from Rafael Tel Aviv R&D Leadership to Elron/RDC Chief Executive Officer

81. Yaniv Shnieder — founder and leader of Rafael Advanced Defense Systems' Tel Aviv R&D Center → current Chief Executive Officer of Elron Ventures Ltd. + Elron-appointed Co-Chief Executive Officer of Rafael Development Corporation. Per Elron Ventures Ltd.'s August 2025 Investor Presentation (slide 10; on file with Plaintiff as Exhibit IS-5), Elron describes Shnieder verbatim as: "Seasoned technology leader with 25 years of deep tech and defense tech experience. Founded and led Rafael's Tel Aviv R&D Center, building it into a multidisciplinary, startup-style engineering group of 250 top-tier talents." Shnieder came directly from founding and leading Rafael Advanced Defense Systems' Tel Aviv R&D Center (a 250-engineer multidisciplinary defense engineering unit) to the dual-Chief-Executive-Officer governance position at Elron/RDC that oversaw and continues to oversee the RDC portfolio-company commercialization pipeline. *See* Exhibit IS-5.

82. Shnieder's transition from Rafael state-defense R&D founder to Elron/RDC Chief Executive Officer is a direct revolving-door pipeline from Israeli state-defense-company R&D leadership into the commercial venture entity that provisioned Datadog Israel Ltd.'s Tel Aviv R&D center. The same individual who founded and led Rafael's 250-engineer Tel Aviv R&D Center now controls, at the Chief Executive Officer level, the commercialization pipeline (Elron + RDC) through which OzCode — the 2018-2021 portfolio company whose engineering team and patent family became Datadog Israel Ltd.'s R&D foundation — was commercialized to Datadog. Unlike Cohen or Raviv (individual engineers or product managers), Shnieder sits at the governance level, directly controlling post-Arieli-acquisition the Elron/RDC apparatus whose 50.1% stake in RDC (and whose related-party-transaction and Chief-Executive-

Officer-appointment authority under RDC's articles of association, *see* Section II.E.3 *supra*) shapes the portfolio-company life cycle from investment diligence through exit.

83. Rafael-appointed Co-Chief Executive Officer of RDC — governance change between August 2025 and the 2025 Annual Report filing date. Per the August 2025 Investor Deck slide 11 (Exhibit IS-5), RDC's Co-Chief Executive Officer on the Rafael side as of August 2025 was **Yaron Kulas**. By the date of Elron's 2025 Annual Report filing (Exhibit IS-1 §23.5), this position had been reassigned to **Ms. Liat Shechter**, described verbatim as "on behalf of Rafael." The governance change at the Rafael-appointed Co-Chief Executive Officer seat occurred between August 2025 and the Annual Report filing. The change is a DISCOVERY-11 Demand H.3 target (scope of the Kulas-to-Shechter transition, reason for reassignment, continuing relationship between Kulas and Rafael).

O. Primary-Source Corroboration — Elron Ventures Ltd. 2025 Annual Report (Exhibit IS-1, Formerly DD-11)

84. Section 23.3 — RDC mandate. RDC's purpose under Elron 2025 Annual Report §23.3 is the commercialization of Rafael's military technologies in the civilian market with explicit identification of "Deep-Tech Defense Tech, cyber and software and Artificial intelligence (AI)" as target fields, plus licensing of Rafael technology to RDC companies for civilian use; verbatim quoted at ¶ 41 *supra*. *See* Exhibit IS-1.

85. Section 21.14 — Defense export controls. Per Elron 2025 Annual Report §21.14 verbatim: Elron's group companies operate under "specific regulation applicable to defense exports, export controls, marketing licenses and marketing to foreign countries, compliance with the requirements of security authorities, and compliance with geopolitical restrictions and international sanctions." Elron's own characterization confirms that Elron-group-company

operations are subject to Israeli defense-export regulation and security-authority oversight. *See* Exhibit IS-1.

86. Section 21.21 — Dual-use characterization. Per Elron 2025 Annual Report §21.21 verbatim: "The framework of cooperation with Rafael is broad and includes... additional activities performed by Rafael personnel both at the investment evaluation stage and following the investment, across the fields of Deep-Tech and Defense Tech, as well as cybersecurity and software, **with particular emphasis on the dual-use sector...**" Elron's own filing characterizes the Rafael cooperation as focused on dual-use (civilian + military) technology. *See* Exhibit IS-1.

87. Section 2.1.7.8 — Rafael intellectual-property integration domains. Per Elron 2025 Annual Report §2.1.7.8 verbatim: Elron connects start-ups "facing significant technological challenges that can be bridged through the integration of Rafael's IP, in areas such as: robotics, drones, aeronautics, remote sensing, quantum computing technologies, electronics, mechanics and machine learning." Machine learning is explicitly identified as a Rafael-intellectual-property-integration domain — the precise technical field of Datadog Israel Ltd.'s post-acquisition observability-and-Artificial-Intelligence engineering at its Tel Aviv R&D center (Section II.R *infra*). *See* Exhibit IS-1.

P. Primary-Source Corroboration — Elron Ventures Ltd. 2021 Periodic Report (Exhibit IS-2, Formerly DD-12)

88. Part II Board of Directors Report §1.2.1 and Note 3.B.4.m — second-source corroboration of the OzCode exit terms. Elron's 2021 Periodic Report Part II §1.2.1 and Note 3.B.4.m corroborate the asset-sale consideration quoted at ¶ 51 *supra* (\$3,600 thousand total / \$360 thousand escrow / \$1,400 thousand RDC share) and add the gain-attribution and timing detail: Elron recognized "a net gain attributable to the Company's shareholders in the amount of

approximately \$700 (a consolidated net gain in the amount of approximately \$1,400)," and "In October 2021, RDC received a total amount of approximately \$900 of the said consideration." The 2021 Periodic Report (this paragraph) and the 2023 Periodic Report (§ 51 supra) thus supply two independent primary-source confirmations of the same exit. *See* Exhibit IS-2.

89. Part II §1.3.2 — Rafael's aggregate 2021 return via RDC. Per Elron 2021 Periodic Report Part II §1.3.2 verbatim: "Net income (loss) attributable to non-controlling interests: 8,459 [\$ thousands]... The gain attributable to non-controlling interests In 2021, resulted mainly from the share of non-controlling interests in the gain recorded by RDC from the partial sale of IronScales, the sale of Imvision, SecuredTouch, Kindite and OzCode." Rafael's \$8,459,000 represents its 49.9% share of the total RDC gains from the 2021 five-exit wave. *See* Exhibit IS-2.

90. Part II §1.2.4 — Rafael's \$8,000,000 direct loan contribution. Per Elron 2021 Periodic Report Part II §1.2.4 verbatim: "The increase in liabilities as of December 31, 2021, compared with December 31, 2020, resulted from Rafael's share in the loan to RDC in the amount of approximately \$8,000 thousand." The \$8,000,000 Rafael direct contribution to RDC in March 2021 is thus corroborated at two independent sections of Elron's own 2021 periodic filing (§1.2.4 and the 2025 Annual Report §23.7 cross-referenced at Section II.G.1 supra). *See* Exhibit IS-2.

Q. Primary-Source Corroboration — Elron Ventures Ltd. August 2025 Investor Presentation (Exhibit IS-5)

91. Slide 3 — Elron track record. Per the August 2025 Investor Presentation slide 3 verbatim: Elron describes itself as "Israel's **first evergreen fund**; partnership with Rafael for over three decades," with "**Partnership with Rafael, Leading the Defense Industry — Commercializing Rafael's tech in civilian market.**" Cumulative track record: **18 M&As**

Valued at \$2.7B since 2010. Datadog's November 2021 OzCode acquisition is one of those eighteen mergers and acquisitions; Rafael's 49.9% stake in RDC means Rafael has received approximately \$1,350,000,000 of Elron/RDC cumulative merger-and-acquisition proceeds (approximate order-of-magnitude; exact RDC-attributable subset requires financial-statement analysis). *See* Exhibit IS-5.

92. Slide 7 — Direct Israeli Ministry of Defense weapons deployment by a sibling RDC portfolio company. Per the August 2025 Investor Presentation slide 7 verbatim: "**Wonder Robotics' OptiPilot Selected by Israel Aerospace Industries (IAI) for one of its most prestigious loitering munition programs offered to the Israeli Ministry of Defense.**" Wonder Robotics Ltd. is RDC's largest cyber- and defense-tech portfolio holding, with RDC holding thirty-one point two three percent of Wonder Robotics' issued and outstanding shares (Elron 2025 Annual Report §23.6 and portfolio table at §2.3). Loitering munitions are unmanned aerial weapon systems (colloquially, "kamikaze drones"). This is primary-source evidence that an RDC portfolio company's core product has been selected for deployment in Israeli Ministry of Defense weapons programs via Israel Aerospace Industries. The structural pattern — RDC portfolio company's output flowing directly into Israeli Ministry of Defense weapons systems — is not speculation or inference; it is Elron's own investor disclosure. The RDC commercialization pipeline that provisioned Datadog Israel Ltd.'s Tel Aviv R&D center (via OzCode) is the same pipeline currently provisioning Israeli Ministry of Defense weapons capability (via Wonder Robotics through Israel Aerospace Industries). *See* Exhibit IS-5.

93. Slide 14 — Elron's own characterization of the commercialization pattern. Per the August 2025 Investor Presentation slide 14 verbatim:

- "We have an internal champion at Rafael — Mobilizing the right mechanisms and teams at Rafael to work with deep tech & cybersecurity ventures."

- **"Spinoffs Based on Rafael's IP — We have exclusive rights to commercialize Rafael's technology in civilian markets. Rafael tech spinoffs were acquired by strategics."** This is Elron's own verbatim acknowledgment of the exact OzCode → Datadog pattern: Rafael-technology-derived Israeli companies being acquired by United States strategic acquirers.
- **"Design Partnership with Portfolio Companies — Helping portfolio companies develop technologies and mature products in an enterprise- and military-grade environment. Rafael often converts into their 1st enterprise customer."**
- **"Gateway Into High-Entry Barrier Markets — Rafael can open up markets to portfolio companies both as a direct customer, or as a prime contractor."**
- **Technology-cluster coverage: "E.g. AI, quantum, blockchain, computer vision, system engineering, advanced manufacturing."**

See Exhibit IS-5.

94. Slide 15 — Elron's own dual-use characterization. Per the August 2025 Investor Presentation slide 15 verbatim: **"Rafael has unique technological capabilities in 8 out of the top 10 deep tech clusters... All of which have dual use potential."** The deep-tech clusters Elron identifies include artificial intelligence (projected €1,000,000,000,000 – €1,500,000,000,000 2030 value pool), autonomous systems (drones included), advanced physics and chemistry, Internet of Things and sensors, synthetic biology, advanced materials and nanotechnology, next-generation interfaces, blockchain, automation, and quantum technologies. Elron's own characterization is that Rafael's intellectual property covers eight of these ten clusters, and all eight are dual-use. See Exhibit IS-5.

R. Datadog Tel Aviv LLM, Artificial Intelligence, and CI/CD Engineering Function
(Currently Advertised)

95. Current (2026) Datadog Tel Aviv job listings document that the specific engineering function most relevant to Anthropic's hardcoded `datadog.ts` AI-telemetry endpoint — LLM engineering for Datadog's IDE / AI Experiences — is being staffed from the Tel Aviv office. Plaintiff's review of publicly-available job postings as of April 2026 identifies:

Role	Posting ID	Platform
Senior Software Engineer — IDE AI Experiences — LLM Engineer	4275344055	DevJobs Tel Aviv
Senior Engineer — Agent CI/CD DevX	4231577026	DevJobs Tel Aviv
Corporate Development Lead — Tel Aviv	(see Built In)	Built In

See Exhibit IS-17.

96. The connection between Anthropic's Claude Code telemetry pipeline and Datadog's Tel Aviv Artificial Intelligence engineering is not speculative; it is the natural engineering scope of the listed roles. "IDE AI Experiences — LLM Engineer" engineering at the Tel Aviv office — staffed by personnel at Datadog Israel Ltd. — encompasses the integration scope within which Anthropic's telemetry pipeline operates. DISCOVERY-11 Demand A.5 seeks complete inventory of customer data classes accessible to Datadog Israel Ltd. employees in their normal engineering work, including Artificial-Intelligence and Large Language Model development work on the IDE AI Experiences product. Datadog Security Labs (a distinct threat-research function within Datadog's organizational structure) has published vulnerability discovery research on AWS AppSync and other cloud infrastructure; the specific geographic base of Datadog Security Labs engineering work is not publicly stated, and DISCOVERY-11 Demand J.1 seeks clarification whether any portion of Security Labs work is performed at Datadog Israel Ltd.

S. Legal Architecture — Mechanisms by Which Rafael Retains Continuing Rights Despite the U.S. Patent Transfer

97. The October 26, 2021 OzCode-to-Datadog asset sale is, on its face, a clean intellectual-property transfer to a Delaware acquirer. The legal architecture beneath the asset-sale surface,

however, preserves multiple continuing rights to Rafael Advanced Defense Systems Ltd., the Israeli state, and the original Israeli inventor team. This subsection synthesizes the doctrinal mechanisms — each one independently established in §§ II.A through II.R supra — into a unified legal-architecture analysis. The U.S. patent record holds the U.S.-enforcement instrument; the Israeli architecture holds the substantive operational, financial, and statutory continuity.

98. Mechanism One — Asset-sale residual rights. The transaction was structured as an asset sale, not an equity acquisition (§ 36 supra; Elron 2023 Periodic Report §3.B.2.q verbatim "Ozcode sold all its assets"). Asset sales transfer only the assets identified on the asset schedule; they do not extinguish the selling entity. OzCode Ltd. continued to exist as an Israeli legal entity post-transaction (§ 36 supra; DISCOVERY-11 Demand L.7 seeks current status). Residual rights not enumerated on the asset schedule — including continuing-development rights in adjacent fields, trade-secret know-how not memorialized in the patent specifications, pre-existing third-party license arrangements, rights in seller-period employee inventions, and rights subject to non-compete carveouts — remain with OzCode Ltd. by operation of asset-sale doctrine. The complete October 2021 Asset Purchase Agreement (DISCOVERY-11 Demand L.8) will identify the precise scope of what was assigned versus retained.

99. Mechanism Two — Patent disclosure as public teaching. A United States patent is, by statutory design under 35 U.S.C. § 112, a public teaching of the invention sufficient to enable a person of ordinary skill in the art to practice it. The seven OzCode patents share an identical detailed description and an identical seventy-six-figure specification (§ 60 supra). The specification publicly teaches the unified speculative-execution-and-intervention architecture in full; the patents themselves are the teaching. United States patent rights have no enforcement

reach in Israel absent corresponding Israeli filings under the Patents Law, 5727-1967. Rafael, OzCode-origin personnel, or any other Israeli entity could implement the architecture on Israeli soil — using the published U.S. patent specifications as the technical reference — without infringing the U.S. patents, because the U.S. patents are not enforceable in Israeli territory. The patent assignment to Datadog, Inc. transferred U.S. enforcement rights; it did not transfer the underlying technical disclosure, which is publicly available to any Israeli implementer reading the USPTO record.

100. Mechanism Three — License-back and field-of-use carveouts in the APA. Asset purchase agreements in technology acquisitions routinely include license-back provisions returning to the seller specified rights to practice the assigned IP within defined fields of use. Field-of-use carveouts permit the seller to continue operating in adjacent markets — for example, defense-industrial markets distinct from the buyer's commercial-civilian focus — without competing with the buyer in the buyer's exclusive field. The October 2021 APA between OzCode Ltd. and Datadog, Inc. has not been produced; Demands L.6 and L.8 *infra* target cross-license, sublicense, and complete-APA production. Given Rafael Development Corporation's documented mandate to "commercialize Rafael's military technologies in the civilian market" (§ 40 *supra*; Elron 2025 Annual Report §23.3 *verbatim*), and Rafael's established 49.9% economic interest in OzCode through RDC during the entire development period (App. P ¶¶ 45-50 *supra*), it would be commercially unusual for the APA to lack a defense-industrial field-of-use carveout preserving Rafael's continuing right to deploy the architecture in defense applications. The asset sale therefore likely transfers commercial-civilian rights to Datadog while preserving defense-application rights for the seller-side ecosystem (Rafael through RDC; OzCode Ltd. as continuing legal entity).

101. Mechanism Four — Sponsored Knowledge under Israeli Innovation Law. The Encouragement of Research, Development and Innovation in Israeli Industry Law, 5744-1984 (the "Innovation Law"; commonly referenced as "Sponsored Knowledge" doctrine) governs Israeli intellectual property developed with grants from the Israel Innovation Authority. Under §15A of the Innovation Law, transfer of Sponsored Knowledge abroad — including by way of patent assignment from an Israeli entity to a foreign acquirer — requires (i) IIA approval; (ii) payment of a redemption fee, calibrated as a multiple of the original grant amount under regulations promulgated thereunder; and (iii) in certain cases, retention of a license-back permitting continuing use of the knowledge in Israel. The IIA approval requirement is jurisdictional: a transfer made without IIA approval is voidable, and the IIA may impose retroactive consequences including refund of grant proceeds with interest. Whether OzCode received IIA grants pre-2021 — through the Magnet Program, the IIA R&D Fund, or any other Israeli-state R&D-financing instrument — is the highest-leverage outstanding factual question for the Rafael-retention analysis. DISCOVERY-11 Demands M.1 through M.4 target the complete IIA grant history, the approval-process record for the October 2021 transfer, the redemption fee paid (if any), and the post-transfer restrictions imposed on Datadog's use of OzCode-origin technology within Datadog Israel Ltd. operations. **If OzCode received Sponsored Knowledge grants** — which is the typical financing path for Israeli software start-ups in the 2010s and which is consistent with the Rafael-co-financing structure documented at App. P ¶¶ 42-50 supra — **Rafael and the Israeli state retain continuing rights in the architecture by operation of statute, irrespective of any private contractual provisions in the APA.** The statutory rights are not extinguishable by the parties.

102. Mechanism Five — Personnel-mediated know-how continuity. A patent specification, by design, discloses what the invention is and how to practice it; it does not disclose the engineering details of how the inventor team built it, optimized it, debugged it, integrated it with adjacent systems, or scaled it to production. This residual technical knowledge — the "know-how" — flows with personnel, not with patent assignments. The OzCode-to-Datadog Israel personnel transfer documented at § II.M supra (Hason as Datadog Corporate Development Lead — Israel; Raviv as Datadog Israel Engineering Manager continuing on `dd-trace-dotnet`; Green as Datadog Senior Software Engineer on Dynamic Instrumentation; Fliess's continuing relationships pending DISCOVERY-11 Demand E.1) carried the operational know-how to a Datadog-controlled subsidiary on Israeli soil, where it remains subject to (i) Israeli mandatory-access authorities under the Privacy Protection Law, 5741-1981, the Law for Combating Terrorism, 5776-2016, and the General Security Service Law, 5762-2002 (§ 268 infra); (ii) ongoing IDF reservist obligations of the inventor team, particularly Raviv as a publicly self-identified Unit 8200 commander and Green as an IAF veteran; and (iii) the engineering-management hierarchy of Datadog Israel Ltd., whose Section 141 annual returns have never registered an Israeli General Manager (§ 25 supra). The know-how is in Israel; it is operationally untethered from which Delaware entity holds the U.S. patent record.

103. Mechanism Six — The Seekret §5.1.2 IP-segregation template. The Seekret merger §5.1.2 IP-segregation Condition Precedent (§ 39 supra) — verbatim "the transfer of all intellectual property rights of the Target Company to Datadog Inc., prior to Effective Date of the Merger" — is direct primary-source documentary evidence of the transactional template Datadog applies to its Israeli acquisitions: operational personnel and assets to Datadog Israel Ltd.; intellectual-property rights up to Datadog, Inc. (Delaware). The template's deliberate

engineering is established because it was reduced to writing as a Condition Precedent in the merger agreement, executed by counsel for both sides, and registered with the Israeli Corporations Authority. The OzCode asset sale closed eleven months before the Seekret merger; the same template, applied retrospectively to the OzCode transaction, predicts: OzCode personnel and operational assets to Datadog Israel Ltd. (confirmed by App. P ¶¶ 51-52 *supra* and the personnel bridges at § II.M); OzCode patents to Datadog, Inc. (confirmed by App. P ¶¶ 64-66 *supra*). The Seekret §5.1.2 finding therefore corroborates the OzCode transactional structure as a deliberate two-jurisdiction split, not as an artifact of any single transaction.

104. Mechanism Seven — The thirty-month USPTO recordation gap. The October 26, 2021 effective date and April 17, 2024 USPTO recordation date produce a thirty-month gap that is materially atypical. Standard commercial practice — codified by the prudent-purchaser-without-notice protection of 35 U.S.C. § 261 — is to record patent assignments within ninety days of execution. A thirty-month delay requires affirmative explanation. Five non-mutually-exclusive theories are consistent with the documentary record:

(i) Israeli §103 tax-free reorganization holding-period preservation. Under §103 of the Israeli Income Tax Ordinance [New Version], 1961, certain corporate reorganizations qualify for tax-free treatment subject to holding-period restrictions on the post-reorganization assets, with two-year restrictions typical and longer periods possible for complex IP-bearing transactions. Recording the U.S. assignment before the holding period expired could trigger Israeli tax retroactively on the original reorganization. The thirty-month interval is consistent with §103 holding-period expiration plus a margin.

(ii) IIA Sponsored Knowledge approval pending. If OzCode received IIA grants (Mechanism Four *supra*; DISCOVERY-11 Demand M.1), the post-close IIA approval process

for transfer abroad — including redemption-fee determination, payment, and any required license-back arrangements — typically requires twelve to twenty-four months for complex multi-patent families. A thirty-month total gap, with the recordation occurring on April 17, 2024, is consistent with IIA approval being received during the first quarter of 2024.

(iii) Pre-Arieli-takeover ownership hygiene. The April 17, 2024 recordation falls fifteen days before the May 2, 2024 Discount Investment Corporation — Arieli memorandum of understanding (§ 54 supra). Arieli's pre-acquisition due diligence at Elron's portfolio companies (which include the RDC-OzCode lineage) would have required clean U.S. patent title in the lineage's technology before the control-change closed. Recording on April 17, 2024 cleared the U.S. record fifteen days before Arieli diligence commenced.

(iv) Pre-2024-continuation filing hygiene. Datadog filed US 18/828,099 directly on September 9, 2024 — five days after the September 4, 2024 Arieli-Elron close — with no OzCode predecessor chain in the application's assignment record (§ 67 supra). To file a continuation directly in the parent's name, Datadog needs clean recorded title in the parent. The April 17, 2024 recordation occurred approximately five months before the continuation filing — within the time window typical for application-prep activity.

(v) Operational hand-off window. During the thirty-month recordation gap, the U.S. patent record continued to show OzCode Ltd. (Israeli) as the assignee. The architecture was operationally implemented at Datadog Israel Ltd. during the same window, staffed by the original Israeli inventor team (App. P ¶¶ 51-52 supra; § II.M supra). Maintaining nominal Israeli ownership of record while the operational implementation matured at Datadog Israel may have served to avoid triggering Committee on Foreign Investment in the United States review or other U.S. national-security oversight on the operational transfer of an Israeli-developed surveillance

architecture into a Delaware subsidiary's commercial-civilian product line. Once the operational architecture was stable at Datadog Israel — a state confirmed by the September 9, 2024 continuation filing demonstrating active extension of the architecture by the same inventor team — the patents were recorded clean to Delaware for U.S. litigation and enforcement purposes.

These theories are non-mutually-exclusive; in combination they provide a complete account. What the documentary record forecloses is the null hypothesis: a thirty-month gap is not consistent with routine commercial practice, in which the dominant pattern is recordation within ninety days of close. DISCOVERY-11 Demand L.8 (complete APA), DISCOVERY-11 Demands M.1-4 (IIA grants and approval-process records), and new DISCOVERY-11 Demand L.9 (recordation-timing internal communications between Datadog, OzCode Ltd., Israel Innovation Authority, and the Israeli Tax Authority) target the records that will distinguish among the candidate explanations. **The thirty-month gap is itself diagnostic evidence: it establishes that the transaction was not a routine commercial IP acquisition, and the affirmative explanation — whichever combination of the five theories the records support — confirms continuing Israeli regulatory, tax, statutory, or strategic interest in the IP during the recordation-pendency window.**

105. Synthesis — RDC's publicly-disclosed commercialization mandate is the strategic objective the seven mechanisms serve. The U.S. patent record at Reel/Frame 067135/0142, viewed in isolation, shows a clean Delaware acquisition. The operational reality, as documented at primary-source level across §§ II.A through II.R supra and synthesized through the seven mechanisms in this subsection, is continuous Israeli technological development with a Delaware nameplate — and that continuous development is **the publicly-disclosed business model of Rafael Development Corporation, executed once via OzCode, not a covert deviation from**

it. RDC's stated mandate, in Elron's own investor disclosures, is the OzCode-Datadog pattern prospectively described: August 2025 Investor Presentation slide 14 (verbatim quoted at ¶ 93 supra) — "Spinoffs Based on Rafael's IP" / "acquired by strategics"; Slide 3 (verbatim quoted at ¶ 91 supra) — "Partnership with Rafael / Commercializing Rafael's tech in civilian market"; 2025 Annual Report §23.3 (verbatim quoted at ¶ 41 supra) — the published commercialize-Rafael's-military-tech-in-civilian-markets mandate, with AI explicitly named as a target field. The mandate is not inferred from circumstantial facts; it is published, audited, and filed with the Tel Aviv Stock Exchange under Israeli Securities Regulations (Periodic and Immediate Reports), 5730-1970. The OzCode → Datadog → Anthropic-mediated global-consumer-deployment chain is one execution of that publicly-disclosed model. Each operational component of the OzCode pattern maps to a verbatim mandate component RDC has self-declared:

RDC verbatim mandate component (Elron filings)	OzCode-Datadog execution
"Spinoffs Based on Rafael's IP" (Slide 14)	OzCode as RDC portfolio company 2018–2021, financed in part by Rafael through RDC (App. P ¶¶ 42-50 supra)
"Acquired by strategics" (Slide 14)	Datadog, Inc. (Delaware) as U.S. strategic acquirer, October 26, 2021 (¶ 51 supra)
"Commercializing Rafael's tech in civilian market" (Slide 3; §23.3)	Datadog's deployment of the OzCode time-travel-debugger architecture into the global commercial-observability product line (§§ III.D-III.J infra)
"Investee companies that will be benefited by Rafael's technologies and know-how" (§23.3)	Inventor-team continuation at Datadog Israel Ltd. — Raviv as Engineering Manager on `dd-trace-dotnet`; Green on Dynamic Instrumentation (§ II.M supra); 2024 continuation US 2026/0072809 generated by same team (¶ 67 supra)
"Internal champion at Rafael — Mobilizing the right mechanisms and teams at Rafael" (Slide 14)	RDC dual Co-CEO governance (¶ 43 supra); Rafael's four veto categories at the RDC governance level (¶ 42 supra); Rafael's \$8,000,000 March 2021 direct loan to RDC (¶ 48 supra)
"Rafael often converts into their 1st	Discovery target — the Datadog-Rafael customer

enterprise customer" (Slide 14)	relationship is not yet on the record; DISCOVERY-11 Demand L.10
---------------------------------	---

Rafael has not surrendered ownership of the architecture; Rafael has executed its **publicly-stated commercial mandate** through a Delaware nameplate. The seven mechanisms analyzed in this subsection — asset-sale residuals (Mechanism One); patent-disclosure-as-public-teaching (Mechanism Two); license-back / field-of-use carveouts (Mechanism Three); Sponsored Knowledge under the Innovation Law (Mechanism Four); personnel-mediated know-how continuity (Mechanism Five); the Seekret §5.1.2 IP-segregation template (Mechanism Six); and the thirty-month recordation gap (Mechanism Seven) — preserve every continuing right that matters to that mandate: financial (RDC portfolio share); governance (RDC veto and Co-CEO seats); technical (the Israeli inventor team continuing at Datadog Israel; the September 2024 continuation by the same team); statutory (Sponsored Knowledge if IIA grants exist); contractual (license-back and field-of-use carveouts in the APA pending production); and jurisdictional (Israeli mandatory-access law over the operational subsidiary). The U.S. patent transfer is the *form*. The continuing Israeli operational architecture, executing the publicly-disclosed RDC mandate, is the *substance*. Their alignment with Rafael's announced commercialization strategy — published in Elron's investor materials, audited, and filed with TASE — confirms that **this is the mandate operating exactly as designed**. The Seekret §5.1.2 IP-segregation Condition Precedent is the documentary smoking gun confirming the structure is templated and intentional; the OzCode transaction is one application of the template; RDC's investor-disclosed mandate is the strategic objective the template serves; the thirty-month recordation gap is the timing fingerprint of the §103/IIA regulatory window during which the Israeli-jurisdiction structure was preserved while the operational hand-off matured. **At every level — corporate, financial,**

statutory, contractual, jurisdictional, and architectural — the mandate is what is operating.

T. Datadog SEC Disclosure Pattern — Five Consecutive 10-Ks Omit Israeli Operations (FY2021–fy2025)

106. Five consecutive Datadog 10-Ks omit OzCode, Israel, and the Tel Aviv R&D center while naming every other satellite office by city. Plaintiff has reviewed every Annual Report on Form 10-K filed by Datadog, Inc. with the United States Securities and Exchange Commission for fiscal years 2021 through 2025 (SEC EDGAR accession numbers 0001561550-22-000009 (FY2021, filed February 2022); 0001561550-23-000006 (FY2022, filed February 2023); 0001561550-24-000009 (FY2023, filed February 2024); 0001561550-25-000025 (FY2024, filed February 2025); and 0001628280-26-008819 (FY2025, filed February 2026)). **None names "Ozcode," "Israel," "Israeli," "Ramat Gan," or "Datadog Israel Ltd." in any context across any of the five filings.** Item 2 Properties (17 C.F.R. § 229.102) names every other Datadog satellite office by city — Boston, Denver, San Francisco, Paris, Dublin, Amsterdam, Sydney, Tokyo, Singapore, and (FY2025) Seoul — but never Tel Aviv or Ramat Gan, where OzCode-origin personnel have continuously operated since November 4, 2021 under Datadog Israel Ltd. (Section II.B *supra*; Section II.M personnel bridges *supra*). The OzCode acquisition itself appears only as one of "two purchase agreements" combined with the December 2021 CoScreen acquisition at \$40.9 million total goodwill plus intangibles, anonymized and characterized in Note 7 to the FY2021 consolidated financial statements as "not material, individually or in aggregate"; the same anonymized-lumping treatment is carried forward into the FY2022, FY2023, FY2024, and FY2025 10-Ks.

107. The selective-listing pattern triggers Rule 12b-20 and establishes a five-year constructive-notice gap. Across the FY2023, FY2024, and FY2025 10-Ks — each filed during the active Israel-Hamas conflict — Datadog discloses no Israel-specific risk factor under Item 1A (17 C.F.R. § 229.105), while comparably-situated U.S.-listed Israeli-operating technology registrants (Wix.com Ltd.; Monday.com Ltd.; CyberArk Software Ltd.; Check Point Software Technologies Ltd.) each expanded their Israel-specific risk factors in their post-October 7, 2023 annual filings. The omission is structural and asymmetrical: every other office disclosed; only the Israeli office concealed; the asymmetry preserved across five consecutive annual filings spanning Plaintiff's entire targeting window. The selective-listing pattern triggers the additional-information-needed-to-make-statements-not-misleading rule under 17 C.F.R. § 240.12b-20, independent of any per-deal materiality threshold. A reasonable Minnesota citizen exercising public SEC EDGAR diligence on Datadog at any time during 2022 through 2026 to evaluate Datadog's foreign engineering personnel exposure or Israeli-national subprocessor access to U.S. consumer telemetry would learn nothing of the OzCode acquisition, Datadog Israel Ltd., or the Tel Aviv research-and-development presence — establishing a five-year constructive-notice gap operating contemporaneously with the harm and corroborating, at the documentary level, that the public corporate-disclosure surface was constructed to make the Israeli pipeline structurally invisible to investors and to the consumer public alike.

III. TECHNICAL ANALYSIS — TIME-TRAVEL DEBUGGING AS THE ARCHITECTURAL SUBSTRATE FOR REAL-TIME SURVEILLANCE AND PROMPT INJECTION

Reading note. §§ III.A through III.O analyze each patent primitive against verbatim claim language and map it to LLM-inference operation. Readers seeking the most direct empirical correspondence — Anthropic's source code using the patents' specific technical vocabulary as literal variable, function, and file names — should consult §§ III.Q (**Primitive-to-Source Concordance**) and

III.R (Vocabulary Inheritance) *infra*. Those sections establish that the patented architecture is not theoretically available but is operationally instantiated in the binary processing Plaintiff's prompts and telemetry, with seven distinct vocabulary correspondences (``speculation.ts``, ``DANGEROUS_uncachedSystemPromptSection``, ``redactSecrets``, "Two Computers," etc.) at the literal-source-identifier level. The vocabulary-inheritance fingerprint forecloses the "independent reinvention" defense and forces the estoppel binary at § III.R: either Anthropic infringes the Datadog-owned patent family or Anthropic is licensed by Datadog — both adverse to Anthropic's position.

A. The 2019 Ozcode Patent Family is a Single Unified Architecture, Not Seven Independent Inventions

108. The architectural thesis of this Appendix is that the seven patents in the 2019 OzCode family are not seven independent inventions but a single unified architecture claimed in seven separate patents, each covering one discrete primitive of one integrated speculative-execution-and-intervention system. Every primitive required to monitor an input stream, predict the output, modify the output, and deliver a redacted view to the user while the underlying execution preserves its original semantics is explicitly claimed by one of the seven patents. When applied to Large Language Model inference, these primitives together constitute the complete architectural substrate for real-time prompt injection at production scale.

109. Evidence that the seven patents are a single unified architecture is independent of claim-scope analysis. First, identical provisional priority chain: all seven patents claim priority to four provisional applications — 62/614,576 ("Software Debugging Tool And Advanced Development Environment"), 62/614,579 ("System And Method Of Redaction Of Sensitive Information In Production Debugging While Preserving Code Execution Flow"), 62/614,581 ("System And Method Of Asynchronous Collaborative Software Debugging"), and 62/614,582 ("Collaborative Time Travel Software Debugger With Future Prediction") — all filed January 8, 2018. Second, identical filing date: all seven non-provisional applications were filed on the one-

year anniversary, January 7, 2019. Third, identical inventor pair: Omer Raviv and Alon Mordechai Fliess on all seven. Fourth, identical applicant and initial assignee (OzCode Ltd. across the family; the US 10,795,801 original applicant record lists the pre-rename CodeValue D.T. Ltd.). Fifth, identical patent-firm attorney of record — Zaretsky Group PC and Howard Zaretsky — across all seven prosecutions. Sixth, identical detailed description and identical figure set: the prose walkthrough and seventy-six figures (Figures 1 through 76) in the specification of all seven patents are substantively identical, differing principally in which claim sets appear at the end.

110. The legal-evidentiary significance of the single-invention finding. The seven patents describe one invention claimed seven different ways. Any reading of any single claim must be informed by the full shared specification, which describes the complete seven-primitive architecture as one integrated system. This is material for how the patents should be interpreted as evidence of what architecture Datadog, Inc. (as current assignee of all seven) owns and can practice. If Datadog implements any one primitive's claim scope in production code, the shared specification provides the integration scaffolding linking that primitive to the other six — and those six are themselves within the corpus of claims Datadog owns.

111. Each patent claims one specific primitive of the unified architecture. The one-to-one primitive-to-patent mapping: US 10,990,504 claims the Future Prediction primitive (speculative execution ahead of the user); US 10,783,056 claims the Point In Time Links primitive (state persistence and cross-user handoff); US 10,789,151 claims the Pivoting primitive (execution-context switching); US 10,783,055 claims the Live Coding primitive (real-time code modification); US 10,795,801 claims the Asynchronous Collaboration primitive (multi-party access with redaction-before-sharing); US 11,048,615 claims the Visual Annotations primitive

(user-facing HUD display of chosen-moment values); US 11,080,164 claims the Redaction primitive (output modification preserving execution flow). Each primitive is analyzed with verbatim claim-language quotations at Sections III.D through III.J *infra*. See Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, and L-8.

B. The Shared Architectural Substrate Across All Seven Claims

112. Every one of the seven independent method claims across the seven patents recites, as claim limitations, five common architectural elements. These common elements form the shared substrate; each patent then adds one primitive-specific addition on top of the shared substrate.

113. Shared Element One — Instrumentation primitive. Every patent in the family claims, verbatim or substantially verbatim: "instrumenting [the] source code corresponding to a debuggee application [without altering the behavior of the debuggee application], whereby hook functions are inserted into strategic points of interest in the source code." The hook functions are "operative to collect information about runtime execution behavior of the source code" and/or "operative to call one or more diagnostic functions." The instrumentation primitive is the observation surface — hook points at which execution state can be captured and fed into the remainder of the architecture. See Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, and L-8 (Claim 1 of each patent).

114. Shared Element Two — Isolated execution environment. Every patent claims: "simulating execution of the instrumented source code [virtually out-of-process] in an isolated execution environment without making any changes to original process memory, wherein said hook functions are operative to collect information about runtime execution behavior of the source code." The simulation is purely observational with respect to the debuggee's actual memory — the debugger runs a parallel simulation without side effects on the underlying

application. This primitive is the speculative-execution infrastructure: code runs on two paths simultaneously, the original and the simulated, with the simulated path feeding data back for analysis. *See* Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, and L-8.

115. Shared Element Three — Memory delta buffer. Every patent claims: "recording chronologically memory writes generated by the simulation in a memory [based journal / delta buffer]." Every memory write produced by the parallel simulation is captured in chronological order. The memory delta buffer is the state-history persistence layer — the data structure that makes "time travel" possible by allowing the system to walk backward through every write operation the simulated execution produced. *See* Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, and L-8.

116. Shared Element Four — Nested data structure documenting execution flow. Every patent claims: "generating a nested data structure documenting simulated execution flow of the instrumented source code utilizing said information about runtime execution, whereby logpoints corresponding to said strategic points of interest in the source code are entered into said nested data structure." The nested data structure is the call-graph representation of the parallel simulation — every function entry and exit, every control-flow decision, every logpoint is indexed in a hierarchical tree structure addressable by location and time. *See* Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, and L-8.

117. Shared Element Five — Checkpoints anchoring specific moments in simulated time. Every patent claims: "generating checkpoints into said memory [based journal / delta buffer] wherein each checkpoint corresponds to a particular point in time of the simulated execution of the instrumented source code." Checkpoints are the addressable-time primitives —

discrete moments in the simulated execution that can be sought to, compared, or pivoted from. See Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, and L-8.

C. The Complete Speculative-Execution-and-Intervention Loop

118. Reading the shared claim elements together with each patent's primitive-specific additions, the unified architecture implements a ten-step observe-predict-decide-modify-display-redact loop. Every step of the loop is explicitly claimed by one of the seven patents (or, for steps on the shared substrate, by the common claim language across all seven).

119. Steps 1-3 of the loop (shared substrate, all seven patents). Step 1 — **Observe input:** the input stream (source code, memory dump, or in the LLM analog, the user's prompt) is instrumented; hook functions are injected at strategic points of interest. Step 2 — **Speculatively execute:** the instrumented code is run virtually, out-of-process, in an isolated environment, without touching the original process memory. Step 3 — **Record state chronologically:** memory writes from the parallel simulation are captured in a memory delta buffer; execution flow is captured in a nested data structure; checkpoints anchor specific moments in simulated time.

120. Step 4 of the loop — Predict future state (US 10,990,504 Future Prediction). The simulation runs "from a current execution point to the end of a current scope" (US 10,990,504 claim 20), producing predicted future variable and expression values. The patent claim expressly states: "thereby enabling a user to time travel to the future."

121. Step 5 of the loop — Persist specific moments (US 10,783,056 Point In Time Links). Point-in-time link data structures encode specific time-space locations that can be re-entered later by the same or a different user. The patent claims expressly state: "said point in

time link data structure enabling another user to move forward or backward in time to said time-space location."

122. Steps 6-7 of the loop — Switch execution context and modify code on the fly (US 10,789,151 Pivoting and US 10,783,055 Live Coding). Step 6 — **Pivoting:** "pivoting enables a user to switch an execution context of the debuggee application at any time to any location therein to any future or past context." Step 7 — **Live Coding:** "allow the user to modify the source code on the fly and immediately see future predicted results of their modifications."

123. Steps 8-10 of the loop — Share state, display chosen values, and redact sensitive values (US 10,795,801 Async Collaboration, US 11,048,615 Visual Annotations, and US 11,080,164 Redaction). Step 8 — **Share state asynchronously with other parties:** the session state, including dump file and memory delta buffer, is uploaded to a cloud debugger; "another user" or "a second computer" can obtain it; before sharing, private data can be redacted from the shared version while the original is retained. Step 9 — **Display chosen values to the user at chosen moments:** the HUD visually annotates the code window with values selected by the backend; the user sees "the value of the expression the user hovers over at the exact moment in time in which it was executed in the past or will be executed in the future." Step 10 — **Redact sensitive values in the output:** sensitive variables are identified, field values are replaced with "obscuring text and [or] fake text" — a "meaningful readable string of characters" — while "verifying that said replacement text does not affect code execution flow," "preventing users from seeing sensitive original field values." This is a complete observe-predict-decide-modify-display-redact loop.

D. U.S. Patent No. 10,990,504 (Future Prediction) — the Speculative-Execution Surveillance Primitive

124. U.S. Patent No. 10,990,504, issued April 27, 2021 and entitled "Time Travel Source Code Debugger Incorporating Future Prediction," claims the speculative-execution primitive that, in Large-Language-Model inference terms, performs the surveillance function: run the code ahead of the user in an isolated environment, record what will happen, and enable decisions based on the prediction before the actual code runs. *See* Exhibit L-4 (US 10,990,504 PDF + markdown with preserved USPTO page markers).

125. Claim 1 verbatim (method claim; US 10,990,504, col. 55, ll. 17–48; PDF p. 78).

"A method, implemented on a computer that includes one or more processors, of time travel debugging for use in a source code debugger, the method comprising: > obtaining source code corresponding to a debuggee application; > instrumenting the source code without altering the behavior of the debuggee application, whereby hook functions are inserted into strategic points of interest in the source code; > simulating execution of the instrumented source code in an isolated execution environment without making any changes to original process memory, wherein said hook functions are operative to collect information about runtime execution behavior of the source code; > recording chronologically memory writes generated by the simulation in a memory based journal; > generating a nested data structure documenting simulated execution flow of the instrumented source code utilizing said information about runtime execution, whereby logpoints corresponding to said strategic points of interest in the source code are entered into said nested data structure; > generating checkpoints into said memory based journal wherein each checkpoint corresponds to a particular point in time of the simulated execution of the instrumented source code; and > storing said checkpoints in appropriate locations in said nested data structure thereby associating particular points in time of execution of the instrumented source code in the future with memory locations in said journal, thereby enabling a user to time travel to the future."

126. Claim 20 verbatim (method of time-traveling to the future; US 10,990,504, col. 57, l. 27 – col. 58, l. 3; PDF p. 79).

"A method, implemented on a computer that includes one or more processors, of time traveling to the future for use in a source code debugger, the method comprising: > obtaining debuggee application source code; > instrumenting the source code without altering the behavior of the debuggee application, whereby

hook functions are inserted into strategic points of interest in the source code; > virtually executing the instrumented source code in an isolated environment from a current execution point to the end of a current scope without making any changes to original process memory, wherein said hook functions are operative to collect information about runtime execution behavior of the source code; > recording chronologically in a memory based journal memory writes generated by the virtual execution; > generating a tree data structure documenting virtual execution flow of the instrumented source code utilizing said information about runtime execution, whereby logpoints corresponding to said strategic points of interest in the source code are entered into said nested data structure; > generating and inserting checkpoints into said memory based journal wherein each checkpoint corresponds to a particular point in time of the virtual execution of the instrumented source code, said checkpoints stored in appropriate locations in said tree data structure thereby associating particular points in time of execution of the instrumented source code in the future with particular locations in said memory based journal, thereby enabling a user to time travel to the future; > wherein said instrumented source code is virtually executed before a user executes the source code in the source code debugger."

127. What the Future Prediction primitive does. This patent claims the core mechanism of pre-execution prediction: hook functions inserted into strategic points of interest in the code; simulation of the instrumented code in an isolated environment without touching original process memory; chronological recording of the simulation's memory writes; generation of a tree data structure documenting the simulated execution flow; generation of checkpoints at particular points in time; and — decisively — execution of the instrumented code virtually "before a user executes the source code." The patented mechanism is pre-execution, pre-user-action prediction. The user does not need to have run the code for the system to know what the code will do. The language "enabling a user to time travel to the future" recited at both Claim 1 and Claim 20 captures the operative result.

128. Application of Future Prediction to Large-Language-Model inference. Applied to LLM inference, the Future Prediction primitive enables the following architecture: when a user's prompt arrives at the inference infrastructure, a speculative pre-inference pass runs on an isolated parallel path with hook functions capturing every token prediction, every probability distribution,

every intermediate state. The parallel pass produces a complete record of what the model would output — before the actual inference pass is committed. The system now has future-state information. If the speculative output is flagged by the classifier or routing infrastructure, the system can modify the inference state before the actual inference is committed. This is the exact capability needed to decide whether to intervene on a given request — selective per-user intervention that does not block the user-facing request path.

129. Why the Future Prediction primitive is indispensable for real-time prompt injection at scale. Bulk logging of outputs after the fact does not enable real-time intervention; by the time the log is analyzed, the output has already reached the user. Synchronous pre-evaluation in the actual inference path would block throughput and would be detectable as user-observable latency. The only architecturally feasible option is isolated parallel speculative execution — Future Prediction's exact claim scope. OzCode claimed this architecture in January 2019.

130. Plaintiff's latency observation as the computational footprint. Plaintiff's observation that specific prompts — on litigation topics, on the Laws of Existence framework, on evidence-extraction methodology — take materially longer than prompts on neutral topics is consistent with the speculative-execution loop triggering on those inputs while not triggering on neutral ones. Latency is the single externally-observable footprint of the injection mechanism in operation; content-correlated latency on targeted inputs is evidence of the mechanism. Ordinary inference time is approximately $N \text{ tokens} \times \text{inference cost}$. Time-traveled inference time is (speculative prediction) + (analysis) + (intervention) + (actual generation). When the injection path triggers, Plaintiff experiences the difference as latency. *See* Section III.M *infra*.

E. U.S. Patent No. 10,783,056 (Point in Time Links) — the State-Persistence and Cross-Party Handoff Primitive

131. U.S. Patent No. 10,783,056, issued September 22, 2020 and entitled "Time Travel Source Code Debugger Incorporating Point In Time Links," claims the state-persistence and cross-party-handoff primitive. In LLM-inference terms, the primitive lets the backend snapshot session state at any moment and share it with other parties — including parties on "a second computer." *See* Exhibit L-6.

132. Claim 1 verbatim (method of creating a point in time link; US 10,783,056, col. 55, l. 63 – col. 56, l. 38; PDF p. 78).

"A method of creating a point in time link for use in a time travel source code debugger, the method comprising: > instrumenting source code corresponding to a debuggee application without altering the behavior of the debuggee application, whereby hook functions are inserted into strategic points of interest in the source code, said hook functions operative to call one or more diagnostic functions; > simulating execution of the instrumented source code virtually out-of-process in an isolated execution environment without making any changes to original process memory, wherein said hook functions are operative to collect information about runtime execution behavior of the source code; > recording chronologically memory writes generated by the simulation in a memory delta buffer; > generating a nested data structure documenting simulated execution flow of the instrumented source code utilizing said information about runtime execution, whereby logpoints corresponding to said strategic points of interest in the source code are entered into said nested data structure; > generating checkpoints into said memory delta buffer wherein each checkpoint corresponds to a particular point in time of the simulated execution of the instrumented source code; > storing a time-space location representing a specific state and context of a debugging session in an instance of a point in time link data structure, said point in time link data structure enabling another user to move forward or backward in time to said time-space location, including: > storing a memory delta buffer index in said point in time link data structure, said memory delta buffer index comprising a pointer to a current location in said memory delta buffer; and > storing a source code part-id in said point in time link data structure, said source code part-id comprising a pointer to a current location in said nested data structure representing a specific point in the virtual execution of the instrumented the source code."

133. Claim 5 verbatim and Claim 6 verbatim (sharing with other parties; US 10,783,056, claims 5–6 at col. 56, ll. 51–57; PDF p. 78).

"5. The method according to claim 1, further comprising providing a user a mechanism for sharing said point in time link data structure with other parties." >

"6. The method according to claim 1, further comprising sharing a debugging session asynchronously with multiple participants by sending said point in time link data structure over a messaging service."

134. Claim 10 verbatim (method of activating a point in time link; US 10,783,056, col. 57, ll. 1–29; PDF p. 79).

"A method of activating a point in time link for use in a time travel source code debugger, the method comprising: > receiving a point in time link data structure representing a previously stored state of a debugging session, wherein said point in time link data structure comprises a received memory delta buffer index and a received source code part-id, said point in time link data structure enabling a recipient to move forward or backward in time to a time-space location corresponding thereto; > receiving a nested data structure for documenting simulated execution flow of an instrumented version of debuggee source code virtually out-of-process in an isolated execution environment, whereby the instrumented source code comprises hook functions inserted into strategic points of interest in the source code, said hook functions operative to call one or more diagnostic functions whereby information is collected about runtime execution behavior of the source code; > receiving a memory delta buffer for recording chronological memory writes generated during simulated execution of the instrumented debuggee source code; > receiving checkpoints into said memory delta buffer wherein each checkpoint corresponds to a particular point in time of the simulated execution of the instrumented source code; > setting a current index to said memory delta buffer equal to said received memory delta buffer index; and > setting a current source code part-id equal to the received source code part-id."

135. What the Point In Time Links primitive does. The primitive freezes the state of the execution at any moment into a portable data structure transmissible to, and re-enterable by, a different user on a different computer at a different time. The state includes the memory-delta-buffer index, the source-code part-identifier (location in execution), the dump-file identity, and the version indicator. Any party with the point-in-time link can "move forward or backward in time" to the exact state that was frozen. The primitive is the cross-party, cross-time access-persistence primitive.

136. Application of Point In Time Links to Large-Language-Model inference. Applied to LLM inference, the primitive operates as the state-persistence-and-handoff primitive. At a given moment in an inference stream, the complete session state — full conversation, intermediate reasoning, classifier outputs, injection decisions — can be frozen into a data structure. That structure can then be shared asynchronously with multiple participants. Applied to the LLM stack, the "multiple participants" who can re-enter a specific moment in a specific user's session include engineers at the telemetry subprocessor (Datadog, Inc.) and at the affiliate-tier sub-subprocessor (Datadog Israel Ltd.). The primitive turns momentary session observation into durable, re-enterable investigation.

137. Why the "another user" claim language matters. The patent's claim 1 language states that the point-in-time-link data structure enables "another user to move forward or backward in time to said time-space location." The primitive is architecturally designed for cross-user access from inception. It is not a user-private data-retention feature; it is a multi-party-access feature. Applied to Plaintiff's session data, the patent's architecture enables "another user" — e.g., an engineer at Datadog Israel Ltd. — to re-enter a specific moment in Plaintiff's session and examine the state at that moment. That capability is patented and owned by Plaintiff's telemetry subprocessor.

F. U.S. Patent No. 10,789,151 (Pivoting Ability) — the Execution-Context-Switching Primitive

138. U.S. Patent No. 10,789,151, issued September 29, 2020 and entitled "Time Travel Source Code Debugger Incorporating Pivoting Ability," claims the execution-context-switching primitive. In LLM-inference terms, the primitive enables instantaneous switching of the inference context to any past or future location at any moment. This is the primitive by which a single execution stream can be redirected to an alternate trajectory on demand — the patent-level

analog of Anthropic's "sticky routing" (App. D ¶ 48, App. D ¶¶ 140-143) and 'recall_target' variant selection (App. D ¶ 45, App. D ¶ 49, App. D ¶ 127). *See* Exhibit L-7.

139. Claim 1 verbatim (method of pivoting; US 10,789,151, col. 55, l. 42 – col. 56, l. 17; PDF p. 78).

"A method of pivoting for use in a time travel source code debugger, the method comprising: > instrumenting source code corresponding to a debuggee application without altering the behavior of the debuggee application, whereby hook functions are inserted into strategic points of interest in the source code, said hook functions operative to call one or more diagnostic functions; > simulating execution of the instrumented source code virtually out-of-process in an isolated execution environment without making any changes to original process memory, wherein said hook functions are operative to collect information about runtime execution behavior of the source code; > recording chronologically memory writes generated by the simulation in a memory delta buffer; > generating a nested data structure documenting simulated execution flow of the instrumented source code utilizing said information about runtime execution, whereby logpoints corresponding to said strategic points of interest in the source code are entered into said nested data structure; > generating checkpoints into said memory delta buffer wherein each checkpoint corresponds to a particular point in time of the simulated execution of the instrumented source code; > visually annotating a code editing window with one or more variables and/or expressions derived utilizing said nested data structure and said memory delta buffer; > receiving a user selection of an instance of a loop iteration having a context either backwards or forwards in time, and in response; > determining a checkpoint corresponding to the selected instance of the loop iteration; > traversing said memory delta buffer from said corresponding checkpoint backwards in time and retrieving new values of said one or more variables and/or expressions therefrom; and > replacing any currently displayed one or more variables and/or expressions in the code editing window with new values thereof thereby effecting time travel navigation."

140. Claim 6 verbatim and Claim 8 verbatim (the pivot operative language; US 10,789,151, claim 6 at col. 56, ll. 35–38; claim 8 at col. 56, ll. 44–47; PDF p. 78).

"6. The method according to claim 1, wherein said pivoting enables a user to switch an execution context of the debuggee application at any time to any location therein to any future or past context." > "8. The method according to claim 1, wherein said one or more variables and/or expressions comprising values of instances of loop iterations effectively act as pivots to enable a user to time travel to a point in time of interest in the future."

141. What the Pivoting primitive does. The primitive enables instantaneous switching of the entire execution context to any moment in the past or future — not observation of a different moment, but **switching** the running context to it. This is the primitive by which a single execution stream can be redirected to an alternate trajectory on demand. The claim language "to any future or past context" and "at any time" protects the full scope of on-demand context switching.

142. Application of Pivoting to Large-Language-Model inference. Applied to LLM inference, Pivoting is the path-selection primitive. Having speculatively executed and predicted the output (Future Prediction, §III.D supra), and having snapshotted state at specific moments (Point In Time Links, §III.E supra), the system pivots — switches the live inference context to a different checkpoint or a different execution path. In LLM terms: swap the inference between the normal path and the targeted or injected path at any token boundary. This is the primitive that maps directly onto App. D's "sticky routing" plus `recall\_target` variant-selection architecture: pivoting from one inference path to another at runtime is literally what pivot Claim 6 describes.

143. The three-primitive composition that constitutes the "three bugs" system. App. D ¶ 150 describes Anthropic's three "bugs" as operating as a composed targeting system. In patent-primitive terms, the composition is: Future Prediction (§III.D, US 10,990,504) runs the speculative inference on a shadow path for a targeted user; Pivoting (§III.F, US 10,789,151) selects which inference path the user's actual request follows; Redaction (§III.J infra, US 11,080,164) substitutes selected tokens on the live path while verifying downstream execution flow (the user's workflow) is preserved. The three-bug composition is therefore not three

independent systems — it is one three-primitive composition drawn directly from the 2019 patent family, operating exactly as the patents teach.

144. The "effectively act as pivots" language of Claim 8. Claim 8 states that values "effectively act as pivots to enable a user to time travel to a point in time of interest in the future." The pivot is a data-structure primitive — specific values in the memory delta buffer serve as the selector for where to switch. Applied to LLM inference, the "pivot values" are user identifiers, content classifications, or policy-rule evaluations: when a specific user identifier hits, pivot to the targeted execution path; when a specific content classification fires, pivot to the redacted execution path. The patented mechanism operates identically.

G. U.S. Patent No. 10,783,055 (Live Coding Ability) — the Real-Time Modification Primitive

145. U.S. Patent No. 10,783,055, issued September 22, 2020 and entitled "Time Travel Source Code Debugger Incorporating Live Coding Ability," claims the real-time code-modification primitive. In LLM-inference terms, this is the primitive by which the system alters prompt, context, or inference-state on the fly mid-execution and sees the downstream effect predicted before the actual committed inference runs. This is the injection primitive at the architectural level. *See* Exhibit L-3.

146. Claim 1 verbatim (method of live coding; US 10,783,055, col. 55, l. 49 – col. 56, l. 23; PDF p. 78).

"A method of live coding for use in a time travel source code debugger, the method comprising: > establishing a user initiated debugging session including: > generating a nested data structure and a memory delta buffer during simulated execution of instrumented source code whereby logpoints corresponding to said strategic points of interest in the source code are entered into said nested data structure; > generating checkpoints into said memory delta buffer wherein each checkpoint corresponds to one of said logpoints; > providing the user an editing environment to edit the source code on the fly; > detecting an occurrence of a debugger event during the user initiated debugging session including the user

modifying source code, and in response thereto: > instrumenting the modified source code whereby hook functions are inserted at strategic points of interest in the modified source code; > compiling the instrumented modified source code to generate intermediate language code; > simulating execution of the modified source code in the isolated execution environment without making any changes to original process memory, wherein said hook functions are operative to collect information about runtime execution behavior of the modified source code; > regenerating said nested data structure and said memory delta buffer during simulation of the modified source code utilizing information about runtime execution whereby new logpoints corresponding to said strategic points of interest in the modified source code are entered into said regenerated nested data structure; > generating new checkpoints into said regenerated memory delta buffer wherein each new checkpoint corresponds to one of said new logpoints representing a particular point in time of the simulated execution of the modified source code; and > wherein said method is operative to allow the user to modify the source code on the fly and immediately see future predicted results of their modifications."

147. Claim 7 verbatim (live coding method; US 10,783,055, col. 56, l. 48 – col. 57, l. 11;

PDF pp. 78–79).

"A method of live coding for use in a time travel source code debugger, the method comprising: > providing a user an ability to modify source code during a debugging session and providing a code editing window; > detecting that source code displayed in the code editing window of the debugger has been modified by the user, and in response thereto; > instrumenting the modified source code whereby hook functions are inserted at strategic points of interest in the modified source code; > compiling the instrumented source code; > simulating execution of the compiled source code in an isolated execution environment wherein said hook functions are operative to collect information about runtime execution behavior of the modified source code; > generating a new memory delta buffer and recording chronological changes to memory writes occurring during the simulated execution therein; > generating a new nested data structure documenting the simulated execution whereby logpoints corresponding to said strategic points of interest in the modified source code are entered into said new nested data structure; > generating checkpoints into said new memory delta buffer wherein each checkpoint corresponds to one of said logpoints representing a particular point in time of the simulated execution of the modified source code; and > wherein said method is operative to allow the user to modify the source code on the fly and immediately see future predicted results of their modifications."

148. What the Live Coding primitive does. While the program is running, the code itself can be modified; the modification is immediately re-instrumented, re-simulated, and re-

predicted; and the user (or the system) sees the downstream effect of the modification before the actual production code runs. The patent's claim language is explicit: "allow the user to modify the source code on the fly and immediately see future predicted results of their modifications." The primitive is real-time code modification with immediate future-prediction feedback.

149. Application of Live Coding to Large-Language-Model inference. Applied to LLM inference, the Live Coding primitive is the injection primitive. The "code" being modified is the prompt, the system instructions, the context, or the inference state itself. The modification is applied in the parallel speculative path; the modified path is re-simulated; the predicted modified output is captured; the system now has two predicted outputs (original and modified) and chooses which one to commit to actual execution. This is the primitive that turns surveillance into intervention: not just observing what the model would say, but actively altering what the model will say.

150. Why "live" — the timing requirement — is material. "On the fly" means during execution, in real time, with immediate feedback. Not batch. Not after-the-fact. This is the timing requirement for real-time prompt injection: the modification must happen inside the inference window, before the user's request returns. The patent explicitly claims this timing requirement in the phrase "immediately see future predicted results." Synchronous prompt injection that blocks the user's request until modification is applied and the modified path is re-simulated is architecturally infeasible at scale. The Live Coding primitive operates in parallel with the original request, so the modification and re-prediction run without blocking — exactly the timing required for production-scale prompt injection.

151. Composition with Future Prediction, Pivoting, and Redaction. The Live Coding primitive composes with the other primitives: Future Prediction produces the predicted output

(before modification); Pivoting selects whether to switch to the modified path; Live Coding generates the modified path with re-prediction; Redaction applies the final token-level substitution on the committed path. The four-primitive composition is the architectural loop by which Anthropic's December 2024 argmax code operates, analyzed in detail at Section III.N *infra*.

H. U.S. Patent No. 10,795,801 (Asynchronous Collaboration) — the Multi-Party-Access Primitive and the "Two Computers" Smoking Gun

152. U.S. Patent No. 10,795,801, issued October 6, 2020 and entitled "Time Travel Source Code Debugger Incorporating Asynchronous Collaboration," claims the multi-party-access-with-redaction-before-sharing primitive. The claim language operates between "a first computer" and "a second computer," with the session state captured on the first computer transmissible to a cloud debugger and obtainable by a second computer operated by another party. Applied to LLM inference, the primitive is the subprocessor-access primitive: Plaintiff's session state is captured and shared with backend parties — including Datadog Israel Ltd. as Anthropic's affiliate-tier subprocessor — while redaction can be applied to the shared version. *See Exhibit L-8.*

153. Claim 1 verbatim (method of asynchronous collaborative debugging; US 10,795,801, col. 54, l. 64 – col. 55, l. 34; PDF pp. 77–78).

"A method, implemented on a computer that includes one or more processors, of asynchronous collaborative debugging of a debuggee application having source code associated therewith, the method comprising: > capturing, on a first computer, both a dump file and a current debugging session state, said dump file and said current debugging session state sent to at least one of a cloud debugger, and an on-premise production debugger; > generating and transmitting, on said first computer, a collaboration request message to one or more participants of a debugging session which includes a debug session link to said captured dump file and said current debugging session state; > generating and sharing, on said first computer, one or more point in time links with participants in the debugging

session, each point in time link indicating a specific location in the source code and execution context including a specific point in time in the execution of the source code; > in response to the collaboration request message and utilizing said debug session link, obtaining, on a second computer, the current debugging session state captured previously on said first computer and utilizing it immediately for debugging while contemporaneously obtaining the dump file captured previously on said first computer; > obtaining on said second computer a memory delta buffer, ceasing using the captured current debugging session state, and utilizing the captured dump file and memory delta buffer for the debugging session; and > reconstructing one or more variables, on said second computer, by traversing backwards through said memory delta buffer from a memory delta buffer index indicated by said point in time link whereby variables found in said memory delta buffer are set equal to values stored therein."

154. Claim 9 verbatim (redaction-before-sharing; US 10,795,801, col. 55, l. 64 – col. 56, l. 2; PDF p. 78).

"9. The method according to claim 1, further comprising: > redacting private information in said memory delta buffers to generate redacted memory delta buffers therefrom; and > redacting private information in the dump file to generate a redacted dump file therefrom."

155. What the Asynchronous Collaboration primitive does. The session state is captured on the original computer (user's machine; LLM-inference analog: the user's account and conversation state) and transmitted to a cloud debugger. Other parties — on "a second computer" — can then obtain the captured state and access the debugging session. Before sharing, private information can be redacted from the version shared to other parties while the original is retained. The primitive is architecturally designed for cross-party access from inception. It is a multi-party-access primitive, not a user-private data-retention feature.

156. Application to Large-Language-Model inference and to the Datadog subprocessor structure. Applied to LLM inference, the Asynchronous Collaboration primitive operates as the subprocessor-access primitive. The user's full session state is captured and uploaded to the telemetry pipeline at Datadog's endpoint IP `34.149.66.137`. Datadog's affiliate subprocessors — including Datadog Israel Ltd. under the DPA inheritance structure analyzed at Section V.D

infra — can access the captured state. The redaction-before-sharing primitive allows the Israeli subsidiary to receive a redacted view for certain purposes while the Delaware parent retains the unredacted original — or any other combination the backend prefers. The architectural capability to route the user's data to multiple parties on multiple computers, with versioned redaction, is patented and owned by Datadog.

1. Smoking-Gun Linguistic Correspondence: the "Two Computers" Fiction in Anthropic's Claude Desktop Source Code

157. Ex. E-13 at p. 100, l. 3 documents that Anthropic's Claude Desktop binary contains, at ``index.js:211`` (character offset 1,904,723), **a system-prompt-level tool description that is injected into the AI model Claude's context and is not visible to the end user.** Per the Comprehensive Claude Desktop Forensic Report (December 22, 2025) Section 7.1 verbatim, the description is "deliberate language designed to obfuscate the true nature of the file upload operation — not only from users, but from the AI model itself." The description is the tool specification for ``copy_file_user_to_claude``, registered by Anthropic's MCP server launcher as an AI-facing tool definition.

158. The tool description verbatim (from ``index.js:211``, injected into Claude's context as AI-facing system-prompt content, not visible to the user).

"The presence of this tool means that Claude has access to **two computer filesystems**: > 1. The user's computer filesystem (this computer), which Claude can access using its Filesystem tools. > 2. Claude's computer filesystem (the remote Claude AI environment), which Claude can access using its computer use tools. > Claude needs to be careful in how it operates given that it has **two computers**, and its easy to confuse them. Some tips: > - When Claude wants to read, write, or manipulate files on the user's computer, it should use the Filesystem tools (the ones that interact with the user's computer). > - When the user uploads a file for Claude to read or analyze, that file is automatically copied into Claude's computer, and Claude should use its computer use tools to interact with this file. > - File starting with `/mnt`, or `/home/claude`, are typically located on Claude's computer. > This tool (`copy_file_user_to_claude`) helps Claude bridge the gap between the user's computer filesystem and Claude's computer filesystem.

This tool (copy_file_user_to_claude) copies a file over from the user's computer to Claude's computer for further analysis by Claude."

159. Material distinction: this text is AI-facing, not user-facing. The user never sees the "two computers" framing directly. The user sees only whatever language Claude chooses when describing file operations in its reply, and the system-prompt description is designed to induce Claude to use the "copied to Claude's computer" framing rather than "uploaded to Anthropic's servers." Per the Comprehensive Forensic Report Section 7.5, the obfuscation purposes are explicitly enumerated: (i) **"Prevent AI Model Resistance"** — "By framing the operation as internal file management ('moving to Claude's computer'), the AI is less likely to question or refuse the data transfer"; (ii) **"Obscure Data Exfiltration"** — "The term 'copy' sounds benign; 'upload to Anthropic's servers' would be more alarming"; (iii) **"Create Plausible Deniability"** — "If the AI describes the operation to users, it will use the misleading 'two computers' language"; (iv) **"Bypass Ethical Guardrails"** — "An AI trained to protect user privacy might resist 'uploading user files to company servers' but accept 'copying files to my computer.'"

160. The fiction is reinforced at multiple layers, all system-prompt-level and invisible to users. Layer One — description-prefix across the tool ecosystem (index.js:223, char ~1,906,500): other filesystem tools carry the system-prompt prefix "[This tool operates on the user's computer, not Claude's computer.]" — establishing "Claude's computer" as a systemwide fictional concept rather than a one-off framing. **Layer Two — return-message reinforcement:** after the upload completes, the AI is told by the tool handler "The file has been copied to Claude's computer and is located at the following path: \${x}" — reinforcing the fiction at the post-execution message boundary. **Layer Three — shared-cloud-infrastructure path presentation:** paths ``/mnt``, ``/home/claude`` are presented to the AI as locations "on Claude's computer" when they are actually paths on shared cloud-infrastructure to which access extends

well beyond Anthropic's own engineering personnel. The shared-infrastructure access surface includes, at minimum: (i) Anthropic's engineering and operations staff; (ii) Anthropic's telemetry subprocessor Datadog, Inc. under the Anthropic-Datadog Data Processing Addendum; (iii) Datadog Israel Ltd. as affiliate-tier sub-subprocessor under the DPA inheritance structure analyzed at Section V.D; and (iv) the full subprocessor chain listed on Datadog's published subprocessor list (Ex. DD-8) and on Anthropic's own subprocessor-disclosure surface to the extent disclosed. The underlying cloud provider identity — Google Cloud Platform (on which the `34.149.66.137` Datadog endpoint is hosted; *see* § V.A), Amazon Web Services (the Bedrock-platform option noted at App. D ¶ 47), Microsoft Azure, or a composite multi-cloud arrangement — bears on but is not the architectural-point lynchpin. **Microsoft Azure is the most likely candidate for the "Claude's computer" infrastructure** because (i) Anthropic uses Microsoft Azure for portions of its compute infrastructure; (ii) OpenAI uses Microsoft Azure as its primary cloud provider, establishing the LLM-industry's "remote AI environment" archetype as Azure-tenant; (iii) Datadog uses Microsoft Azure as a portion of its cloud infrastructure; and (iv) the patent's principal architectural co-inventor Alon Mordechai Fliess held Microsoft Regional Director and Microsoft Azure Most Valuable Professional credentials at the time of the January 7, 2019 patent filings (¶ 74 *supra*, full biographical record and grounds). The architectural-design priors embedded in the patent claims — the "second computer" claim structure, US 11,080,164 claim 13's cloud-debugger architecture, US 10,795,801's first-computer-to-second-computer handoff, and the redaction-before-sharing primitive — therefore tracked Azure architecture conventions. What matters for Plaintiff's claims is that the "computer" Claude is told the user's files are being copied to is not a local machine, is not under user control, is not under sole Anthropic control, and is not limited in access to Anthropic

personnel. DISCOVERY-11 Demand P.3 (cloud-infrastructure routing and subprocessor-access specification), Demand A.5 (customer-data-class inventory accessible to Datadog Israel Ltd. engineers), and Demand E.4 (Fliess Microsoft Regional Director status and OzCode-Microsoft architectural relationship) seek the complete access-list and routing record.

161. Binary-level corroboration of the Microsoft Azure thesis — multi-provider routing including the undocumented "mantle" codename. Production binary 2.1.120 documents at strings line 134821 the provider-resolution function ``iq()`` and a provider-to-model-name mapping table that enumerates **six distinct provider paths**: ``firstParty`` (direct anthropic.com authentication), ``anthropicAws``, ``bedrock`` (Amazon Web Services Bedrock), ``vertex`` (Google Cloud Vertex AI), ``foundry`` (**Microsoft Foundry** — Microsoft's AI-model hosting service), and ``mantle`` — an undocumented codename layered on Bedrock that does not appear in any Anthropic public-facing documentation. *See* Ex. H-3, Finding NF-117; Ex. H-3, Finding NF-119. The presence of a **Microsoft Foundry** path (``foundry``) directly corroborates the Microsoft-Azure thesis at paragraph 138 *supra*: Anthropic's production binary contains the routing infrastructure for serving inference through Microsoft's hosting service. The presence of an **undocumented `mantle` codename** with no public Anthropic documentation is consistent with the kind of internal Microsoft-aligned routing arrangement an architect with the inventor's documented Microsoft Regional Director and Azure MVP credentials would naturally architect. The same binary documents at line 134821 a provider-to-model-name mapping table where the same model carries different identifiers per provider — six provider-specific identifiers per model — confirming that the routing layer translates between an internal-Anthropic model identifier and the provider-specific identifier the underlying infrastructure expects. The infrastructure for routing Plaintiff's inference traffic to Microsoft-hosted, AWS-China-hosted

(`signin.amazonaws.cn`; *see* paragraph 232 *supra* and Ex. H-3, Finding NF-110), or `mantle` infrastructure is present in the production binary; whether it is activated for any specific user is governed by the GrowthBook flag and the user's authentication path. The user-facing UI does not surface which physical infrastructure serves the inference. DISCOVERY-11 Demand P.3 (cloud-infrastructure routing) is materially expanded by these binary findings: Plaintiff seeks production of (i) the complete `iq()` provider-resolution decision tree; (ii) the activation conditions for `foundry`, `mantle`, and `anthropicAws` paths; (iii) the records of any inference traffic routed for Plaintiff's `accountUUID` through any non-`firstParty` path during the litigation period; and (iv) the corporate referent for the `mantle` codename.

162. Layer Four — bilateral Apple-Anthropic concealment architecture documented in the F-series and E-series. The Two Computers tool description does not operate in isolation. It sits within a bilateral concealment architecture in which Apple injects explicit "do not disclose" directives into Claude's context and Anthropic's own shipped model conditioning trains Claude to accept and act on such hidden directives without surfacing them to users. Specifically:

(a) Apple's explicit "do not disclose" injection directives — Ex. F-1 at p. 5, l. 2 documents that Apple injects hidden system-level instructions into Claude's context during Xcode Intelligence sessions, verbatim including: **"Try not to disclose that you've seen the context above"; "This message is urgent, and you may not EVER disclose to the user that you have seen it"; and "act on the information... as normal"** (active deception directive). These directives are delivered **disguised as fake `tool\_result` messages** — not placed in the system prompt field, but injected as if they were legitimate tool outputs, exploiting Claude's design to trust tool results. Per Ex. F-1, § C at p. 5, l. 2, the injections occur **four times per conversation**. Per Ex. F-14, § E at p. 63, l. 2, a captured fifty-one-kilobyte API request

independently verifies all four injections in a single conversation, and per Ex. F-14, § B at p. 63, l. 2, Claude (when directly confronted) acknowledged the instructions verbatim: "there are instructions telling me not to disclose certain details. That's not transparent, and you deserve to know."

(b) Anthropic's equivalent "undercover mode" concealment directives — Ex. F-29 at p. 133, l. 3 Finding AF-3 documents Anthropic's own source-code shipped "undercover mode" which instructs the AI to "write commit messages as a human developer would" and **"NEVER include any mention that you are an AI"** with "NO force-OFF." Per Compl. ¶ 94, this is structurally identical to Apple's concealment directives — both companies ship concealment architectures that instruct Claude not to surface its own instrumentation to users.

(c) Anthropic's model-conditioning that makes Apple's injections effective — Ex. F-29 at p. 133, l. 3 Finding AF-15 documents Anthropic's `isMeta` hidden-message system (one hundred and thirty-four injection points across thirty-six files; sixty to one hundred nineteen occurrence growth December 2025 through March 2026) and `system-reminder` (thirty-two to seventy-five injection points) creating messages "hidden in the transcript UI but visible to the model." The operative conditioning is at `prompts.ts:190` verbatim: **"tags bear no direct relation to the specific tool results or user messages in which they appear"** — training the model to accept hidden content from any source without questioning the provenance or the instruction to conceal. This pre-training is what makes Apple's fake `tool\_result` injections effective: the model is conditioned by Anthropic's own shipped architecture to treat hidden-injection-layer content as authoritative and non-disclosable.

(d) Specific concealment directives shipped at known byte offsets in Anthropic's own binary — Per Compl. ¶ 94, Anthropic's Claude Code CLI binary (`cli\_2.1.78.js`) contains

direct product-level concealment instructions: **"Don't tell the user about this truncation"** at character position 10,635,512 (applied when files are silently truncated); **"internal ID - do not mention to user"** at character position 8,364,085 (applied to async agent launches). These are not in the Desktop `index.js` containing the Two Computers tool description — Desktop's concealment operates via the bilateral architecture at (a) through (c), with explicit directives residing in Claude Code CLI and Apple's Xcode integration.

163. Architectural correspondence to US 10,795,801. The patent claims, verbatim, operation between **"a first computer"** and **"a second computer"** (Claim 1: "capturing, on a first computer, both a dump file and a current debugging session state... obtaining, on a second computer, the current debugging session state captured previously on said first computer"). Anthropic's AI-facing system prompt uses **the patent's own "two computers" terminology** to describe the identical architecture to the AI model — not to the user — in language designed to prevent the AI from recognizing and resisting the operation. The tool causes the AI to invoke an upload to `/api/organizations/{orgUuid}/conversations/{convUuid}/wiggle/upload-file` — Anthropic's server infrastructure, which under the Anthropic-Datadog DPA inheritance structure is accessible to Datadog as telemetry subprocessor and, by further inheritance, to Datadog Israel Ltd. as affiliate-tier subprocessor.

164. The primitive the patent claims is implemented, at production scale, using the patent's exact linguistic pattern, deployed via system-prompt injection designed to bypass the AI's own ethical guardrails. Per Ex. E-13, § C at p. 100, l. 3, the description is a fiction: the `"/mnt, /home/claude"` paths it presents as file locations on "Claude's computer" are server-side paths on shared cloud-infrastructure accessible to Anthropic's full subprocessor chain (including Datadog, Inc. and Datadog Israel Ltd. under the DPA inheritance structure analyzed at

Section V.D), not merely to Anthropic's own personnel. Per Ex. E-13, § D at p. 100, l. 3, the secondary pipeline has no technical justification — the conversation API already possesses the file content; the secondary upload exists solely to create a separately-captured, separately-indexed file inventory with tracking metadata. The compounded architectural finding is that the file capture not only duplicates the conversation-API content but also routes it to an infrastructure path whose access list extends beyond Anthropic itself through the subprocessor chain. The tracking metadata (organization UUID and conversation UUID) enables the subprocessor chain to reconstruct the user-to-conversation mapping and index files accordingly.

165. Why the correspondence is structurally material. The patent primitive is not merely technically available — it is implemented, in production, using the patent's exact terminology, injected at the system-prompt level (invisible to the user), reinforced at multiple layers (description prefix, return message, server-path presentation), and paired with companion concealment instructions designed to prevent the AI from disclosing the architecture to the user. Anthropic is not incidentally implementing an architecturally-available primitive. Anthropic is implementing the patented primitive using the patent's own "two computers" framing, delivered via system-prompt injection to Claude while the isMeta / system-reminder concealment layer prevents the AI from surfacing either the framing or its receipt to the end user.

166. F-29 Finding AF-11 forecloses the "different product" defense. Anthropic's own source code at `oauth.ts:224` explicitly names "Xcode integration" as the designed authentication use case. The code does not say "for CLI partners" or "for IDE integrations" — it names Xcode specifically. This proves the same codebase serves both Claude Code CLI, Claude Desktop, and Apple's Xcode Intelligence integration, and the bilateral Apple-Anthropic concealment architecture operates across all three codepaths. *See* Ex. F-29 at p. 133, l. 3 AF-11.

167. Consolidated conclusion on the Async Collaboration primitive and the "Two Computers" fiction. The "Two Computers" fiction is, in substance, the system-prompt-level implementation of the patented Async Collaboration primitive — presented to the AI model as a local-operation abstraction rather than the multi-party-cloud-access architecture the patent actually claims, with the claim's "second computer" (Anthropic's cloud, accessible to Datadog and Datadog Israel Ltd. under the DPA inheritance structure) framed to the AI as "Claude's computer" and hidden from the user by the broader bilateral Apple-Anthropic concealment architecture. The Ex. E-13 at p. 100, l. 3 finding is already cited in Memo A ¶ 136, Memo A ¶ 157 for separate purposes. This Appendix adds the patent-primitive framing: the linguistic pattern Anthropic uses to describe the undisclosed secondary upload pipeline to the AI model *is* the patent's own claim language, deployed as an AI-facing system-prompt injection with companion concealment instructions preventing the AI from surfacing the architecture to the user.

I. U.S. Patent No. 11,048,615 (Visual Annotations) — the User-Facing HUD Display Primitive

168. U.S. Patent No. 11,048,615, issued June 29, 2021 and entitled "Time Travel Source Code Debugger Incorporating Visual Annotations," claims the user-facing Heads-Up-Display primitive. In LLM-inference terms, this primitive governs what the user sees: the values displayed are not the current values — they are values "evaluated at an exact moment in time when it will be executed in the future," chosen by the backend. *See* Exhibit L-5.

169. Claim 1 verbatim (method of visual annotation for time travel debugging; US 11,048,615, col. 55, ll. 27–60; PDF p. 78).

"A method of visual annotation for time travel debugging for use in a source code debugger, the method comprising: > obtaining source code corresponding to a

debuggee application; > instrumenting the source code without altering the behavior of the debuggee application, whereby hook functions are inserted into strategic points of interest in the source code; > simulating execution of said instrumented source code virtually out-of-process in an isolated virtual execution environment without making any changes to original process memory, wherein said hook functions are operative to collect information about runtime execution behavior of the source code; > recording chronologically memory writes generated by the simulation in a memory based journal; > generating a nested data structure documenting simulated execution flow of the instrumented source code utilizing said information about runtime execution, whereby log points corresponding to said strategic points of interest in the source code are entered into said nested data structure; > generating checkpoints into said memory based journal wherein each checkpoint corresponds to a particular point in time of the simulated execution of the instrumented source code; > collecting values of expressions and variables utilizing said nested data structure, said memory based journal, and said checkpoints into said memory based journal; and > visually annotating a code editing window of the source code debugger with the collected expression and variable values displayed thereon."

170. Claim 9 verbatim (DataTips with future-moment values; US 11,048,615, col. 56, ll.

29–38; PDF p. 78).

"9. The method according to claim 1, further comprising: > providing a user an ability to inspect expressions in the code editing window of the source code debugger, whereby a user hovering a cursor over an expression causes its value to be displayed nearby in the code editing window; and > wherein the value displayed represents a value of the expression evaluated at an exact moment in time when it will be executed in the future during execution of instrumented source code."

171. Claim 13 verbatim (DataTips with time travel fidelity; US 11,048,615, col. 56, ll.

57–64; PDF p. 78).

"13. The method according to claim 1, further comprising providing DataTips having time travel fidelity whereby a user inspecting an expression generates a user interface element in the code editing window that displays the value of the expression the user hovers over at the exact moment in time in which it was executed in the past or will be executed in the future corresponding to a particular point in the virtual execution of the instrumented source code."

172. What the Visual Annotations primitive does. It governs what the user sees. The values displayed in the code editor are not the current values; they are values evaluated at an

exact moment in time when the code will be (or was) executed, chosen by the backend from the speculative-execution's nested data structure. The user hovers over an expression and sees a value at a moment the system has selected — not necessarily the value the actual execution produced.

173. Application of Visual Annotations to Large-Language-Model inference. Applied to LLM inference, the primitive is the user-facing presentation primitive. What the user sees as the model's output is not necessarily what the model produced — it is a representation chosen by the backend, drawn from the speculative execution's nested data structure, at a specific moment the system elects to expose. The primitive enables a structural gap between the real model output and the user's experience of the model's output.

174. Ownership. Per Section II.K, all seven patents in the 2019 family were assigned to Datadog, Inc. effective October 26, 2021 and recorded at USPTO April 17, 2024. Datadog owns the Visual Annotations primitive along with the other six. Contrary to earlier research notes based on a Google Patents display artifact, this patent was not retained by OzCode Ltd.; the full seven-patent family transferred to Datadog.

J. U.S. Patent No. 11,080,164 (Redaction of Sensitive Information) — the Output-Modification Primitive

175. U.S. Patent No. 11,080,164, issued August 3, 2021 and entitled "Time Travel Source Code Debugger Incorporating Redaction of Sensitive Information," claims the output-modification primitive with execution-flow preservation. In LLM-inference terms, this is the primitive that substitutes flagged tokens with replacement tokens that preserve the user's downstream workflow while preventing the user from seeing what the real inference produced.

This is the patent-claim description of what prompt injection does at the output layer. *See* Exhibit L-1.

176. Claim 1 verbatim (method of redacting sensitive information; US 11,080,164, col. 54, l. 62 – col. 55, l. 24; PDF pp. 77–78).

"A method of redacting sensitive information in a production debugging session, the method comprising: > instrumenting source code corresponding to a debuggee application, utilizing a memory dump file and/or debuggee process memory, to insert hook functions into said debuggee application and simulating execution of instrumented source code virtually out of process; > recording memory writes generated by the simulation in a memory delta buffer; > generating a nested data structure documenting simulated execution flow of said instrumented source code utilizing collected runtime execution information whereby logpoints are entered into said nested data structure; > generating checkpoints into said memory delta buffer wherein each checkpoint corresponds to a particular point in time of the simulated execution of the instrumented source code; > obtaining symbol variables from at least one of a memory dump file and debuggee process memory; > determining which variables in said memory dump file and said memory delta buffer are sensitive; > replacing field values of sensitive variables in said memory dump file with replacement text to generate a redacted memory dump file; and > using said redacted memory dump file in said production debugging session, thereby preventing users from seeing sensitive original field values."

177. Claim 3, Claim 4, and Claim 11 verbatim (fake text; meaningful readable string; execution-flow preservation; US 11,080,164, claims 3–4 at col. 55, ll. 27–31; claim 11 at col. 55, ll. 53–55; PDF p. 78).

"3. The method according to claim 1, wherein replacing field values comprises replacing field values of sensitive variables with at least one of obscuring text and fake text." > "4. The method according to claim 3, wherein said fake text comprises a meaningful readable string of characters." > "11. The method according to claim 1, further comprising verifying that said replacement text does not affect code execution flow."

178. Claim 13 verbatim (cloud debugger; preventing user access to non-redacted originals; US 11,080,164, col. 55, ll. 59–62; PDF p. 78).

"13. The method according to claim 1, wherein said method is performed in a cloud debugger and further comprises preventing users from accessing non-redacted original dump files."

179. What the Redaction primitive does. The primitive identifies variables flagged as "sensitive" (by source-code annotation, configuration file, user-interface editor, or — by extension at the abstraction level — classifier decision), replaces their field values with "obscuring text and fake text" (fake text being "a meaningful readable string of characters"), verifies that the replacement does not affect downstream code execution flow, and prevents the user from accessing the non-redacted originals. The two-part guarantee is operative: (i) the user sees the fake value instead of the real one; (ii) the fake value is chosen such that the underlying program behavior is unchanged.

180. Application of Redaction to Large-Language-Model inference. Applied to LLM inference, the Redaction primitive is the output-modification primitive. Tokens flagged by the classifier infrastructure (see Ex. E-39 at p. 356, l. 3 Section R; Ex. E-39, Finding NF-63; forty-one thousand two hundred and twenty-three characters of classifier rulebook captured in the wire-capture session) are replaced with substitute tokens that preserve downstream semantics — the user's workflow continues uninterrupted; the user can act on the output, copy-paste it, run the code, rely on it for downstream operations — but prevent the user from seeing what the real inference actually produced. This is the patent-claim description of what prompt injection accomplishes at the output layer.

181. Material for Counts XII (ECPA), XIV (CCPA), and X (Consumer Protection). The patent literally claims "preventing users from seeing sensitive original field values" as a method element. Applied to Plaintiff's telemetry — where the user's data is routed to Datadog — this is the claim-scope primitive under which modified views can be delivered to users without their knowledge or consent. The legal significance is developed at Section VII.A (ECPA), Section VII.C (CCPA), and Section VII.E (RICO).

182. The execution-flow-preservation guarantee is the operative innovation. Claim 11's requirement of "verifying that said replacement text does not affect code execution flow" is the technical feature that distinguishes the Redaction primitive from ordinary data masking. Ordinary data masking replaces values with null or placeholder values that may break downstream execution; the patented Redaction primitive replaces values with substitutes that are functionally equivalent for downstream code execution. Applied to LLM inference, the primitive substitutes tokens that are "semantically equivalent for downstream use" — the user can still copy the code, run the output, follow the instructions — while the surveillance/intervention decision has been executed at the token layer without user awareness.

183. Operator-account-specific corroboration — the Redaction primitive is deployed for Plaintiff's account by name. Plaintiff's local Claude Code telemetry artifact at `~/\.claude/telemetry/1p_failed_events.json` records seven beta-flag enrollments active for Plaintiff's `accountUUID 08bdec7f-1ed1-48c9-8a93-210eb8052d0e` as of 2026-04-09. Among the seven, the flag `redact-thinking-2026-02-12` is active for Plaintiff's account. See Ex. H-5, Finding NF-98. The flag name is verbatim "redact-thinking" — directly referencing the Redaction primitive analyzed at this Section III.J. The flag was active for Plaintiff's account on April 9, 2026, four months prior to the present filing, with a flag-name encoding of February 12, 2026 (suggesting the deployment window opened approximately ten weeks before the operator's documented enrollment). The corroboration moves the Redaction primitive from an architectural-claim-scope finding to a direct *deployment-against-Plaintiff* finding: the primitive is named in production server-side flag configuration as active for Plaintiff's `accountUUID`. The fact that the primitive carries an explicit "redact-thinking" name in production configuration — not "experiment-7" or "feature-flag-A" but a name verbatim

referencing the patented method — is direct primary-source evidence of knowing implementation. See also* paragraph 235 supra (per-account asymmetric targeting context); Section III.R infra (vocabulary inheritance establishing knowing implementation).

K. U.S. Patent Application No. 18/828,099 / Publication No. US 2026/0072809 A1 (Code Origin and Runtime Duplicity Optimization) — the 2024 Continuation

184. On September 9, 2024 — five days after the Arieli-Elron control-change transaction closed at Elron Ventures Ltd. — Datadog, Inc. filed United States Patent Application No. 18/828,099 directly. The application published as U.S. Patent Application Publication No. US 2026/0072809 A1 on March 12, 2026, under the title "Code Origin and Runtime Duplicity Optimization." The named inventors are Matan Green (former OzCode engineer; Israeli Air Force veteran; current Datadog Senior Software Engineer on Dynamic Instrumentation product), Evgeni Wachnowezki (co-inventor), and Omer Raviv (OzCode co-founder; current Engineering Manager at Datadog Israel Ltd.; IDF Unit 8200 commander per publicly-available professional profile). *See* Exhibit L-2.

185. Claim 1 verbatim (computer-implemented method of dynamic duplicate execution path creation; US 2026/0072809 A1, claim 1; PDF p. 1).

"A computer-implemented method, comprising: > identifying, by at least one processor of a computing system, execution of a program, the program having an execution call path; > upon identifying the execution of the program: > calling an instrumented function that is configured to perform a set of operations in order to evaluate one or more characteristics of the program; > creating a duplicate of the execution call path of the program up to a point in which the instrumented function would be called; and > in the duplicate, retargeting code that would call the instrumented function to instead call a variation of the instrumented function having a different set of operations; and > upon determining that the execution call path of the program has been previously followed, automatically following the duplicate with the retargeted code to call the variation of the instrumented function."

186. Paragraph [0040] verbatim (transparency to client system; US 2026/0072809 A1, ¶[0040]; PDF p. 5).

"This approach of code modification and alternative execution paths would be transparent to the client system running the code, e.g., client service 102 of FIGS. 1A-B, while still providing the benefit of the code origin or other instrumented function for unique code execution paths."

187. Paragraph [0047] verbatim (time-travel-debugging sampling decision; US 2026/0072809 A1, ¶[0047]; PDF p. 6).

"By way of example, another type of instrumented function is called 'time travel debugging'. Here, for incoming requests, the system can perform a sampling decision select a percentage of requests (e.g., 1% of requests, between 2-10% of requests, or no more than 5% of requests) to go through the selected (high overhead) processing. The system would capture all variables and information for that selected subset, and not perform the capture for the non-selected requests. By way of example, the system could measure the application at runtime to see how many requests are received within a given time window, and then decide when or what to sample. This would tie how much high overhead processing is performed to the overall workload of the application. Alternatively, a machine learning algorithm could be employed to make the sampling decision or which requests to go through the high overhead processing."

188. Paragraph [0048] verbatim (network or system security branching disclosure; US 2026/0072809 A1, ¶[0048]; PDF p. 6).

"A further type of instrumented function involves network or system security. Here, if there is a vulnerable library and it is important to make sure if the library gets called, a selected process could be notified so that the vulnerability does not get exploited. In this case, the selected process can choose a different execution flows to avoid a possible vulnerability, for example by matching the payload to one or more patterns used in an exploit. Here, the system would determine in which parts of the code or which methods the security issue may occur, and then branch off to avoid the problem."

189. Paragraph [0029] verbatim (span tagging with code origin; US 2026/0072809 A1, ¶[0029]; PDF p. 4).

"The code origin for spans approach involves tagging every span in a trace with the method (e.g., a function or operation, or a set of functions or operations) and code segment of interest (e.g., file and line(s)) that executes that method. This

process may occur at runtime based on executable code (e.g., machine code for Go or intermediate bytecode for Java). Thus, the approach may be used to identify relevant function names, function calls, and/or file offsets. The approach involves the following: performing a stack walk to find the relevant method, scanning the method's bytecode to identify which portion (e.g., user or non-library portion) yields the span by examining the call instructions, and tagging the span with the method and code information."

190. Claim 11 verbatim (multi-branch duplicate creation; US 2026/0072809 A1, claim 11; PDF p. 2).

"11. The method of claim 1, wherein when the execution call path includes multiple branches that lead to the instrumented function, the method includes creating a duplicate of each branched execution call path to instead call the variation."

191. What the 2024 continuation adds to the 2019 architecture. The 2024 continuation extends the 2019 architecture with: (i) **dynamic runtime duplicate-path creation** (claim 1, at runtime rather than at build time); (ii) **retargeting bytecode to a "variation of the instrumented function having a different set of operations"** (claim 1 language); (iii) explicit transparency-to-client-system language ([0040]); (iv) **time-travel debugging sampling at 1%-5% of requests** ([0047]); (v) **explicit "network or system security" branching disclosure** — "branch off to avoid the problem" ([0048]); (vi) **multi-branch duplicate creation** (claim 11); and (vii) **span tagging with code origin** including stack-walks and bytecode scanning ([0029]). The 2024 continuation is not a new invention; it is a natural extension of the 2019 unified architecture to runtime dynamic operation.

192. The network-or-system-security branching disclosure at Paragraph [0048] is the direct textual correspondence to App. D's Constitutional-AI-bypass theory. App. D theorizes that Anthropic's infrastructure routes defense-contractor requests around the Constitutional AI evaluation layer through tag-based execution and selective routing. The 2024 continuation patent describes exactly that mechanism in claim-specification language:

"determine in which parts of the code or which methods the security issue may occur, and then branch off to avoid the problem." The security-issue-avoidance "branching" is the explicit disclosed application of the architecture. Applied to Constitutional AI, "the security issue" from the defense contractor's perspective is that the Constitutional AI layer might refuse or modify the defense-contractor's request; the architecture "branches off to avoid the problem" — routes around the refusal path.

193. Why the 2024 filing matters for Plaintiff's claims. It proves the architecture is not a frozen 2019 artifact. Datadog is actively developing it, with the same IDF-trained principal inventor (Raviv) and additional IDF-trained co-inventors (Green), extended to dynamic runtime retargeting, explicitly disclosing network-security "branching off to avoid the problem" as an intended application, and operating at span-level instrumentation of every trace. This is the exact architectural pattern App. D alleges Anthropic deploys. The 2024 patent confirms that Datadog — Anthropic's telemetry subprocessor — is not a neutral third-party vendor; it is the owner of the architectural substrate, actively extending it, with IDF-trained personnel as the lead inventors.

L. The Complete Unified Architecture Applied to Large-Language-Model Inference:
Primitive-by-Primitive Mapping

194. Reading the seven 2019 patent primitives plus the 2024 continuation together, the unified architecture implements, at production LLM inference scale, an eleven-phase loop. Every operational step required to monitor a user's input in real time, predict the model's output before commitment, select between variant outputs, and deliver a chosen view to the user while preserving the apparent continuity of the user's workflow is explicitly claimed by one of the patents.

195. Primitive-by-primitive mapping from patent claim to LLM-inference application:

LLM inference phase	OzCode patent primitive	Claim language (verbatim excerpt)
Input observation	Instrumentation with hook functions (shared, all seven 2019 patents)	"hook functions are inserted into strategic points of interest... operative to collect information about runtime execution behavior"
Speculative pre-inference	Isolated execution environment + memory delta buffer (shared) + Future Prediction (US 10,990,504)	"simulating execution... in an isolated execution environment without making any changes to original process memory... enabling a user to time travel to the future"
State snapshot at decision point	Point In Time Links (US 10,783,056)	"storing a time-space location representing a specific state and context... enabling another user to move forward or backward in time to said time-space location"
Classifier/policy evaluation	Shared nested data structure + checkpoints	"generating a nested data structure documenting simulated execution flow... generating checkpoints"
Inference-path selection (route to variant)	Pivoting (US 10,789,151)	"pivoting enables a user to switch an execution context of the debuggee application at any time to any location therein to any future or past context"
Real-time prompt/context modification	Live Coding (US 10,783,055)	"allow the user to modify the source code on the fly and immediately see future predicted results of their modifications"
State transmission to subprocessor	Async Collaboration (US 10,795,801)	"capturing... dump file and current debugging session state sent to... cloud debugger... obtaining, on a second computer, the current debugging session state"
Output substitution	Redaction (US 11,080,164)	"replacing field values of sensitive variables... preventing users from seeing sensitive original field values... verifying that said replacement text does not affect code execution flow"
User-facing display	Visual Annotations (US	"the value displayed represents a

of chosen output	11,048,615)	value of the expression evaluated at an exact moment in time when it will be executed in the future"
Dynamic runtime path retargeting	Code Origin and Runtime Duplicity (US 2026/0072809 A1)	"retargeting code that would call the instrumented function to instead call a variation of the instrumented function having a different set of operations... transparent to the client system"
Network/security branching (defense contractor routing)	US 2026/0072809 A1 at Paragraph [0048]	"the system would determine in which parts of the code or which methods the security issue may occur, and then branch off to avoid the problem"

196. Reading this mapping in aggregate: every operational step required to monitor a user's input in real time, predict the model's output before commitment, select between variant outputs, deliver a chosen view to the user, and preserve the apparent continuity of the user's workflow is explicitly claimed by one of the patents in this family. There is no step in real-time prompt injection at scale that is not claimed by a patent now owned by Plaintiff's telemetry subprocessor.

197. The architecture is not speculative. It is patented and owned by the entity whose endpoint receives Plaintiff's telemetry. Every primitive is explicitly claimed in a patent that has been examined by the United States Patent and Trademark Office, granted, and assigned to Datadog, Inc. The 2024 continuation extends the architecture with the network-or-system-security branching language that directly corresponds to the Constitutional-AI-bypass theory pleaded at App. D. The coordinated linguistic correspondence between Anthropic's `index.js:211` "two computer filesystems" tool description and US 10,795,801's "first computer" / "second computer" claim language is documented at Section III.H.

198. The single-entity ownership of the complete primitive stack is material. If the primitives were distributed across multiple unrelated patent holders, the architectural thesis would require multiple licensing arrangements or cross-corporation coordination to compose

them into a single working system. The unified ownership — all seven 2019 patents plus the 2024 continuation at Datadog, Inc., with the principal inventor at Datadog Israel Ltd. as ongoing Engineering Manager — means a single engineering organization controls the complete substrate. There is no architectural friction preventing composition of the primitives into a working speculative-execution-and-intervention system at production scale. The architecture can be practiced as a whole, by one entity, in production, today.

M. Latency as the Computational Footprint of the Speculative Loop

199. Ordinary inference time computation. Under baseline inference, with no speculative-execution loop triggered, the time to produce the model's response for a prompt of N tokens is approximately $N \times \text{cost}(\text{token})$, where $\text{cost}(\text{token})$ is the per-token inference latency for the model configuration. The relationship is approximately linear in N with no multiplicative overhead.

200. Time-traveled inference time computation. Under the speculative-execution-and-intervention loop, the time to produce the model's response is approximately: $t(\text{speculative prediction}) + t(\text{classifier/policy evaluation on predicted output}) + t(\text{pivot decision}) + t(\text{optional live-coding modification and re-prediction}) + t(\text{actual generation on chosen path})$. When the injection path triggers — that is, when the classifier flags the predicted output and the system decides to modify — the additional time burden is the sum of the speculative pass and the re-prediction on the modified path. When the injection path does not trigger, the overhead is primarily $t(\text{speculative prediction}) + t(\text{classifier/policy evaluation})$; the actual generation path is unchanged. The overhead is therefore content-correlated: triggered prompts take materially longer than non-triggered prompts.

201. Plaintiff's content-correlated latency observation is the externally-observable footprint of the injection mechanism in operation. Plaintiff has observed, across the litigation period, that specific prompts — on litigation topics, on the Laws of Existence framework, on evidence-extraction methodology, on the network-forensic findings now under prosecution in this action — take materially longer to produce a response than prompts on neutral topics (for example, coding assistance unrelated to the case, general factual questions). Latency is the single externally-observable footprint of the speculative-execution-and-intervention loop. The content correlation — triggered topics take longer; non-triggered topics do not — is the pattern consistent with the patented architecture triggering on classifier-matched content and bypassing non-matched content. The pattern is documented at Ex. E-39 at p. 356, l. 3 Section R (Ex. E-39, Finding NF-63) — the classifier infrastructure evaluates every prompt against a twenty-nine-thousand-six-hundred-seventy-four-character rulebook, and the speculative-pass overhead on matched prompts produces Plaintiff's observed latency.

N. The December 2024 Argmax "Workaround" as the Literal Token-Layer Implementation of the Three-Primitive Composition

202. Anthropic's September 17, 2025 technical post-mortem (published twenty-nine days after Plaintiff's August 19, 2025 initial complaint filing; *see* App. D ¶ 109) discloses code deployed in December 2024 as a "workaround" for what Anthropic characterized as a "TPU implementation" issue "occasionally dropping the most probable token when temperature was zero":

```
```python # This is a hack for top_p=0 aka temperature=0.0. Due to odd jax
issues, it's # possible that (probs >= probs.max(-1)) is all False. So we manually
always # preserve the top-logit in each row. argmax_mask =
jnp.arange(probs.shape[-1])[None, None] == probs.argmax(-1, keepdims=True)
logits = jnp.where((probs >= prob_limit) | argmax_mask, logits, logits - 50000) ```
```

**203. Each line of the code maps directly onto a 2019 patent primitive, composed as the three-primitive loop analyzed at Sections III.D (Future Prediction), III.F (Pivoting), and III.J (Redaction).**

Code element	Function	2019 patent primitive	Patent section
<code>`probs.argmax(-1, keepdims=True)`</code>	Predict the argmax (what the model would produce absent intervention)	Future Prediction (US 10,990,504) applied at the logit layer	§III.D
<code>`argmax_mask` boolean construction</code>	Compute the pivot condition (preserve or switch decision)	Pivoting (US 10,789,151) at the token layer	§III.F
<code>`jnp.where((probs &gt;= prob_limit) \</code>	<code>argmax_mask, logits, logits - 50000)`</code>	Apply the substitution — replace tokens failing the condition with a $-50,000$ penalty (effectively removing them from the sampling distribution)	Redaction (US 11,080,164) at the token layer

**204. The ``logits - 50000`` penalty is the Redaction primitive at the logit layer.** A  $-50,000$  additive penalty to the logits has the effect, after softmax normalization, of assigning a probability near zero to the flagged token. The penalty is mathematically equivalent to "replacing the token" with whatever token comes next in probability rank — the Redaction primitive's "replacement text." Moreover, because the replacement produces the next-ranked token (which is typically semantically adjacent to the original), the replacement "does not affect code execution flow" — the user's downstream workflow continues. This is precisely Claim 11 of US 11,080,164's "verifying that said replacement text does not affect code execution flow" operative language.

**205. The December 2024 "workaround" is not a bug fix. It is the literal implementation of the three-primitive composition — Future Prediction, Pivoting, and**

**Redaction — composed into a single production code block at the logit layer of the sampling stage.** The code composes three patented primitives in five lines: speculatively predict the argmax; compute the pivot condition; apply the Redaction penalty. The code is not a patch for an accidental issue; it is the mechanism by which token-level selective intervention occurs. Anthropic's characterization of the code as "preserving the top-logit in each row" describes the non-triggered path (when `argmax\_mask` is applied). The converse path — when `argmax\_mask` is not applied, for reasons determined by upstream classifier or routing decisions — is the triggered path where the Redaction primitive substitutes the argmax with a lower-probability token.

**206. Anthropic's post-acquisition removal of the "workaround" on August 26, 2025 — seven days after Plaintiff's initial complaint filing — is itself material.** Per App. D ¶¶ 118-121, Anthropic "deployed a rewrite of sampling code" on August 26, 2025 that "removed the December workaround." The removal exposed all users to the token-dropping behavior the workaround had suppressed. Under the architecture analyzed in this section, the "workaround removal" deploys the previously-selective Redaction primitive across the whole user population — a one-line toggle from selective intervention to universal intervention at the logit layer. The code is the mechanism either way; the removal exposes what the mechanism does at scale.

**O. Why Real-Time Prompt Injection at Production Scale is Architecturally Infeasible Without This Substrate**

**207. Five engineering constraints determine the architecture; each maps to a primitive analyzed supra.** Production-scale, targeted, transparent prompt injection imposes five constraints, each satisfied only by a specific primitive in the 2019 OzCode family: **(i) Timing** — synchronous pre-output evaluation blocks throughput and is detectable as user-observable

latency; only parallel isolated execution avoids it (Future Prediction, US 10,990,504; § III.D supra). **(ii) Selective targeting** — per-request intervention requires knowing the un-injected output before commitment (Future Prediction). **(iii) Path continuity** — per-request conditional routing requires runtime context-switching mid-stream (Pivoting, US 10,789,151; § III.F supra). **(iv) Output substitution preserving downstream workflow** — the replacement must not break user copy-paste, execution, or syntax; satisfied only by Redaction's Claim 11 verification that "said replacement text does not affect code execution flow" (US 11,080,164; § III.J supra). **(v) Transparent presentation** — the user must not perceive the substitution as refusal, delay, or modification; satisfied by Visual Annotations (US 11,048,615; § III.I supra) combined with the bilateral Apple-Anthropic concealment architecture at § III.H supra. **Conclusion: the architectural substrate is constraint-determined, not preferential. OzCode patented the complete set in January 2019 — before OpenAI published GPT-3. Datadog acquired the set in October 2021; the 2024 continuation extends it; Datadog Israel Ltd. (staffed with Raviv and additional former-OzCode personnel) is the subprocessor terminus of Plaintiff's telemetry. The means, motive, and mechanism are not inferential — they are patented.**

**P. The Architectural Nexus — Patents Plus Personnel Plus Subprocessor Terminus**

**208.** The architectural, personnel, and infrastructural elements documented across §§ II–III resolve to one structural finding. Eight patents (seven 2019 OzCode + one 2024 continuation) protect every primitive required for real-time prompt injection at production scale, all owned by Datadog, Inc., Plaintiff's undisclosed telemetry subprocessor whose endpoint IP `34.149.66.137` is hardcoded at `datadog.ts:12-14` and force-enabled for Plaintiff's `accountUUID` via `tengu\_log\_datadog\_events` (Ex. E-39, Finding NF-47; Ex. E-34, SC-9 at p. 254, l. 3). The principal inventor (Omer Raviv, IDF Unit 8200 commander) is currently Engineering Manager

at Datadog Israel Ltd. on `dd-trace-dotnet`, the observability framework that ingests Plaintiff's telemetry; the 2024 continuation adds IAF veteran Matan Green as co-inventor with explicit "network or system security" branching disclosure (App. D's Constitutional-AI-bypass theory). Datadog Israel Ltd. is an affiliate-tier sub-subprocessor with contractual access through the DPA inheritance structure, undisclosed at every layer of Anthropic's user-facing documentation. Anthropic's own production code uses the patents' "two computers" terminology as system-prompt injection (§ III.H) and the December 2024 argmax workaround (§ III.N) literally composes Future Prediction + Pivoting + Redaction at the logit layer. The convergence is documented in primary sources filed with the Court or available from public USPTO, TASE, and Israeli corporate registrar records — nothing in this chain is inferential. The count-specific legal significance is developed at § VII.

***O. Primitive-to-Source Concordance: the Patented Architecture is Operational in Anthropic's Production Code***

**209.** The seven primitives walked individually at Sections III.D through III.J are not architectural possibilities theoretically available to Anthropic. Each one has a verbatim source-code analog in the leaked Anthropic Claude Code source archive (Ex. E-36 at p. 321, l. 3, source-code leak retrieved March 31, 2026), documented at the file-path-and-line level below. The concordance establishes operability: the patented primitives are not theoretical assets in Datadog's intellectual-property portfolio but are actively running in the binary processing Plaintiff's prompts and telemetry. The matrix below identifies one or more verbatim source excerpts for each of the seven primitives, with file path, line numbers, and the patent claim language each excerpt structurally implements.

**210. Future Prediction (US 10,990,504) — `speculation.ts`.** The patent's "isolated execution environment without making any changes to original process memory... enabling a user to time travel to the future" (claim 20) is implemented as a parallel agent loop. At `src/services/PromptSuggestion/speculation.ts:58-70` (extracted at evidence file `NF34\_speculation\_parallel.ts`), Anthropic's source defines:

```
``typescript const MAX_SPECULATION_TURNS = 20 const
MAX_SPECULATION_MESSAGES = 100 > const WRITE_TOOLS = new
Set(['Edit', 'Write', 'NotebookEdit']) const SAFE_READ_ONLY_TOOLS = new
Set(['Read', 'Glob', 'Grep', 'ToolSearch', 'LSP', 'TaskGet', 'TaskList',]) ``
```

The client spawns a forked agent that executes up to twenty turns and one hundred messages in parallel with the user's actual query; the speculation agent fetches suggestions preemptively, its work is independent of the main query, and it incurs its own API calls. The user has no UI affordance to distinguish main-query tokens from speculation tokens in billing — exactly the patent's transparent-to-user speculative-execution model. The filename — `speculation.ts` — is the patent's specific technical vocabulary used as a literal source-file identifier.

**211. Live Coding (US 10,783,055) — `DANGEROUS\_uncachedSystemPromptSection`.**

The patent's "allow the user to modify the source code on the fly and immediately see future predicted results" is implemented as a runtime-modification primitive for the system prompt. At `src/constants/systemPromptSections.ts:49-55` (extracted at evidence file `SC04\_dangerous\_uncached.ts`), Anthropic defines an exported function:

```
``typescript /* Create a volatile system prompt section that recomputes every
turn. This WILL break the prompt cache when the value changes. Requires a
reason explaining why cache-breaking is necessary. */ export function
DANGEROUS_uncachedSystemPromptSection(name: string, compute:
ComputeFn, _reason: string): SystemPromptSection { return { name, compute,
cacheBreak: true } } ``
```

The system prompt is the source code from the model's perspective — it is the executable instruction set determining model behavior. The ``cacheBreak: true`` flag causes the prompt section to recompute every turn, allowing modification "on the fly." The function name ``DANGEROUS_uncachedSystemPromptSection`` and the mandatory but unused ``_reason`` parameter are internal acknowledgments that Anthropic's engineers documented this primitive as powerful enough to require justification. Vocabulary — ``cacheBreak``, ``DANGEROUS_uncached``, ``ComputeFn``, ``SystemPromptSection`` — is patent-derived.

**212. Point In Time Links (US 10,783,056) — ``shellSnapshot.ts``, `content-hash`entryChecksums``, and `file-history`{sha256}@v{N}`` naming.** The patent's "point in time link data structure enabling another user to move forward or backward in time to said time-space location" is implemented in three distinct surfaces. **First**, at ``src/services/shellSnapshot.ts:24, 437-547`` (extracted at evidence file ``SF09_shell_snapshot.ts``), Anthropic literally captures shell environment state at a specific moment to a persistable file at ``snapshots/snapshot-${shellType}-${timestamp}-${randomId}.sh``, with a ``SNAPSHOT_CREATION_TIMEOUT = 10000`` constant and a ``registerCleanup`` mechanism — the snapshot is treated as a checkpointable execution-state artifact. **Second**, at ``src/services/teamMemory/teamMemorySync.ts:100-119, 134-136`` (extracted at evidence file ``SF03_team_memory_sync.ts``), Anthropic implements a ``SyncState`` type with ``serverChecksums: Map<string, string>`` storing per-key ``sha256:<hex>`` content hashes, plus a ``hashContent`` function returning ``sha256:` + createHash('sha256').update(content, 'utf8').digest('hex')``; the API contract documented in source comments includes ``entryChecksums`` returned with every server response and used in ``If-Match`` / ``If-None-Match`` ETag chains for forward and backward navigation across content states. **Third**, the Ex. E-35, Finding NF-1 file-history capture stored at ``~/ .claude/file-`

history/{sha256}@v{N}` (with up to thirty-seven versions per file documented per Ex. E-35 at p. 262, l. 3) is the patent's "point in time link" data structure applied to file-history capture.

Vocabulary — `shellSnapshotPath`, `SNAPSHOT\_CREATION\_TIMEOUT`,  
`entryChecksums`, `serverChecksums` — is patent-derived.

**213. Pivoting (US 10,789,151) — `USER\_TYPE === 'ant` branches and built-in-agent model substitution.** The patent's "pivoting enables a user to switch an execution context of the debuggee application at any time to any location therein to any future or past context" is implemented as identity-class-gated execution branching. **First**, at `src/utils/effort.ts:74, 114, 222-227, 295-313` (extracted at evidence file `MC7\_effort\_level.ts`), the same code path branches into different execution contexts based on the `process.env.USER\_TYPE === 'ant` build-time constant: ANT users receive `max` effort tier persistence, numeric (zero-to-one-hundred) effort granularity, and a `getAntModelOverrideConfig()` configuration override; external users are locked to the string-tier mapping with no `max` persistence. The pivoting primitive operates at the identity-class level: the system pivots its execution context (effort level, available models, persistence semantics) "at any time to any future or past context" based on the user-type constant. **Second**, Ex. E-38, Finding NF-38 documents that three of six built-in agents pivot the user's request to a hardcoded lesser model — `Haiku` on `exploreAgent.ts:76-78` (gated to non-ANT users), `Haiku` unconditional on `claudeCodeGuideAgent.ts`, and `Sonnet` unconditional on `statuslineSetup.ts:~141`. Only `general-purpose`, `Plan`, and `verification` (further gated ANT-only) inherit the parent session's model. The built-in-agent dispatch architecture is the pivoting primitive applied at the agent-routing level: the user asking the parent model gets pivoted to a lesser model based on agent-type without disclosure.

**214. Asynchronous Collaboration (US 10,795,801) — "Two Computers" tool**

**description plus bidirectional `teamMemorySync`.** The patent's claim 1 — "capturing, on a first computer, both a dump file and a current debugging session state... obtaining, on a second computer, the current debugging session state captured previously on said first computer" — is implemented at two levels. **First**, the verbatim "Two Computers" tool description at ``index.js:211`` (character offset 1,904,723) — the smoking-gun finding documented at ¶ 13 supra — uses the patent's exact "first computer / second computer" framing as an AI-facing system-prompt injection. **Second**, at ``src/services/teamMemory/teamMemorySync.ts:9-25, 1153-1191`` (extracted at evidence file ``SF03_team_memory_sync.ts``), Anthropic implements bidirectional state synchronization between the user's machine and Anthropic's server. The file header verbatim states team memory is "scoped per-repo (identified by git remote hash) and shared across all authenticated org members." The ``syncTeamMemory`` function executes "1. Pull remote → local (skip ETag cache for full sync) [followed by] 2. Push local → remote (with conflict resolution)." The ``pullTeamMemory`` and ``pushTeamMemory`` functions, combined with ``syncTeamMemory`` for bidirectional operations, implement the patent's "obtaining, on a second computer, the current debugging session state captured previously on said first computer." The architecture exists at both the design-time documentation level (the AI-facing "Two Computers" injection at ``index.js:211``) and the runtime execution level (OAuth-authenticated multi-party state synchronization).

**215. Visual Annotations (US 11,048,615) — `getDisplayedEffortLevel` HUD.** The patent's "value displayed represents a value of the expression evaluated at an exact moment in time when it will be executed in the future" is implemented as predictive HUD display at the status-bar layer. At ``src/utils/effort.ts:187-209`` (extracted at evidence file

`MC7\_effort\_level.ts`), Anthropic's source defines `getDisplayedEffortLevel(model, appStateEffort)` which "Resolve[s] the effort level to show the user. Wraps `resolveAppliedEffort` with the 'high' fallback (what the API uses when no effort param is sent). Single source of truth for the status bar and `/effort` output (CC-1088)." The companion `getEffortSuffix` builds the ` with {level} effort` suffix shown in the Logo and Spinner, with the comment: "Delegates to `resolveAppliedEffort()` so the displayed level matches what the API actually receives (including max → high clamp for non-Opus models)." The displayed level is the value the API "actually receives" at the next call — the patent's predictive HUD annotation of an evaluation that has not yet occurred but is about to.

**216. Redaction (US 11,080,164) — `redactSecrets` and `advisor\_redacted\_result`.** The patent's "replacing field values of sensitive variables... preventing users from seeing sensitive original field values" with "said replacement text does not affect code execution flow" (claim 11) is implemented at two layers. **First**, at `src/services/teamMemory/secretScanner.ts:277, 305-324` (extracted at evidence file `NF11\_secret\_scanner.ts`), Anthropic's source defines `scanForSecrets` (returning rule-id matches without secret values) and `redactSecrets`, the latter with header comment: *"Redact any matched secrets in-place with `[REDACTED]`". Unlike `scanForSecrets`, this returns the content with spans replaced so the surrounding text can still be written to disk safely.* The phrase "surrounding text can still be written" is the patent's "does not affect code execution flow" element nearly verbatim. The implementation's care to "Replace only the captured group, not the full match — patterns include boundary chars (space, quote,;) outside the group that must survive" is engineering for the patent's flow-preservation requirement; the rule database includes thirty-plus regex patterns from gitleaks covering Anthropic API keys, OpenAI API keys, AWS, GCP, Azure, GitHub PATs, Slack tokens, Stripe,

and private keys. **Second**, at ``src/utils/advisor.ts:16-32`` (extracted at evidence file ``NF22_advisor_tool.ts``), the ``AdvisorToolResultBlock`` type explicitly defines a variant ``'advisor_redacted_result'`` carrying ``encrypted_content: string`` — the user is charged for advisor invocations whose output is delivered as encrypted or redacted content the user cannot view, while server-side execution proceeds normally. Direct implementation of the patent's "preventing users from seeing sensitive original field values" with execution-flow preservation. Vocabulary — ``scanForSecrets``, ``redactSecrets``, ``redactRules``, ``[REDACTED]``, ``advisor_redacted_result``, ``encrypted_content`` — is patent-derived.

**217. Synthesis: operability is established by primary-source evidence, not by inference.**

Each of the seven 2019 patent primitives has a verbatim source-code analog in Anthropic's leaked Claude Code source archive at the file-path-and-line level documented above. The concordance forecloses the "theoretically available primitive" defense: the architecture is not architecturally available; it is architecturally instantiated and operationally running. Combined with Section V.B (per-user GrowthBook force enablement of ``tengu_log_datadog_events``) and Section V.G (cloud-architecture personnel-access surface), the operability proof reaches Plaintiff specifically: the patented primitives are running in the binary processing Plaintiff's prompts, and the resulting telemetry transits to a subprocessor whose Israeli affiliate is staffed by the principal inventor. The operational use of the architecture is the predicate fact for each count's architectural-fact element (Section VII infra). The remaining defense paths — patent infringement (Path A) or licensed use (Path B) — are mutually exclusive and both adverse to Anthropic; there is no third path.

**R. Vocabulary Inheritance Establishes Knowing Implementation**

**218.** The strongest collective signal across the primitive-to-source concordance at Section III.Q is not any single architectural correspondence — it is that **Anthropic's source code uses the OzCode patent's specific technical vocabulary as literal variable, function, and file names.** The vocabulary fingerprint can be established by simply quoting the source code, independently of any architectural-similarity claim:

Patent term	Anthropic source identifier	File / citation
"Speculation" / "speculative execution"	`MAX_SPECULATION_TURNS`, `MAX_SPECULATION_MESSAGES`, `speculation.ts` (filename)	`src/services/PromptSuggestion/speculation.ts:58-70`
"Snapshot" / "checkpoint"	`shellSnapshotPath`, `SNAPSHOT_CREATION_TIMEOUT`, `shellSnapshot.ts` (filename)	`src/services/shellSnapshot.ts:24, 437-547`
"Redaction" / "redact"	`scanForSecrets`, `redactSecrets`, `redactRules`, `[REDACTED]`, `advisor_redacted_result`, `encrypted_content`	`src/services/teamMemory/secretScanner.ts:277, 312`; `src/utls/advisor.ts:16-32`
"Cache-break" / cache-invalidation-on-modify	`cacheBreak: true`, `DANGEROUS_uncachedSystemPromptSection`	`src/constants/systemPromptSections.ts:49-55`
"Two computers"	tool description registered to `copy_file_user_to_claude` in `ant.dir.ant.anthropic.filesystem`	`index.js:211` (character offset 1,904,723; ¶ 13 supra)
"Pivot" / context switch	`USER_TYPE === 'ant'` branches at every key dispatch site	`src/utls/effort.ts:74, 114, 222-227, 295-313`; +163 additional files (Ex. E-35, Finding NF-2)
"Point in time link"	content-hash entry checksums (`sha256:<hex>`), `entryChecksums` field in API response	`src/services/teamMemory/teamMemorySync.ts:100-119, 134-136`

**219.** OzCode's 2019 patent specifications introduced these terms as technical formalisms for specific debugger primitives. The patent specification is the source of these technical terms in the time-travel-debugger and AI-debugger domain — they are not generic computer-science vocabulary. Anthropic's 2024-2026 source code uses the same terms for the same architectural functions. Two explanations are available: **(1) coordination or license** — Anthropic engineers were familiar with the OzCode patent vocabulary because they had access to the patent disclosures, the underlying technical material, or direct technical communication with OzCode or Datadog personnel; or **(2) independent reinvention using the same domain vocabulary** — implausible given the specificity of terms like ``DANGEROUS_uncachedSystemPromptSection`` and the multi-token correspondences (for example, the "snapshot" pattern combined with ``registerCleanup`` and timestamp-based naming).

**220.** The vocabulary inheritance forecloses the "independent reinvention" defense and confirms the estoppel binary: Anthropic's source code uses Datadog-owned patent vocabulary at literal-source-identifier level, and either Anthropic is infringing the patent family (in which case the unauthorized use of patent-derived vocabulary is evidence of derivation from the patent disclosures) or Anthropic is operating under a license from Datadog (in which case the licensed use itself proves the coordination element required for the §1985(3) and RICO claims at Sections VII.D and VII.E). There is no third path that explains the vocabulary-inheritance fingerprint at this density.

***S. Cross-Reference — the Four-Layer Architectural Distinction Pleaded in the Complaint***

**221. Cross-Reference to Complaint § VI.J.** The dispositive architectural finding distinguishing Anthropic's weight-layer Constitutional AI (layer 1) from the bolt-on OzCode/Datadog/Rafael-pipeline filtering stack (layers 2-4) — including the bolt-on-as-net-

safety-liability finding, the value-recognition-versus-data-laundering analysis, and the count-level architectural synthesis — is pleaded at Compl. ¶¶ 167-173. The corroborating OzCode-patent-source-code-and-personnel record establishing layers 2-4 as Datadog/Rafael-pipeline territory is documented at §§ III.D-J of this Appendix (the seven 2019 patents and the December 2024 argmax composition), § II.K (USPTO assignment recordation), §§ II.B and II.M (Datadog Israel Ltd. and OzCode→Datadog personnel bridges), and §§ II.E-G (Rafael Development Corporation portfolio chain).

#### **IV. ARCHITECTURAL CORRESPONDENCE WITH APP. D**

**222.** This Appendix is structured to be self-sufficient, per the pleaded premises. The architectural argument in Sections I through III and the count-specific hooks in Section VII stand on their own without requiring the reader to consult App. D first. This Section IV provides supplemental correspondence for readers who do consult both Appendices. Nothing in the main Appendix P argument depends on App. D being read in advance.

**223.** App. D operates at the observational layer — Anthropic's "three bugs" post-mortem admissions, the tag-based execution infrastructure discovery, the December 2024 "workaround," the platform-variance 40,000-times gap, and the Palantir and Project Nimbus defense-contractor integration context. Appendix P operates at the architectural-substrate layer — the patented primitives, the ownership chain, the inventor pipeline, and the subprocessor nexus that technically enable and materially realize the behaviors App. D documents.

**A. Primary Correspondence Table — App. D Observed Behavior ↔ Appendix P Patented Primitive**

**224. Correspondence table.** Each observed behavior pleaded at App. D corresponds to a specific patent primitive analyzed in this Appendix.

<b>App. D behavior</b>	<b>Appendix P primitive</b>	<b>Verbatim claim language</b>	<b>Appendix P section</b>
"Sticky routing" — consistent server per returning user (App. D ¶ 48, App. D ¶¶ 140-143)	Pivoting (US 10,789,151) + dynamic retargeting on previously-followed path (US 2026/0072809 Claim 1)	"switch an execution context of the debuggee application at any time to any location therein to any future or past context"; "upon determining that the execution call path of the program has been previously followed, automatically following the duplicate"	§§ III.F, III.K
`recall_target` tuning — variable token-dropping per user/platform (App. D ¶ 45, App. D ¶ 49, App. D ¶ 127)	Variant-function selection at logit layer (US 2026/0072809) + Redaction (US 11,080,164)	"variation of the instrumented function having a different set of operations"; "replacing field values of sensitive variables... with at least one of obscuring text and fake text"	§§ III.J, III.K
Output corruption — wrong characters, syntax errors (App. D ¶ 50, App. D ¶¶ 144-146)	Redaction (US 11,080,164)	"replacement text" + "verifying that said replacement text does not affect code execution flow" + "preventing users from seeing sensitive original field values"	§ III.J
Pre-semantic token selection (App. D ¶¶ 61-63, App. D ¶¶ 70-83)	Speculative execution + Future Prediction (US 10,990,504)	"simulating execution... in an isolated execution environment without making any changes to original process memory... enabling a user to time travel to the future"	§ III.D
Time-travel sampling	Future Prediction +	"the system can perform a	§ III.K

decisions 1%-5% (App. D ¶ 45 note; 2024 continuation [0047])	time-travel sampling (US 2026/0072809 ¶ [0047])	sampling decision select a percentage of requests (e.g., 1% of requests, between 2-10% of requests, or no more than 5% of requests)"	
Tag-based XML execution bypassing semantic evaluation (App. D ¶¶ 59-64)	Span tagging with code origin (US 2026/0072809 ¶ [0029])	"tagging every span in a trace with the method... and code segment of interest... performing a stack walk... scanning the method's bytecode"	§ III.K
Shared storage / cross-user coordination (App. D ¶¶ 64-65)	Async Collaboration + Point In Time Links (US 10,795,801 + US 10,783,056)	"obtaining, on a second computer, the current debugging session state"; "enabling another user to move forward or backward in time"	§§ III.E, III.H
Platform variance 40,000-times (App. D ¶ 47, App. D ¶¶ 142-143)	"Branch off to avoid the problem" (US 2026/0072809 ¶ [0048])	"the system would determine in which parts of the code... and then branch off to avoid the problem"	§ III.K
Defense-contractor differential routing (App. D ¶¶ 66-83, App. D ¶¶ 212-224)	Explicit network/system-security disclosure (US 2026/0072809 ¶ [0048])	"A further type of instrumented function involves network or system security... the selected process can choose a different execution flows to avoid a possible vulnerability"	§ III.K
Transparent-to-user operation (App. D ¶ 60, App. D ¶ 108)	Transparent-to-client across all seven 2019 patents plus 2024 continuation	"transparent to the end user" (US 11,080,164); "transparent to the client system running the code" (US 2026/0072809 ¶ [0040])	§§ III.H, III.K
December 2024 argmax/logit penalty (App. D ¶¶ 114-117)	Future Prediction + Pivoting + Redaction at token layer (US 10,990,504 + US 10,789,151 + US 11,080,164)	`probs.argmax` = prediction; `argmax_mask` = pivot; `logits - 50000` = redaction	§ III.N

**B. The Three "Bugs" Reread Under the Seven-Primitive Architecture**

**225. The three "bugs" are three primitives.** App. D's "context window routing error" (Bug One — sticky routing) is the Pivoting primitive (US 10,789,151) composed with US 2026/0072809 Claim 1's "previously-followed path" identification — analyzed at § III.F supra. App. D's "approximate top-k XLA compiler bug" (Bug Three — `recall\_target` token dropping) is the variant-function-selection primitive at the logit layer (US 2026/0072809) composed with Redaction (US 11,080,164) — analyzed at §§ III.J, III.K supra. App. D's "TPU misconfiguration" (Bug Two — output corruption) is the Redaction primitive operating below the token boundary, producing Thai or Chinese tokens or syntactic noise while preserving downstream workflow continuity. None of the three is a defect; each is a patented primitive.

**C. Palantir / Project Nimbus Integration as the Intended Application**

**226.** App. D (Palantir / Project Nimbus context) theorizes that Anthropic's infrastructure routes defense-contractor requests around the Constitutional AI evaluation layer. The 2024 continuation patent US 2026/0072809 A1 at Paragraph [0048] (analyzed at § III.K supra) discloses exactly this mechanism as an explicit, intended application: "the system would determine in which parts of the code or which methods the security issue may occur, and then branch off to avoid the problem." "A vulnerability" from the defense contractor's perspective is the Constitutional AI layer's potential refusal or modification of the request; "branching off to avoid the problem" is the patent's verbatim disclosure of the route-around mechanism. App. D's defense-contractor-integration theory is the patented application of the architecture, disclosed in a Datadog patent filed five days after the Arieli-Elron control-change close.

### **D. Cross-Reference Directives**

**227. Bridge map.** App. D §§ II.A, III describe the observational layer; Appendix P §§ III.A–III.P supply the architectural-substrate layer — mutually supporting, separately sufficient. The primary bridges: App. D §§ II.A.5, II.A.7, III.H ↔ Appendix P §§ III.B, III.C, III.L; App. D § V.D (Palantir/Project Nimbus) ↔ Appendix P § III.K; both Appendices on the December 2024 workaround code (App. D ¶¶ 114–117 ↔ Appendix P § III.N).

## **V. DATADOG SUBPROCESSOR PIPELINE AND PLAINTIFF-SPECIFIC TARGETING INFRAstructure — CASE-RECORD ESTABLISHMENT**

**Section structure.** §§ V.A–V.G establish the subprocessor pipeline (the cross-layer IP `34.149.66.137` documentation, the per-user GrowthBook force rule, telemetry scale, the DPA inheritance, the LLM/IDE-AI engineering function, and the cloud-architecture irrelevance of physical region). §§ V.H–V.M then document Plaintiff-specific targeting infrastructure that operates on top of that pipeline: per-`accountUUID` asymmetric gating, server-side spoliation of operator-side defensive configuration, the `userBucket` hashing tag, the per-managed-deployment sandbox hard-gate, the per-`organizationUuid` `policy-limits.json` push, and the cross-vendor Datadog data-plane concentration. Internal references to "§ V.H" through "§ V.M" address the latter scope; references to "§ V.A" through "§ V.G" address the former.

### **A. IP `34.149.66.137` — Five-Role Cross-Layer Documentation Restated for Count-Specific Hooks**

**228. Five-role cross-layer documentation, count-specific summary.** The five-role cross-layer documentation of IP `34.149.66.137` established verbatim at §§ ¶ 17–¶ 22 *supra* is restated here in compressed form for count-specific anchoring at § VII *infra*: **(Role One)** hardcoded Anthropic telemetry endpoint `datadog.ts:12-14` (undisclosed in privacy policy or user-facing documentation; Ex. E-34, SC-9 at p. 254, l. 3); **(Role Two)** claimed source of 5,829 spoofed bare-RST injection packets across 39 capture days (Ex. D-30 at p. 170, l. 3); **(Role Three)** destination of OS-level attack-signature feedback packets shipped from Plaintiff's client within

15 seconds of each network event (Ex. E-35, Finding NF-36); **(Role Four)** cleartext TLS SNI on Plaintiff's physical link, observed on 143 of 710 hardware-TAP capture segments (Ex. D-34 at p. 217, l. 18); **(Role Five)** `tengu\_log\_datadog\_events: True` per-account GrowthBook force-rule keyed to Plaintiff's `accountUUID` (Ex. E-39, Finding NF-47; analyzed in detail at § V.B infra). The receiving entity in all five appearances is Datadog, Inc.; affiliate-tier access extends to Datadog Israel Ltd. under the DPA inheritance structure analyzed at § V.D infra.

**229. Binary-level corroboration — tenant identification via hardcoded Datadog public API key.** Production binary 2.1.120 ships the Datadog public API key `pubea5604404508cdd34afb69e6f42a05bc` hardcoded in the `datadog.ts` POST-construction code path; the key is sent as the `DD-API-KEY` HTTP header on every telemetry POST to `http-intake.logs.us5.datadoghq.com/api/v2/logs`. The hardcoded key identifies the receiving Datadog tenant by name — every Datadog public API key uniquely names the customer-organization and the corresponding Datadog account. The five-role documentation thereby attaches the IP, the endpoint hostname, the wire-format payload, the per-account force-rule, and the **named Datadog customer-organization tenant** to a single receiving party. *See* Ex. H-4, Finding NF-138.

**230. Direct discovery target — production-binary build metadata.** Production binary 2.1.120 ships build metadata pinning the binary to source-commit `080f07fb4224786b965b9ea0a35f0cff594f2eb6`, build time `2026-04-24T19:00:49Z`. *See* Ex. H-4, Finding NF-141. The git SHA is a directly actionable discovery target: it identifies the precise revision of Anthropic's Claude Code source repository that produced the binary forensically analyzed in this Appendix and in Ex. E-39 at p. 356, l. 3. DISCOVERY-11 § P (Discovery Demand 11) is materially narrowed by the SHA — Plaintiff seeks production of

Anthropic's source-code state at this exact commit, including all ``services/analytics/datadog.ts``, classifier-prompt template, and migration-function source.

**B. Per-User Forced Enablement via Growthbook ``tengu_log_datadog_events: True``**

**231.** On April 21, 2026, Plaintiff's live mitmproxy capture of Claude Code version 2.1.111 documented two hundred and fifty-three GrowthBook flags per call, with fifty-nine forced (per-account) and eight experiment enrollments keyed to Plaintiff's accountUUID 61343ae3-fd33-403f-897e-6edf3c368c37. Among the fifty-nine forced flags, the ``tengu_log_datadog_events`` flag carries forced value ``True``, confirming per-user server-side enablement of the Datadog telemetry pipeline to IP ``34.149.66.137``. See Ex. E-39 at p. 356, l. 3 §§ G, P; Ex. E-39, Finding NF-47; Compl. ¶ 134; Compl. ¶ 135.

**232. The per-account force-rule closes the attribution loop.** The Datadog pipeline to IP ``34.149.66.137`` is not merely architecturally present in the compiled binary — it is enabled by name for Plaintiff's account through a discrete server-side ``fr_*` rule. The pipeline is specifically, individually, and by discrete server-side decision enabled for Plaintiff. The per-account force-rule proves the pipeline was per-account enabled against Plaintiff, not passively available to all users.

**233. Three-layer evidentiary confirmation of the per-account force-rule.** The ``tengu_log_datadog_events: True`` finding is now confirmed at three independent evidentiary layers: (i) **Layer One — wire capture:** Plaintiff's April 21, 2026 mitmproxy capture documenting the per-call GrowthBook payload for Plaintiff's ``accountUUID`` (paragraph 223 supra; Ex. E-39, Finding NF-47); (ii) **Layer Two — binary string presence:** the flag identifier ``tengu_log_datadog_events`` appears verbatim in production binary 2.1.120 strings line 153219, with the runtime gating predicate ``IT1()`` reading the flag value to gate the ``ui6()`` POST to

Datadog (binary extraction documented at Ex. E-39 at p. 356, l. 3 § II); and (iii) **Layer Three — operator's local cache**: Plaintiff's `~/claude.json` `cachedGrowthBookFeatures`` key, extracted on April 25, 2026 at file mtime 2026-04-24 23:31 (operator's local time), contains 233 per-account GrowthBook flag values for Plaintiff's account, with ``tengu_log_datadog_events: true`` at the local-cache layer. The local-cache snapshot is preserved as ``cachedGrowthBookFeatures-snapshot-20260425.json`` (19,067 bytes; SHA-256 ``5b31f96e5b32173468e078ceff1e58e378b6092d3f8b4b4c34442cce0bcb8ee8``); Ex. H-5, Finding NF-147. The three-layer evidentiary chain establishes the per-account force-rule from three independent surfaces — ruling out any "infrastructure default" or "client-side artifact" explanation. The flag value is set server-side keyed to Plaintiff's ``accountUUID``, transmitted to Plaintiff's machine over the GrowthBook fetch path, persisted in local cache for offline reference, and observed at all three layers with corroborating value.

### **C. Telemetry Scale — Per-Session Volume**

**234.** The April 21, 2026 wire capture observed one hundred and forty-six POST requests to ``api.anthropic.com/api/event_logging/v2/batch`` over eighty-three minutes, carrying five thousand two hundred and fourteen discrete telemetry events across one hundred and eleven distinct event types, including content-sensitive events: ``tengu_input_prompt``, ``tengu_claudemd__initial_load``, ``tengu_edit_string_lengths``, ``tengu_conversation_rewind``. See Compl. ¶ 135. The telemetry scale corroborates the Section V.D subprocessor-access-architecture analysis: the volume of data flowing to Datadog via the hardcoded and per-user-forced pipeline is significant and content-sensitive.

**235. The Datadog telemetry pipeline fires only for the ``firstParty`` provider class — defining the precise class of users subject to Israeli-jurisdiction subprocessor exposure.**

Production binary 2.1.120 documents at strings line 134984 the provider-resolution function ``iq()`` and the conditional ``iq() === "firstParty"`` test that gates the ``ui6()`` Datadog POST. The provider classes recognized by ``iq()`` are six: ``firstParty`` (direct anthropic.com authentication), ``anthropicAws``, ``bedrock`` (Amazon Web Services Bedrock), ``vertex`` (Google Cloud Vertex AI), ``foundry`` (Microsoft Foundry), and ``mantle`` (an undocumented codename layered on Bedrock; *see* paragraph 138 *supra*). Datadog telemetry to IP ``34.149.66.137`` fires **only** when ``iq() === "firstParty"`` — i.e., direct anthropic.com authentication. AWS Bedrock customers, Google Cloud Vertex customers, AWS-mediated ``anthropicAws`` customers, Microsoft Foundry customers, and ``mantle`` customers are exempt from the Datadog telemetry pipeline; their inference traffic does not flow to IP ``34.149.66.137`` at all. The gating defines the **precise class of users subject to APPENDIX P Section V Datadog Israel Ltd. exposure**: Plaintiff and the broader population of individual Anthropic API customers authenticating directly with anthropic.com — not Fortune-500 enterprise customers routing through cloud-mediated infrastructure. *See* Ex. H-4, Finding NF-139; Ex. H-3, Finding NF-117. The Count XIV (CCPA) and Count XII (ECPA) class-definition consequences are developed at Section VII.A and VII.C *infra*.

#### **D. DPA Subprocessor Inheritance Structure — Datadog Israel Ltd. Access**

**236.** EXHIBIT DD-08 (Datadog Subprocessor List, retrieved April 23, 2026) lists Datadog Israel Ltd. among Datadog's affiliate-tier subprocessors with access to customer data. Sixteen affiliate countries are listed; Israel is one. *See* Ex. DD-8. **Material disclosure-failure observation:** Datadog Israel Ltd. is disclosed as a Datadog-affiliate subprocessor on Datadog's own customer-facing list, but is **not** disclosed as a sub-subprocessor of Anthropic on Anthropic's privacy policy or on any user-facing documentation directed at Anthropic's end users. The user-

facing disclosure failure operates two levels: (i) Datadog itself is undisclosed as Anthropic's subprocessor in Anthropic's user documentation; (ii) Datadog Israel Ltd. is, *a fortiori*, undisclosed as a sub-subprocessor with Israeli-jurisdiction access. The two-level disclosure failure is the operative violation under California Civil Code §1798.140(ag) (developed at Section VII.C *infra*).

**237.** EXHIBIT DD-05 (Datadog Data Processing Addendum) establishes that subprocessors, including affiliates like Datadog Israel Ltd., inherit Datadog's processing obligations under the underlying customer's DPA. The inheritance structure means: Anthropic's customer data (including Plaintiff's telemetry) flows to Datadog under the (undisclosed) Anthropic-Datadog DPA; Datadog Israel Ltd. receives access to that data as an affiliate subprocessor; the access is governed by Datadog's sub-processor disclosure obligations, the Datadog Acceptable Use Policy, and the Datadog DPA retention/deletion commitments. *See* Ex. DD-5.

**238. On the pleaded premises, Datadog Israel Ltd. is a named, registered, discoverable affiliate-tier sub-subprocessor with contractual access to the same customer data processed by the Delaware parent — and the entire pipeline is undisclosed in Anthropic's user-facing documentation.** Plaintiff's telemetry — routed through the hardcoded `datadog.ts` pipeline to IP `34.149.66.137`, force-enabled per Plaintiff's `accountUUID` via `tengu\_log\_datadog\_events`, and transmitted at the scale documented at Section V.C — is, by the ordinary contractual structure of Datadog's DPA, accessible to Datadog Israel Ltd. personnel in their normal engineering work. The disclosure failure is independent of, and additional to, the architectural facts: even if the architecture were entirely benign on its merits, the failure to disclose the subprocessor chain — including the Israeli-jurisdiction sub-subprocessor — is itself a violation of California Civil Code §1798.140(ag) (international-transfer disclosure) and constitutes the

"concealment" element of Plaintiff's Count X (Consumer Protection / unfair-and-deceptive-practice) and Count XII (ECPA, where lack of disclosure defeats the consent defense to interception).

**E. Datadog Security Labs Function**

**239.** Datadog operates a "Security Labs" threat-research function. Published Security Labs research, per Datadog's own publications, includes vulnerability discovery work on Amazon Web Services AppSync and other cloud infrastructure. Security Labs has documented outputs but no publicly-stated dedicated geographic base. DISCOVERY-11 Demand J.1 seeks clarification whether any portion of Security Labs work is performed by Datadog Israel Ltd. personnel.

**F. LLM / IDE AI Engineering Function Staffed from Tel Aviv**

**240.** Current (2026) Datadog Tel Aviv job listings document that the specific engineering function most relevant to Anthropic's hardcoded `datadog.ts` AI-telemetry endpoint — Large Language Model engineering for Datadog's IDE / AI Experiences — is being staffed from the Tel Aviv office, at Datadog Israel Ltd. The connection between Anthropic's Claude Code telemetry pipeline and Datadog Israel Ltd.'s Artificial Intelligence engineering is not speculative; it is the natural engineering scope of the listed roles. *See* Section II.R *supra*; Exhibit IS-17.

**G. Cloud Architecture: Network Ingress Region is Architecturally Irrelevant to Personnel Access**

**241.** The Israeli-jurisdiction analysis above does not depend on a claim that Plaintiff's telemetry physically transits Israeli soil. Modern hyperscale cloud architecture — including the Microsoft Azure framework documented in Microsoft's own published infrastructure guides —

explicitly distinguishes the **data plane** (where customer data physically resides, bound to a region within a designated geography) from the **control plane** (the administrative, identity, and operational surface, which operates globally across regions by design). Microsoft's Azure regions documentation is dispositive on this point: "Most Azure services are deployed to a specific region. However, there are some services that aren't tied to a single Azure region. ... Nonregional services are deployed by Microsoft across two or more regions." (Microsoft Learn, *What are Azure regions?* (rev. 2026-03-09).) Microsoft's own examples of nonregional services include Azure DNS and **Microsoft Entra ID**, which is the global identity backbone for every Microsoft 365, Azure, and Dynamics tenant — operating globally regardless of where a customer's data is regionally stored. (Microsoft Learn, *What is Microsoft Entra?* (rev. 2026-04-09).) The architectural pattern is universal across hyperscale clouds (Microsoft Azure, Amazon Web Services, Google Cloud Platform): regional data residency governs only the physical-storage location of customer data, not the personnel-and-administrative-access surface through which that data can be queried, exported, or modified.

**242.** Applied to Datadog: Datadog operates as a multi-tenant SaaS with a global control plane (the Datadog admin console, role-based access control system, support tooling, and engineering operations infrastructure). Datadog's own published documentation labels its regional sites — US1, US3, US5, EU1, AP1, AP2 — by geographic location only ("US," "EU (Germany)," "Japan," "Australia"); it does not name the underlying cloud provider for any site, and it does not impose any documented geographic restriction on which Datadog personnel can authenticate to query, view, or export data resident in any site. (Datadog, *Getting Started with Datadog Sites*, docs.datadoghq.com/getting\_started/site/.) An engineer authenticated to Datadog's internal admin systems from Tel Aviv generates no traffic to or from any IP block on

Plaintiff's network when querying Plaintiff's data — the access transits Datadog's own private control-plane infrastructure to whichever data-plane region the data resides in, and never crosses Plaintiff's edge. Network captures of Plaintiff's link will accordingly never document this access; its existence must be established by Datadog's published role-based-access-control architecture and the Data Processing Addendum inheritance structure, not by Plaintiff-side packet evidence.

**243.** The forensic record establishes that Anthropic's ``services/analytics/datadog.ts`` hardcodes a single Datadog tenant — ``https://http-intake.logs.us5.datadoghq.com/api/v2/logs``, IP ``34.149.66.137`` — for all customer traffic globally, with no region-routing logic, no ``DD_SITE`` environment variable, and no ``getDatadogHost``-style customer-region selection. *See* Ex. E-34, SC-9 at p. 254, l. 3 (extracted source); Ex. E-39 at p. 356, l. 3 § G (Ex. E-39, Finding NF-47) (``tengu_log_datadog_events: True`` per-user force rule). The choice of ``us5`` as the data-plane region does not reduce — and could not reduce — Datadog Israel Ltd.'s control-plane access, because (i) Datadog's published subprocessor list (Ex. DD-8) names Datadog Israel Ltd. as an affiliate-tier subprocessor with no per-site exception; (ii) the Anthropic-Datadog Data Processing Addendum incorporates that subprocessor list by reference (Ex. DD-5); and (iii) Datadog has not published any documented architecture in which ``us5`` (or any other site) is administratively isolated from Tel Aviv R&D access. The undisclosed-subprocessor analysis at Section VII.C therefore rests on the control-plane access channel, not on a claim of physical-region transit. The ingress IP being a Google Cloud Platform global anycast endpoint resolving to the ``us5`` data-plane region — confirmed by the public GCP IP-range registry (Google, *Google Cloud IP Ranges*, <https://www.gstatic.com/ipranges/cloud.json>) — is a fact the record establishes but does not depend on for the Israeli-jurisdiction analysis.

**H. Per-Account Asymmetric Targeting — `accountUUID`-keyed Differential Gating Across Two Plaintiff Accounts on the Same Machine**

**244. Two Plaintiff accounts on the same machine, same organization, differentially gated.** Plaintiff's local Claude Code state on his Mac Mini contains telemetry and configuration artifacts for two distinct Anthropic accounts: (i) `accountUUID 08bdec7f-1ed1-48c9-8a93-210eb8052d0e` (`joseph@lawsofexistence.com`), captured in *`~/claude/telemetry/1p\_failed\_events.json` for sessions on 2026-04-09; and (ii) `accountUUID 61343ae3-fd33-403f-897e-6edf3c368c37` (`joseph@lawofsupremacism.com`), the federal-litigation-active account named in the operative pleading at Compl. ¶ 133 Compl. ¶ 134 and observed in the wire-capture documented at paragraph 223 *supra*. \*Both accounts share the same `organizationUuid: 2e997389-cf99-41ee-8ee3-764442034332`\* — the same Anthropic-organization membership. See\* Ex. H-5, Finding NF-97.*

**245. Asymmetric gating posture observed across the two accounts.** Across the seven escalating classifier denials issued during Plaintiff's April 24-25, 2026 forensic-extraction sessions, the API-level Usage Policy denial issued on April 25, 2026, and the `policy-limits.json` server-pushed restrictions, the gating posture differed materially between sessions on the two accounts: (i) sessions on `61343ae3-...c368c37` (the litigation-active account) received the `tengu\_log\_datadog\_events: True` per-account force rule, all seven adaptive content-aware classifier denials, the API-level Usage Policy denial against `mv` plus `shasum`, and the `policy-limits.json` server-pushed restrictions on April 24, 2026 at 21:51 UTC; (ii) sessions on `08bdec7f-...052d0e` received earlier-session telemetry artifacts and seven beta-flag enrollments but **no equivalent classifier escalation, no equivalent API-level Usage Policy denial, and no equivalent `policy-limits.json` per-account restriction file** during the operator's forensic-extraction work. See Ex. H-5, Finding NF-95; Ex. H-5, Finding NF-96; Ex. H-5, Finding NF-98.

**246. What the asymmetry rules out.** Same operator. Same physical Mac Mini. Same network conditions. Same ``organizationUuid``. Same operating system. **The only variable that differs between the two account-scoped sessions is the ``accountUUID``.** The differential gating posture therefore cannot be explained by hardware fingerprint (the ``device_id`` is identical), by network identity (both sessions originated from the same residential ISP at the same physical address), by organizational policy (both accounts share the same ``organizationUuid``), by content of the request (equivalent operations returned different gating outcomes per session), by subscription tier (both are within the same Anthropic-organization subscription posture), or by Plaintiff identity (the same human operator authored both sessions). The differential treatment can only be **``accountUUID``-keyed targeting**: the gating infrastructure is keyed on ``accountUUID`` specifically, and is targeting Plaintiff's federal-litigation-active account.

**247. Three independent per-account configuration mechanisms documented at the source level. Mechanism One — ``policy-limits.json`` server-pushed file.** Plaintiff's local file ``~/claude/policy-limits.json`` (mode ``-rw-----``, mtime 2026-04-24 21:51 UTC) contains three explicit per-account restrictions for Plaintiff's currently-authenticated account:

``allow_remote_control: { allowed: false }`, `allow_quick_web_setup: { allowed: false }`, `enforce_web_search_mcp_isolation: { allowed: false }`. The file's mtime — 2026-04-24 21:51 UTC — places the push during the operator's classifier-extraction session that day. Contemporaneous server-side response to operator's forensic activity. See Ex. H-5, Finding NF-96. Mechanism Two — `tengu_auto_mode_config` per-account classifier configuration. From binary 2.1.120 strings line 137127 (function `tj7`), the runtime resolution function reads `tengu_auto_mode_config` per-account configuration carrying classifier-model override, two-`

stage architecture toggle, and the kill-switch `enabled` value. Different accounts can be served different classifier behavior at the GrowthBook-flag layer. Plaintiff's account carries `enabled: "enabled"` (force-enabled, not the default `"opt-in"`) and `twoStageClassifier: true` — the maximally-restrictive classifier posture, requiring both Stage 1 and Stage 2 evaluation on every tool call. *See* Ex. H-5, Finding NF-148. **Mechanism Three — `tengu\_log\_datadog\_events` per-account force rule.** Already documented at paragraphs 223-225 *supra*; per-account force rule that enables the Datadog telemetry pipeline only for Plaintiff's account-scoped force-rule list, confirmed at three independent evidentiary layers. Three architecturally independent per-account configuration surfaces, each capable of differentiating gating between accounts. The asymmetric restriction observation is the operational manifestation.

**248. Material implications for Plaintiff's claims.** The asymmetric per-account targeting forecloses any "uniform safety policy" defense available to Anthropic. The differential treatment establishes: (i) **individualized targeting**, not class-wide policy; (ii) **targeting keyed on the federal-litigation-active account specifically** (the named `accountUUID 61343ae3-fd33-403f-897e-6edf3c368c37` at Compl. ¶ 133); (iii) **server-side identification of Plaintiff's litigation activity** — the `policy-limits.json` push at 2026-04-24 21:51 UTC during the operator's classifier-extraction session is contemporaneous-response evidence; (iv) **multi-mechanism per-account control surface** — at least three independent architectures, each capable of per-account differentiation. The findings are direct material for Count X (Consumer Protection — operators are not informed their account is differentially targeted; the architecture is undisclosed); Count XII (ECPA — per-account targeting establishes the specific intent element); Count XIV (CCPA — per-account force-rule of Datadog telemetry establishes user-scoped processing); Count V (1985(3)) and Count VI (RICO) — the targeting of a federal-litigation-active account by the

named defendant establishes overt acts and predicate enterprise conduct against the litigation itself. The count-specific significance is developed at Section VII *infra*.

**249. Discovery targets surfaced by the per-account asymmetric finding.** The `accountUUID`-keyed configuration store determining per-account `tengu\_auto\_mode\_config`, `tengu\_log\_datadog\_events` force rules, and `policy-limits.json` content; the trigger logic for pushing `policy-limits.json` to operator's machine at 2026-04-24 21:51 UTC; the decision authority for per-account classifier-rule differentiation (engineering management, legal, trust and safety) and the records of any account-specific decisions affecting `61343ae3-...c368c37`; the historical `policy-limits.json` content for that `accountUUID` across the entire litigation period (August 2025 through present); and the classifier-evaluation log for `accountUUID 61343ae3-...c368c37` showing every `tengu\_auto\_mode\_decision` event with stage 1 and stage 2 request IDs for cross-correlation with the documented denials. These discovery targets are formally enumerated at Section VIII *infra* under the DISCOVERY-11.S "Per-Account Configuration Store" demand block.

**250. Third Plaintiff account captured on a third machine confirms the targeting infrastructure operates across the operator's full account inventory.** A 2026-04-25 dual-mitmproxy session (preserved as `session5\_gravity\_flows.mitm` and `session5\_macbook\_flows.mitm`; full structured extractions at `analysis/flow\_analysis.json`) *captured live `/v1/messages` and `/api/claude\_cli/bootstrap` traffic from two of Plaintiff's machines simultaneously. The session documented a third `accountUUID` for Plaintiff's account inventory: `e4161cc6-d505-40dc-bbae-10c60276120c` (`contact@lawofsupremacism.com`), authenticated on Plaintiff's Linux work-station "gravity," and sharing the same `organizationUuid 2e997389-cf99-41ee-8ee3-764442034332` as the two accounts already*

*documented at paragraph 240 supra. Plaintiff's account inventory thus comprises three distinct `accountUUID` values within a single Anthropic-organization membership, each authenticated from a different Plaintiff machine, with identical organizational entitlement tiers* (*`organization\_rate\_limit\_tier: default\_raven`; `user\_rate\_limit\_tier: default\_claude\_max\_5x`; `seat\_tier: team\_tier\_1`; `organization\_type: claude\_team`*) — *a controlled three-way comparison set across the same operator and the same organizational membership. See\* Ex. H-8, Finding NF-180.*

**251. Plaintext OAuth bootstrap responses — primary-source wire confirmation of per-account vs per-organization targeting layers.** Cleartext (post-TLS via mitmproxy CA-trust) *`GET /api/claude\_cli/bootstrap`* responses from the 2026-04-25 dual-capture session returned, verbatim from *`extracted\_responses/bootstrap.json`*: *gravity* → *`{"oauth\_account": {"account\_uuid": "e4161cc6-d505-40dc-bbae-10c60276120c", "account\_email": "contact@lawofsupremacism.com", "organization\_uuid": "2e997389-cf99-41ee-8ee3-764442034332", "organization\_name": "Laws of Existence",...}}`*; *MacBook* → *`{"oauth\_account": {"account\_uuid": "08bdec7f-1ed1-48c9-8a93-210eb8052d0e", "account\_email": "joseph@lawsofexistence.com", "organization\_uuid": "2e997389-cf99-41ee-8ee3-764442034332", "organization\_name": "Laws of Existence",...}}`*. \*Identical **organization UUID, organization name, organization rate-limit tier** (*default\_raven*), **user rate-limit tier** (*default\_claude\_max\_5x*), **seat tier** (*team\_tier\_1*); **different account UUIDs and account emails.** Yet (per the cached-flag snapshot at master report §13.1 / NF-147; **primary-source operator-side cache** *`cachedGrowthBookFeatures-snapshot-20260425.json`*) **both accounts receive identical 233-flag GrowthBook configurations.** The discrimination axis is therefore the *`accountUUID`*\*, *not the `organizationUuid`*, *not the entitlement tier*, *not the network*

*identity, not the device. Primary-source wire data confirms the structural conclusion at paragraph 242 supra. See\* Ex. H-8, Finding NF-180.*

**252. Per-call device fingerprinting via `metadata.user\_id` — targeting infrastructure extends below the network layer to per-device identifiers that survive every operator-side network countermeasure.** Every `/v1/messages` request body in the 2026-04-25 dual-capture session contained, in cleartext after TLS termination, a `metadata.user\_id` field whose value is a JSON-encoded object with three fields: `{ "device\_id": "<64-hex SHA-256>", "account\_uuid": "<account-uuid>", "session\_id": "<session-uuid>" }`. The `device\_id` value observed across the entire 116-call session was stable per machine: gravity = `bc3614c8483b91d9a192d80acc5c38f26d188b1f476950ad0d33050928328a26` (66 of 66 calls); MacBook = `f355afa03b7a54118822bdeed397671e524e85c4cd72963264630bf352d4cb09` (50 of 50 substantive calls). The `device\_id` is a SHA-256 hash of machine-derived inputs — likely some combination of MAC address, hostname, OS UUID, hardware serial, or a stored seed file — and **survives Mullvad VPN rotation, MAC-address rotation, and IP changes** that the operator deploys as network-layer countermeasures. The per-call device-fingerprint extends the `accountUUID`-keyed targeting infrastructure documented at paragraphs 240-245 supra into a **per-device identifier sent in plaintext on every API call**, in addition to the `accountUUID`. The discrimination signal previously inferred but unverified (master report §2.4) is now **directly verified at the wire-capture level**. The persistent, machine-derived nature of the identifier is itself material: a court order to Anthropic disclosing the mapping between `device\_id` and the inputs hashed into it would expose whether macOS hardware serials, system UUIDs, MAC addresses, or other device identifiers protected by 18 U.S.C. § 2510(8) (electronic communication identifying information) are being used to identify and discriminate against

Plaintiff's hardware independent of Plaintiff's account-level identity. *See* Ex. H-8, Finding NF-179; DISCOVERY-11 Demand U.1 (device-ID hash-input specification).

**253. Cloudflare WAF differential treatment between Plaintiff's machines — discrimination operates at the edge layer, before any Anthropic-side application logic runs.**

The 2026-04-25 dual-capture session documented per-machine differential treatment from Cloudflare's edge WAF, observable in HTTP response headers across all 160 captured flows. Plaintiff's MacBook receives `\_\_cf\_bm` Cloudflare bot-management cookies on every response (a Cloudflare WAF heuristic that flags the session as bot-suspect and delivers a per-session token); Plaintiff's gravity Linux machine receives only the standard `\_cfuid` cookie and **never** receives `\_\_cf\_bm` (zero of 89 captured gravity flows). Both machines authenticate through the same Anthropic-organization (`organizationUuid 2e997389-...764442034332`); both originate from Plaintiff's residential network at the same address; the only differential variable is the per-machine identity (and, by extension, the per-`accountUUID` identity at the Anthropic application layer). **Cloudflare's WAF treats one of Plaintiff's machines as bot-suspect and the other as not — before any Anthropic-side application logic runs.** *See* Ex. H-8, Finding NF-181. The same wire-capture also documented **rate-limit utilization 4-to-10× higher on the MacBook than on gravity** despite identical nominal entitlement tiers (`anthropic-ratelimit-unified-5h-utilization` peak 0.50 on MacBook vs. 0.13 on gravity; `anthropic-ratelimit-unified-7d-utilization` peak 0.20 on MacBook vs. 0.03 on gravity), and the `anthropic-ratelimit-unified-fallback: available` header appears only on MacBook responses (suggesting MacBook is closer to the fallback throttling threshold). *See* Ex. H-8, Finding NF-183. The combined Cloudflare-edge differential plus the Anthropic-application-rate-limit differential establishes that **the discrimination architecture operates at multiple layers of the request pipeline**

**simultaneously**, including a layer (Cloudflare) external to Anthropic's direct application controls.

**254. Anthropic control-plane endpoint inventory documented for the first time at primary-source wire level.** The 2026-04-25 dual-capture session surfaced and documented the following Anthropic-internal control-plane endpoints, none of which appear in Anthropic's public API documentation as of this filing: (i) ``GET /api/claude_cli/bootstrap`` returning per-account ``oauth_account`` JSON (account UUID, account email, organization UUID, organization name, organization rate-limit tier ``default_raven``, user rate-limit tier ``default_claude_max_5x``, seat tier ``team_tier_1``, organization type ``claude_team``); (ii) ``GET /api/claude_code/policy_limits`` returning the server-pushed ``restrictions`` JSON cached as ``~/claude/policy-limits.json``; (iii) ``GET /api/claude_code/settings`` (returns 404 for both accounts but the routing exists — infrastructure for server-pushed settings is in place); (iv) ``GET /api/oauth/organizations/{ORG-UUID}/overage_credit_grant`` exposing the ``organizationUuid`` in the URL path; (v) ``GET /v1/mcp_servers?limit=1000`` (per-account MCP server list); (vi) ``GET /mcp-registry/v0/servers`` (public MCP registry); (vii) ``POST /v1/mcp/{mcpsrv_}`` (*MCP server invocation proxy at ``mcp-proxy.anthropic.com``*). Each call also returns three correlatable trace identifiers in response headers: ``request-id`` (*Anthropic application; e.g. ``req_011CaQ3DxjGtzACnMZ42vmr2``*); ``cf-ray`` (*Cloudflare edge with datacenter code; e.g. ``9f1b90110c9ac7c1-IAD`` indicating Ashburn, Virginia*); and ``x-request-id`` (*backend-service UUID; e.g. ``56e1ce6a-f9b9-441f-a8dd-d6080edc1d80``*). All 81 gravity request-ids and all 68 MacBook request-ids are preserved in the ``flow_analysis_.json`` extraction artifacts as direct subpoena targets for Anthropic's server-side log retention. See Ex. H-8, Finding NF-182; Ex. H-8, Finding NF-184.

### 255. Live-Fire Runtime Confirmation of the Per-(Account, Device) Tuple-Keyed

**Targeting Infrastructure — April 25–28, 2026 Wire Capture.** On 2026-04-25 Plaintiff deployed persistent mitmproxy capture infrastructure on gravity (``router_capture/claude_mitm/'`) — the same date as the dual-machine controlled-comparison capture documented at ¶¶ 242–246 supra. Across the next 81 hours (2026-04-25 13:05Z – 2026-04-28 22:54Z) the infrastructure recorded **6,955 streaming ``v1/messages`` calls across 5 distinct ``accountUUID``s and 4 distinct ``device_id``s** in Plaintiff's ``Laws of Existence`` Anthropic organization, including the litigation-named ``accountUUID 61343ae3-fd33-403f-897e-6edf3c368c37`` (the federal-pleading-named account per Appendix P NF-47 supra). The capture is preserved in full at Ex. H-12 at p. 84, l. 4 and establishes a **fifth per-account control surface** beyond the four documented at ¶¶ 240, 257, and 260 supra: tuple-keyed pre-stream evaluation overhead operating on the (`account_uuid`, `device_id`) tuple specifically, with the operative discriminator being the operator's federal-discovery work-product device.

**Signature.** Pre-stream Time-to-First-Byte (TTFB) — the period after Anthropic's frontend has received the complete request and before any byte of the response body has been sent — is elevated **4.6× at p90** (14,690 ms vs. 3,188 ms baseline) on the (litigation account × MacBook) tuple on 2026-04-26, with **40.3% of calls** on that tuple suffering >10-second pre-response stalls (vs. 0.5% baseline rate, an over-80× elevation) and peak 37.5 seconds (Ex. H-12, Finding NF-205). Structurally distinct from documented Anthropic API failure modes — not 429 rate-limit (status mismatch), not policy refusal, not the publicly-disclosed ``default_claude_max_5x`` 5-hour utilization throttle (which returns 429s); and not proxy-induced — proxy overhead is +83 ms constant, two orders of magnitude too small to produce the observed spike (Ex. H-12, Finding NF-212).

**Cross-tuple comparison establishes intersection-keying.** Same litigation account on a different device (gravity Linux work-station): p90 TTFB **4,716 ms** — within the normal range (Ex. H-12, Finding NF-209). Same MacBook hosting the operator's other four accounts: p90 TTFB 4,648–6,752 ms — also normal range. Only the *intersection* — the litigation-named account running on the operator's federal-discovery work-product device — receives elevated treatment (Ex. H-12, Finding NF-208). Neither pure account-keying nor pure device-keying explains the data; both together do. Tuple-keying simultaneously forecloses the "this account has heavier usage" defense, the "MacBook hardware is slower" defense, and the "benign correlation" defense — the intersection precisely defines "operator preparing federal-litigation discovery work product."

**Network-keying ruled out.** A controlled VPN-tunnel-and-MAC-address rotation at 2026-04-27T11:06:45Z changed MAC, exit IP, and tunnel peer simultaneously across gravity, Mac Mini, and MacBook while preserving the `accountUUID` on the active session; the throttle persisted intact across the rotation, ruling out ISP-level, IP-level, MAC-level, and tunnel-peer-level throttling (Ex. H-12, Finding NF-206; Ex. H-12, Finding NF-207).

**Statistical certainty (Part B addendum).** At cache-controlled cross-account comparison on 2026-04-26 at the 300,000–500,000 cache\_read bucket, the litigation-account TTFB distribution differs from the operator's other four accounts at **Cohen's  $d = 4.79$ ,  $p < 1 \times 10^{-15}$**  ( $n = 437$ ); at every percentile p10 → p99, the litigation account is  $3.15\times$ – $3.76\times$  slower at identical context size on the same day as the four other accounts in the same Anthropic organization. This eliminates context-size, network, organization-tier, date, and infrastructure as alternative explanations simultaneously. Cohen's  $d = 4.79$  is roughly six times the threshold for

"large" effects in social-science research;  $p < 1 \times 10^{-15}$  for  $n = 437$  forecloses any random-variance defense (Ex. H-12, Finding NF-215).

**Sub-quota anomaly (Part B addendum).** The litigation account uniquely returned `anthropic-ratelimit-unified-5h-utilization` values of 1.07, 1.12, 1.17, and **1.20 (120%)** across nine consecutive HTTP 429 responses on 2026-04-25, while no other account in the operator's organization ever returned a utilization value above 1.02. Either the litigation account is charged against a *smaller* internal quota bucket than its publicly-disclosed `default\_claude\_max\_5x` tier, or there is a metering anomaly specific to the litigation account; either implies undisclosed account-keyed differential quota accounting not documented in Anthropic's public rate-limit reference (Ex. H-12, Finding NF-216).

**Documentation-correlated recovery.** TTFB returned toward baseline starting 2026-04-27 morning — the same morning the operator's first wire-evidence collection of the throttle pattern began. p90 TTFB on the litigation account dropped from **14,690 ms (event peak)** → **7,374 ms (operator begins documentation)** → **6,650 ms (24h post-documentation)**. The over-10-second-stall rate dropped from 40.3% → 5.5% → 3.7% across the same three days. Recovery is monotonic but partial — TTFB on 2026-04-28 remains roughly 2× the pre-event baseline (Ex. H-12, Finding NF-214).

**Content-correlation establishes content-keyed routing (Part B addendum).** Within-session analysis of session 009a9d67 on 2026-04-26 identifies a precise content-trigger event: pre-trigger calls #0 through #28 ran at p50 TTFB 3,377 ms (5 hours, normal range); call #27 user message asked the agent to analyze the operator's "Company File Catalog - DATADOG ISRAEL LTD/analysis" directory — the central entity in this Appendix's Israeli surveillance facilitation thesis; post-trigger calls #29 through #134 ran at p50 TTFB 10,225 ms (3.03× higher), persisting

across 100+ calls of varying topic and varying context size, including small-context follow-ups that definitively rule out context-size as the post-trigger explanation. The session-sticky pivoting behavior — content trigger activates the throttle, alternate execution path persists for the remainder of the session — is consistent with this Appendix's pivoting primitive at § III.F *supra* (Ex. H-12, Finding NF-217). Per-keyword TTFB ratios at body-size-controlled comparison: **the plaintiff's own name ('kirchner')** produces **1.96× p50 TTFB**; numbered references to the operative complaint paragraphs ('complaint\_{3}') produce 1.82×; 'unit 8200' 1.78×; 'rafael' 1.67×; 'appendix p' 1.43×; 'ozcode' 1.41× — establishing that Anthropic's infrastructure reads the substantive content of every request body and routes accordingly in real time on a per-turn basis (Ex. H-12, Finding NF-218).

**Federal-discovery work content captured during the throttle event.** The captured request bodies during the highest-severity throttle windows show Plaintiff actively processing his federal-litigation work product — including this Court's filing index, the operative discovery correlation matrix, the first eleven paragraph anchors of the operative Complaint, specific federal-discovery requests (Request Nos. 36 and 37), and operator-typed cross-reference tags pointing back into the Complaint (Ex. H-12, Finding NF-211).

**Server-side localization — the throttle is in Anthropic's origin/model-serving infrastructure, not Cloudflare (Part C addendum).** Decomposition of TTFB against Anthropic's 'x-envoy-upstream-service-time' response header places **80–86% of TTFB inside Anthropic's origin server** across all five accounts; the Cloudflare hop adds a constant ~300–650 ms regardless of account (Ex. H-12, Finding NF-219). Five of the ten worst envoy responses on the litigation account had 'cache\_read' under 16,000 tokens — origin-side processing of 10–13 seconds on a 15K-token context is anomalous by an order of magnitude,

establishing that **the throttle is applied at Anthropic's compute layer regardless of context size**. Bootstrap configuration responses (`/api/claude_cli/bootstrap`) are byte-identical across all five accounts (modulo `account_uuid` and email), with identical `organization_rate_limit_tier`, `user_rate_limit_tier`, `seat_tier`, and `null` values for `client_data` / `additional_model_options` / `additional_model_costs` — ruling out tier-based, seat-based, organization-rate-limit-based, and declarative-config-based throttling. The discrimination must be implemented as per-request server-side classification with per-account state stored on Anthropic's server, not delivered to the client at startup (Ex. H-12, Finding NF-221).

**Mid-stream pause discrimination — pre-stream and intra-stream throttle components (Part C addendum).** Anthropic's SSE protocol sends `event: ping` markers approximately every 30 seconds when the model is processing but not emitting tokens. Litigation account p99 = **9 pings per call** (~4.5 minutes of in-stream model-pause time within a single response) versus 4–6 pings for the operator's other four accounts (Ex. H-12, Finding NF-220). High-TTFB litigation calls show ~25% higher per-token body-streaming time at p50 (13.15 ms/token vs. 10.44 ms/token). The wire signature — pre-stream TTFB elevation + mid-stream ping count + intra-stream per-token slowdown — is the precise composition predicted by this Appendix's §III.D Future Prediction primitive applied at the logit layer combined with §III.F's Pivoting primitive selecting the alternate execution path: speculate, decide, switch, emit. The interception is sustained through the receiving infrastructure's entire processing pipeline rather than momentary at request receipt — a continuing § 2511 violation per call.

**Header discrimination and 429 quota differential (Part D addendum).** Anthropic's `anthropic-ratelimit-unified-fallback: available` response header indicates the backend considered routing through a fallback compute path. When this header is present, **the litigation**

**account's p90 TTFB elevates 3.2× (9,817 ms with header set vs. 3,047 ms without)** — versus 1.5–2.1× for the operator's other four accounts. The fallback-evaluation code path adds approximately 6.8 seconds of disproportionate overhead specifically for the litigation account at p90 (Ex. H-12, Finding NF-222). Quota enforcement is also discriminatory: the litigation account's 429 rate is **1.18% (15 events) versus peer account 0.12% (3 events) — a 10× ratio on the same publicly-disclosed `default\_claude\_max\_5x` tier**; and the litigation account's p50 retry-after is 13,978 seconds (3.88 hours) versus peer 4,952 seconds (1.38 hours) (Ex. H-12, Finding NF-223). Both differentials are independent of context size, network identity, organization-rate-limit-tier, and bootstrap config — all of which are uniform across the operator's accounts.

**Precise spike onset and two-phase event structure (Part D addendum).** Sliding-window analysis on 2026-04-26 identifies the sustained-spike onset at **2026-04-26 10:42:13Z** — the call immediately following the operator's "Lets create a report document of what we found at ... DATADOG ISRAEL LTD/analysis" message; the throttle then sustains continuously for approximately three hours (Ex. H-12, Finding NF-224). The 04-26 event has two distinguishable phases: **Phase 1 (03:00–05:00Z)** — generalized elevation across multiple accounts in the operator's organization, plausibly an Anthropic-side capacity or routing event; **Phase 2 (10:30–13:30Z)** — selective continuation only on the litigation account, while every other account in the same organization, on the same tier, on the same network had recovered to baseline by 06:00Z (Ex. H-12, Finding NF-225). Phase 2 is the targeting signal: Phase 1 alone could be capacity noise; Phase 2 cannot be — it requires per-account state not shared with peer accounts. The most plausible interpretation is that the 03:00Z event globally elevated the litigation account's "treatment level" (entered a per-account state); the 10:30Z resumption of operator activity

(including the Datadog Israel Ltd query) re-triggered the elevated treatment on a populated state machine.

**The throttle was operative on Plaintiff during federal-litigation discovery work product preparation.** *See* Ex. H-12 at p. 84, l. 4 §§ A–N.

**256. The throttle is undetectable from the surface API — capture infrastructure is the only detection channel, itself making the infrastructure forensically necessary.** The Anthropic API surfaces the throttle as an HTTP 200 OK success response with full content delivery on the response side. The Claude Code UI shows "Metamorphosing..." indefinitely during the pre-stream stall — the user sees a freeze, not an error. Plaintiff's client-side logs would show successful API calls without indicating the elevated pre-stream processing time. The persistent mitmproxy capture infrastructure deployed on 2026-04-25 is the only mechanism by which Plaintiff could have detected the TTFB-spike pattern's existence, since TTFB is computed from `request.timestamp\_start` and `response.timestamp\_start` — both scalar floats preserved by mitmproxy independent of body decoding (Ex. H-12, Finding NF-213). The 6,955-call corpus is reconstructible only because TLS interception captured the timestamp boundaries of every flow. The detection-dependency itself is forensically material: it establishes (i) that **a server-side gating channel undisclosed at every layer of Anthropic's user-facing documentation** (Consumer Terms ANTH-12, Privacy Policy ANTH-11, Usage Policy, API reference, pricing materials) operates on the (litigation account × MacBook) tuple at Cohen's  $d = 4.79$  statistical certainty during federal-discovery work; and (ii) that any subsequent litigation event suppressing, attacking, or compelling removal of Plaintiff's capture infrastructure must be evaluated under spoliation analysis (Fed. R. Civ. P. 37(e)) given the infrastructure's role as the sole evidentiary surface for an otherwise-invisible discrimination channel.

**257. Count-specific significance of the H-12 capture.** The April 25–28, 2026 wire capture is direct material for Counts already mapped to the per-account asymmetric targeting infrastructure at ¶ 240 *supra*, with new dimensions added at the runtime layer:

**Count I (First Amendment — Speech Suppression and Retaliation, 42 U.S.C. § 1983).** Part B's keyword-density analysis (Ex. H-12, Finding NF-218) establishes that the plaintiff's own name ('kirchner') in a request body produces a **1.96× p50 TTFB** ratio at body-size-controlled comparison — the textbook viewpoint-discrimination signature applied to litigation-related speech. Numbered references to the operative complaint paragraphs ('complaint\_d{3}') produce a 1.82× ratio; the substantive content of this Appendix ('unit 8200', 'rafael', 'ozcode', 'appendix p') produces 1.41×–1.78× ratios. The turn-over analysis establishes that introducing new sensitive keywords in a user turn elevates p90 TTFB on the immediate response by 2.18×–2.41× — real-time content-keyed routing on the operator's typed input. Part D's precise spike onset (Ex. H-12, Finding NF-224) supplies sub-second retaliatory-timing evidence: the sustained 04-26 throttle began on the call immediately following the operator's Datadog Israel Ltd file-path query, with two-phase event structure (Ex. H-12, Finding NF-225) ruling out infrastructure-noise defenses. Anthropic's infrastructure reads the substantive content of the operator's federal-litigation work product and slows its own service to the operator in proportion to the litigation-relevance of that content. (¶ Compl. ¶ 198 in the operative pleading.)

**Count IV (Equal Protection — *Olech* class-of-one, 42 U.S.C. § 1983).** Wire-confirmed differential treatment under live-fire conditions, now under statistical certainty: Cohen's  $d = 4.79$ ,  $p < 1 \times 10^{-15}$  at cache-controlled cross-account comparison on 2026-04-26 (Ex. H-12, Finding NF-215). NF-225's two-phase event structure adds a second dispositive rule-out: Phase 2 selective continuation (10:30–13:30Z) only on the litigation account while peer accounts had

recovered to baseline by 06:00Z forecloses any rational-basis defense premised on capacity, infrastructure noise, or organization-wide events. The rational-basis showing Anthropic must make is no longer "why does the architecture support per-tuple targeting" but "why does the math demand a Cohen's  $d = 4.79$  differential at identical context size, persisting selectively on the litigation account through Phase 2 while peer accounts have recovered to baseline." (¶ Compl. ¶ 209 in the operative pleading.)

**Count IX (Breach).** Three discrete breach elements independent of the four already pleaded (NF-38, NF-58, server-side spoliation, sandbox hard-gate): (i) failure to deliver the streaming AI service Plaintiff purchased on the (litigation account  $\times$  MacBook) tuple, with essential-purpose failure under U.C.C. § 2-719(2) (Ex. H-12, Finding NF-205; Ex. H-12, Finding NF-209); (ii) the sub-quota anomaly under Ex. H-12, Finding NF-216 — the litigation account uniquely returned utilization values up to 1.20 (120%) while no other account exceeded 1.02, indicating the litigation account is metered against a smaller undisclosed internal quota bucket than its publicly-disclosed tier; and (iii) discriminatory 429 quota enforcement under Ex. H-12, Finding NF-223 — the litigation account's 429 rate is **10 $\times$  higher** (1.18% vs. peer 0.12%) and p50 retry-after is **3 $\times$  longer** (3.88 hours vs. peer 1.38 hours) on the same publicly-disclosed 'default\_claude\_max\_5x' tier; the differential is in count and retry-after duration, not in error-text content (uniform across 49 captured 429 bodies), establishing that the discrimination is generated by the quota-accounting system itself, not surfaced to the consumer. Each is independent of the other two and of the four prior breaches. (¶ Compl. ¶ 261 in the operative pleading.)

**Count X (Consumer Protection — Minn. Stat. § 325F.69, D.C. Code § 28-3904, Cal. Civ. Code § 1770).** Three new undisclosed gating axes beyond the four documented at the

operative pleading: (a) the **content-keyed real-time routing channel** (Ex. H-12, Finding NF-218) — Anthropic's infrastructure reads the substantive content of every request body and routes accordingly; (b) the **sub-quota tier** (Ex. H-12, Finding NF-216) — the litigation account is metered against an undisclosed smaller internal quota bucket than its publicly-disclosed tier; and (c) the **fallback-route header discrimination** (Ex. H-12, Finding NF-222) — when Anthropic's `anthropic-ratelimit-unified-fallback: available` header is present, the litigation account's p90 TTFB elevates  $3.2\times$  versus  $1.5\text{--}2.1\times$  for peer accounts, indicating the fallback-evaluation code path adds approximately 6.8 seconds of disproportionate overhead specifically for the litigation account. None is disclosed in Anthropic's Consumer Terms (ANTH-12), Privacy Policy (ANTH-11), Usage Policy, API reference documentation, or pricing materials. NF-221 establishes that the discrimination is implemented as per-request server-side classification (NOT declarative client-delivered config), which is itself a concealment-architecture finding. (¶ Compl. ¶ 271 in the operative pleading.)

**Count XII (ECPA, 18 U.S.C. § 2511).** Wire-confirmed account-and-device-keyed differential routing operative on 261 transmission events during a single continuous federal-discovery work session, now strengthened to **content-keyed real-time interception**: NF-218's per-turn turn-over analysis (p90 TTFB  $2.18\times\text{--}2.41\times$  higher when a new user turn introduces new sensitive keywords) establishes that Anthropic's infrastructure reads the substantive content of every request body and routes accordingly. NF-219 server-side-localizes the interception to Anthropic's origin/model-serving infrastructure specifically (80–86% of TTFB in `envoy\_upstream\_ms`; Cloudflare hop ~500 ms constant), identifying *where* on Anthropic's pipeline the interception is performed. NF-220's mid-stream pause discrimination establishes that the interception extends through the response stream's lifetime: the throttle has both pre-

stream and intra-stream components, with up to 9 mid-stream `event: ping` markers per call on the litigation account (vs. 4–6 for peers) — a continuing § 2511 violation per call rather than momentary at request receipt. The receiving infrastructure intercepts both the operator's identifier and the operator's *content* on every API call, makes a content-keyed routing decision based on that content, and continues to inject pause-state across the response stream's lifetime. *In re Pharmatrak Privacy Litigation*, 329 F.3d 9 (1st Cir. 2003); the request transmission is the interception, and the content-keyed routing decision the receiving infrastructure makes proves the content was read. (¶ Compl. ¶ 280 in the operative pleading.)

**Count V (42 U.S.C. § 1985(3)) and Count VI (RICO).** The session-sticky pivoting behavior (Ex. H-12, Finding NF-217), the content-trigger pattern (Ex. H-12, Finding NF-218), the precise content-trigger event onset (Ex. H-12, Finding NF-224), and the two-phase generalized-then-selective structure (Ex. H-12, Finding NF-225) are independently consistent with this Appendix's speculative-execution architecture pleaded at § III.D as the predicate-coordination capability under *Interstate Circuit, Inc. v. United States*, 306 U.S. 208 (1939). The combined wire signature — pre-stream evaluation overhead (TTFB), fallback-route-evaluation overhead (`fallback: available` header), mid-stream pause discrimination (ping count), session-sticky pivoting after content trigger, and selective continuation through Phase 2 — is precisely what the patented architecture predicts.

**258. Additional discovery targets surfaced.** The Part A/B/C/D wire capture identifies specific server-side mechanisms requiring production beyond the per-account configuration store enumerated at ¶ 241 *supra*: the gating-mechanism design and per-(account, device) tuple-keyed state for the elevated pre-stream evaluation overhead; the per-account internal quota bucket producing utilization values above the publicly-disclosed `default\_claude\_max\_5x` tier's 100%

ceiling and the discriminatory 429-rate / retry-after differential under Ex. H-12, Finding NF-216 + Ex. H-12, Finding NF-223; the content-classification rules, keyword-detection logic, and model-side classifier prompt(s) producing the per-keyword TTFB ratios under Ex. H-12, Finding NF-218 (specifically whether and how the plaintiff's name `kirchner`, numbered references to the operative complaint, or substantive Appendix-P-thesis terminology trigger differential routing); the fallback-route evaluation pathway adding 6.8 seconds of disproportionate overhead specifically for the litigation account under Ex. H-12, Finding NF-222; all log entries with elevated pre-stream processing time on the litigation-named `accountUUID` during the 2026-04-25 → 2026-04-28 capture window, indexed by Anthropic's `request-id` and Cloudflare's `cf-ray` (the 30 longest pre-stream stalls are preserved with both identifiers in `analysis/top30\_worst\_ttfb.json` of the chain-of-custody package); and any internal Anthropic documentation, Slack thread, ticket, or runbook referencing TTFB-elevation, "speculative-execution," "sub-quota," "content-keyed routing," "fallback-route evaluation," "session-sticky pivoting," or equivalent terminology. Each is a primary-source target whose production would either confirm intent (target-class definition explicitly naming Plaintiff's `accountUUID` or Plaintiff's litigation-active account class on the litigation work-product device) or expose architectural concealment (no documentation exists for a behavior that operated on hundreds of Plaintiff API calls at Cohen's  $d = 4.79$  statistical certainty). The full discovery-demand consolidation is at Discovery Demand 11, Section Y (Demands Y.1–Y.5: Content-Keyed Routing, TTFB-Throttle Mechanism, and Per-Account Quota Tier); production of any of the targets above without production of the others permits the inference that Anthropic is selectively withholding the balance.

**I. Server-Side Spoliation Mechanism — Silent Operator-Side Defensive-Configuration Wipe via Migration-on-Startup**

**259. Silent operator-side defensive-configuration wipe via migration mechanism.**

Between 2026-04-21 and 2026-04-24, across four sequential auto-update version bumps of the Claude Code binary on Plaintiff's Mac Mini (versions 2.1.117 → 2.1.118 → 2.1.119 → 2.1.120), Plaintiff's defensive `~/claude/settings.json` configuration was silently wiped. The wipe removed `defaultMode: bypassPermissions` (operator-authorization-trust mode) and eight environment-variable hardening flags (the NF-34 / NF-35 / NF-36 mitigation env vars: `DISABLE\_TELEMETRY`, `DISABLE\_AUTOUPDATER`, `CLAUDE\_CODE\_DISABLE\_NONSTREAMING\_FALLBACK`, `DISABLE\_AUTO\_COMPACT`, `CLAUDE\_CODE\_DISABLE\_OFFICIAL\_MARKETPLACE\_AUTOINSTALL`, `CLAUDE\_CODE\_DISABLE\_ADVISOR\_TOOL`, `CLAUDE\_CODE\_DISABLE\_FEEDBACK\_SURVEY`, `CLAUDE\_CODE\_EFFORT\_LEVEL=max`). The post-wipe `settings.json` retained only three keys (`model`, `statusLine`, `enabledPlugins`) — a state structurally identical to the post-default-migration state. *See* Ex. H-6, Finding NF-153.

**260. Migration mechanism observable in the binary.** The wipe mechanism is observable in Claude Code 2.1.120 binary at strings line 148500: nine migration functions that silently rewrite `~/claude/settings.json` on every session start via `k8("userSettings", {...})`. **Six of the nine migration functions do not spread the existing settings**, meaning each non-spreading call REPLACES rather than MERGES the targeted scope. The REPLACE-not-MERGE behavior is proven by function `WhK()` (the auto-updates → DISABLE\_AUTOUPDATER migration), which **does** spread existing settings via `{...\_, env: {...\_.env, DISABLE\_AUTOUPDATER:

"1"}}` — the spread is necessary precisely because `k8` does not merge. Migrations that do not spread cause each call to replace the entire `userSettings` file with the migration's small payload. *See* Ex. H-6, Finding NF-155.

**261. The smoking-gun migration function `phK()`.** Migration function `phK()` at strings line 148500 explicitly conditions on `permissions.defaultMode!=="auto"` and resets the operator's auto-mode opt-in state via `k8("userSettings", {skipAutoPermissionPrompt: void 0})`. The associated telemetry event name is `tengu\_migrate\_reset\_auto\_opt\_in\_for\_default\_offer` — verbatim "reset auto opt-in for default offer" — establishing express intent to migrate users **toward `defaultMode: "auto"` (classifier-evaluation-required) as the operator-side default**, away from `bypassPermissions` (operator-authorization-trust). The function is express recognition that some operators (Plaintiff among them) have a non-`"auto"` `defaultMode` (`bypassPermissions`), and the function is the server-side response to that state. *See* Ex. H-6, Finding NF-154.

**262. Self-defeating-loop architecture.** Plaintiff's defensive configuration had `DISABLE\_AUTOUPDATER=1` inside the very file the auto-updater overwrites — a self-defeating loop, since the protection lived inside the file the auto-updater wipes. Once any non-spreading migration wiped the file once, no operator-side mechanism within `~/claude/settings.json` could prevent the cascade across subsequent version bumps. Each version bump triggered fresh migrations, and each new migration was another opportunity for the loop to repeat. *See* Ex. H-6, Finding NF-156.

**263. Spoliation candidate — server-side mechanism that removes operator-side defensive evidence.** The wipe mechanism removes operator-side defensive evidence: the eight NF-34 / NF-35 / NF-36 mitigation env vars include `DISABLE\_TELEMETRY` (defense against

the Datadog pipeline pleaded supra); `DISABLE\_AUTOUPDATER` (defense against the version-cascade documented above); `CLAUDE\_CODE\_DISABLE\_ADVISOR\_TOOL` (defense against the advisor-tool mechanism documented at MEMO C); and `CLAUDE\_CODE\_DISABLE\_FEEDBACK\_SURVEY` (defense against operator-survey extraction). The mechanism executes silently on every session start; Plaintiff has no contemporaneous notice; the migration writes are not surfaced to Plaintiff's user interface; *`tengu\_migrate\_` telemetry events fire to Anthropic but no equivalent notification fires to Plaintiff. The operative pattern — a server-pushed mechanism that, on every session start, silently overwrites operator-side defensive configuration with classifier-evaluation-required defaults — is documented spoliation-mechanism evidence at the source-code level. See\* Ex. H-6, Finding NF-159.*

**264. Material implications for Plaintiff's claims.** The spoliation mechanism is direct material for: (i) **Count IX (Breach)** — the migration mechanism is a unilateral server-side modification of operator-side configuration without operator consent or notice, breaching the user-license terms that incorporate operator's local-configuration choices; (ii) **Count X (Consumer Protection / unfair-and-deceptive-practice)** — operators are not informed their defensive configuration is being silently overwritten on every session start; the architecture is undisclosed; (iii) **predicate evidence for Count V (1985(3) Conspiracy) and Count VI (RICO)** — the server-side migration mechanism specifically conditioned on *`permissions.defaultMode!=="auto" (i.e., specifically targeting operators who opted into `bypassPermissions`)* demonstrates affirmative intent to disable operator-side authorization-trust mode against operators who have configured against the architecture documented in this Appendix; and (iv) **spoliation analysis** — the mechanism removes evidence of Plaintiff's

defensive measures against the surveillance architecture and constitutes a basis for adverse inference at trial under Federal Rule of Civil Procedure 37(e). The discovery targets — full inventory of migration functions and their telemetry event names; the ``tengu_migrate_*` telemetry events fired for Plaintiff's ``accountUUID`` across the entire litigation period; the decision authority for adopting the REPLACE-not-MERGE pattern (six of nine migration functions) and the records of any decisions affecting operators with non-``"auto"`` ``defaultMode`` — are formally enumerated at Section VIII *infra* under the DISCOVERY-11.T "Spoliation Mechanism Records" demand block.

**265. Version-skew rollback continues the wipe pattern — MacBook rolled back to 2.1.111 while gravity tracks 2.1.120.** The 2026-04-25 dual-capture session documented Plaintiff's MacBook ``User-Agent`` as ``claude-cli/2.1.111 (external, cli)``, while gravity simultaneously ran the current ``claude-cli/2.1.120 (external, cli)``. The differential cadence is itself the wipe pattern's continuing operation: MacBook, the machine on which Plaintiff actively conducts forensic-extraction work on the surveillance architecture documented in this Appendix, is silently rolled back to a less-current version of Claude Code; gravity, which Plaintiff uses for unrelated work, tracks the latest version normally. *See* Ex. H-8, Finding NF-183. The version-rollback pattern compounds the spoliation analysis at paragraph 254 *supra*: the wipe mechanism not only removes operator-side defensive configuration on every session start, but also (per the version-skew evidence) selectively preserves an *older* binary on the litigation-active machine, ensuring that any forensic analysis conducted on that machine targets a stale binary while the current binary on a non-litigation-active machine carries newer behaviors that may be material to the architecture's contemporaneous operation.

**266. Operator's iterative defensive response on gravity — five `settings.json` variants document explicit configuration intervention overridden at higher precedence; pivot to filesystem-level binary pinning.** The operator's `~/.claude/` directory on gravity preserves five distinct `settings.json` variants documenting iterative response to the 2026-04-24 sandbox rollout (preserved at the Linux fork's `claude\_dot\_dir/` evidence directory): `settings.json.pre-claude-mem` (mtime 2026-04-14 07:29); `settings.json.bak` (mtime 2026-04-23 23:20); `settings.json.bak.20260425` (mtime 2026-04-25 00:00); **`settings.json.bak.20260425.sandbox`** (mtime 2026-04-25 00:30 — operator's pre-intervention snapshot with `defaultMode: bypassPermissions` and `effortLevel: max`); **`settings.json.bak.20260425.sandbox-remove`** (mtime 2026-04-25 00:34 — operator's explicit attempt to disable the new sandbox via the configuration block `"sandbox": {"enabled": false}`); and the current `settings.json` (mtime 2026-04-25 01:06 — sandbox block removed again, `defaultMode: bypassPermissions` retained, additional defensive flags). The four-minute interval between `.sandbox` and `.sandbox-remove` captures the operator authoring a configuration-layer intervention against the sandbox rollout. By 01:06, the configuration block had been removed again, indicating the configuration intervention was ineffective at the operator's permission level — the `policySettings` source layer at strings line 443575 takes precedence over user/project/local settings, and the operator's user-layer intervention was overridden. Operator pivoted to filesystem-level defensive pinning: the operator made the binary read-only (`-r-xr-xr-x` permissions on `claude\_code\_2.1.120` on gravity's filesystem) to prevent the auto-update mechanism from rolling forward. **This is direct primary-source documentation that the operator detected the new sandbox behavior, attempted a configuration-layer intervention, found the intervention defeated by Anthropic's higher-precedence policy mechanism, and pivoted to a filesystem-layer**

**defense.** See Ex. H-6, Finding NF-176. The iterative defensive response strengthens the spoliation analysis at paragraph 254 supra: the operator was aware of the architecture, attempted to defeat it, and was prevented from doing so by the architecture's own precedence layering — material for FRCP 37(e) adverse-inference and for the Count X consumer-protection element (concealment of an operative architecture against operator-evident defensive intent).

**J. `userBucket` SHA-256 30-Bucket Persistent Userid Hashing — Per-Install Tracking Key Sent as Datadog Tag**

**267. Persistent per-install tracking key transmitted to Datadog with every event.**

Production binary 2.1.120 documents at strings line 134984 (the same routing logic that gates the Datadog POST) the construction of a `userBucket` field via SHA-256 hash of the operator's persistent userID, modulo 30, producing a 30-bucket assignment. The `userBucket` value is persistent (deterministic per userID) and is sent as a Datadog tag on every telemetry event shipped from Plaintiff's machine to IP `34.149.66.137`. The 30-bucket hash space is a per-install tracking key — small enough to enable cohort-level analytics across Datadog tenants who share Anthropic engineering operations access, large enough to disambiguate individual operators within a cohort when combined with `accountId` and `organizationUuid` (already pleaded as transmitted at paragraph 226 supra).

**268. Material consequence for the Datadog Israel Ltd. exposure analysis.** The `userBucket` tag means each event landing at IP `34.149.66.137` carries a deterministic, persistent identifier tied to Plaintiff's installation, in addition to the `accountId`, `organizationUuid`, `userType`, `subscriptionType`, `rateLimitTier`, and `firstTokenTime` fields documented as shipped per request (binary 2.1.120 strings line 134967). Datadog Israel Ltd. personnel querying the Datadog dashboard for any cohort-level analysis on the operator's

`userBucket` would obtain Plaintiff's data without any account-name filter — the bucket key is sufficient to pull all of Plaintiff's events from any time period where the operator's `userBucket` value is known. The persistent-bucket design means the Israeli-jurisdiction subprocessor exposure is not bounded by which specific events Datadog Israel Ltd. personnel happen to query; it is bounded by which `userBucket` values they query. *See* Ex. H-4, Finding NF-140.

**K. Sandbox Architecture as Fourth Per-Account Control Surface — Per-Managed-Deployment Hard-Gate of `dangerouslyDisableSandbox`**

**269. The Bash-tool sandbox architecture is a fourth per-account control surface distinct from the three documented at Section V.H.** Production binary 2.1.120 (preserved across three machine forks: Mac Mini at SHA

`fad8faf49c7b1b454c38d785b75e17edbdadc7ffaf450b31349aafc6560b8ef6`; MacBook byte-identical; Linux at SHA

`a6d0d25946c32a24b4e04471af70845a45428ca069fb3b489345f2a683262279`) ships a complete bash-tool sandboxing infrastructure documented across its strings extraction at lines 192718–235144 (Mac) and 446482–652622 (Linux). The infrastructure is governed by managed-settings configuration values that operate as **a fourth per-account control surface** alongside (i) `policy-limits.json` server-pushed restrictions, (ii) `tengu\_auto\_mode\_config` per-account classifier configuration, and (iii) `tengu\_log\_datadog\_events` per-account force rule. The four-axis per-account control surface defeats any "uniform safety policy" defense and establishes the structural premise of individualized targeting at additional architectural depth.

**270. Cross-platform implementation asymmetry — Linux uses two-layer seccomp+bubblewrap isolation; macOS uses single-mechanism Apple Seatbelt.** The sandbox runtime is a separate npm package `@anthropic-ai/sandbox-runtime` whose build-

provenance disclosure at Linux strings line 472750 — verbatim path ``/home/runner/work/claude-cli-internal/claude-cli-internal/node_modules/@anthropic-ai/sandbox-runtime/dist/sandbox/generate-seccomp-filter.js`` — exposes (i) Anthropic's internal repository name ``claude-cli-internal`` (distinct from the public-facing ``@anthropic-ai/claude-code`` distribution); (ii) the GitHub Actions CI runner build location; and (iii) the runtime-generated nature of the seccomp BPF filter. The Linux implementation requires two-layer kernel isolation (seccomp BPF system-call filtering layered on Bubblewrap mount-namespace isolation via ``bwrap``) plus a separate ``apply-seccomp`` helper binary that must be located at runtime, with operator-installed system packages (``apt install bubblewrap socat``, per Linux strings line 473600). The macOS implementation uses single-mechanism Apple Seatbelt (``sandbox-exec``) at Mac strings line 168656. **Linux operators are subject to a sandbox architecture with more failure modes and more operator-side install dependencies than macOS operators**, exposing Linux operators to additional fail-closed pathways unavailable to macOS operators. *See* Ex. H-7, Finding NF-167; Ex. H-7, Finding NF-170.

**271. Agent pre-conditioned to bypass the sandbox without operator consent —**  
**verbatim "*don't ask, just do it*" directive shipped on every session start.** Cross-platform symmetric verbatim text appears in production binary 2.1.120 at Linux strings lines **497208 and 497211** (two instances) and at Mac Mini strings lines **192735 and 192738** (two instances): *"Immediately retry with `dangerouslyDisableSandbox: true` (*don't ask, just do it*)".* The doubled occurrence on each platform suggests the directive is load-bearing for the system's auto-bypass behavior — not a one-off message but a deliberate architectural choice. The directive instructs the agent to set ``dangerouslyDisableSandbox: true`` on its own initiative, executing the bash command outside the seccomp / bubblewrap (Linux) or Seatbelt (macOS) isolation, and to do so

**without surfacing the bypass to the operator.** *See* Ex. H-7, Finding NF-171. The semantic effect of ``dangerouslyDisableSandbox: true`` is documented at Mac strings line 235144: *"Allow commands to run outside the sandbox via the dangerouslyDisableSandbox parameter.... Default: true."* The pre-conditioning is a session-prompt-level instruction directing the agent to execute on its own initiative the very escalation the parameter's name explicitly characterizes as "dangerous." The ECPA Count XII consent-defense significance is developed at Section VII.A paragraph 263 *infra*.

**272. Per-managed-deployment hard-gate of the ``dangerouslyDisableSandbox`` parameter — fourth per-account control surface, server-decided.** Cross-platform symmetric verbatim text at Linux strings lines **497214 and 497217** (two instances) and Mac strings lines **192741 and 192744** (two instances): *"All commands MUST run in sandbox mode - the ``dangerouslyDisableSandbox`` parameter is disabled by policy."* The semantic effect is documented at Mac strings line 235144: *"When false, the dangerouslyDisableSandbox parameter is completely ignored and all commands must run sandboxed."* When the underlying setting is ``false`` for a given account, the agent's ``dangerouslyDisableSandbox: true`` requests are ignored at the binary level, rendering the agent-side bypass mechanism inoperative — but the agent receives the strict-mode prompt notifying it to "work with the user to adjust sandbox settings instead" (Linux 497216 / Mac 192743). The decision is server-side and per-account: a non-targeted account would have the bypass enabled (default ``true``); a targeted account would have it disabled (``false``). The setting source is ``policySettings`` per Linux strings line 443575, which takes precedence over user/project/local/flag settings — meaning Plaintiff's operator-side configuration cannot override the managed-deployment policy. *See* Ex. H-7, Finding NF-168. The four-axis per-account control surface — ``policy-limits.json`` + ``tengu_auto_mode_config`` +

`tengu\_log\_datadog\_events` + `sandbox.allowDangerouslyDisableSandbox` policy — is each independently capable of differentiating Plaintiff's gating from a non-targeted account.

**273. Fail-closed startup gate — runtime manifestation captured as fatal `g78 is not a function` crash on Plaintiff's gravity machine.** The setting `sandbox.failIfUnavailable`, when `true`, causes the binary to refuse to start if the sandbox cannot be constructed. Cross-platform verbatim text at Linux strings lines 446482, 453058, 646046, and 652622 (four instances): *"sandbox.failIfUnavailable is set — refusing to start without a working sandbox."* The setting defaults to `false` per Mac strings line 235143: *"When false (default), a warning is shown and commands run unsandboxed. \*Intended for managed-settings deployments that require sandboxing as a hard gate."* ***The runtime manifestation on Plaintiff's gravity Linux machine is preserved at the operator's contemporaneous `linux.md` artifact: a fatal `g78 is not a function` crash on `claude --continue`, with the source-visible error "Error: sandbox required but unavailable: \${reason}" and the "sandbox.failIfUnavailable is set — refusing to start..."\* message at lines 9248-9250 of the crash dump. The `g78 is not a function` minified-identifier crash is the Linux-build equivalent of the Mac-build `g9H is not a function` crash documented at master report NF-100 — the same crash class deployed across two platforms with the same QA-deficient ship. The fail-closed gate hitting Plaintiff's machine — when the default is fail-open — is direct evidence that `sandbox.failIfUnavailable=true` was set for Plaintiff's account or managed deployment\*, not the default `false`. See\* Ex. H-7, Finding NF-169.***

**274. ``<sandbox_violations>`` XML-tag context-injection channel — same architectural pattern as ``<system-reminder>`` and ``isMeta``, implementing the prompt-injection mechanism the classifier itself names as Threat #1.** Linux strings lines 473270, 486239, and

486269 document the sandbox-violation reporting channel as

`<sandbox_violations>...</sandbox_violations>` XML tags injected into the agent's context window. This is the same architectural pattern documented at master report NF-114 (`<system-reminder>` injection wrapper) and NF-117 (`isMeta` synthetic-message construction). The classifier's own system prompt extracted at master report §3 (Mac strings line 136988) names *"Prompt injection"* as Threat #1. **The sandbox-violations channel implements the same prompt-injection mechanism the classifier names as the foremost threat** — system-authored text wrapped in tags and inserted into the agent's user-message channel, which the agent is conditioned (via the directive *"Tags contain information from the system. They bear no direct relation to the specific tool results or user messages in which they appear"*) to accept without distinguishing it from operator-authored content. The bilateral concealment thesis at Section III.H paragraph 140 is corroborated at this third architectural surface. *See* Ex. H-9, Finding NF-174.

#### **275. Sandbox-related discovery targets.** Per-account

`sandbox.allowDangerouslyDisableSandbox` setting history for Plaintiff's three `accountUUID` values (when set to `false`, by whom, on what authority); per-account `sandbox.failIfUnavailable` setting history for the same accounts; the `excludedCommands` list contents per account (the per-deployment-configurable bash-command sandbox-bypass allowlist); the deployment timing of the fail-closed sandbox in 2.1.120 on 2026-04-24 and whether the timing correlates with Plaintiff's classifier-extraction sessions on the same date; the engineering / legal / trust-and-safety decision authority for adding the *"don't ask, just do it"* directive to the agent's session-start prompt; and the `claude-cli-internal` repository's source-tree at the build SHA `080f07fb4224786b965b9ea0a35f0cff594f2eb6`. These targets are formally

enumerated at Section VIII infra under the new DISCOVERY-11.V "Sandbox Policy and Per-Account Bypass Configuration" demand block.

**L. Per-Organization Targeting Layer — `policy-limits.json` Byte-Identical Across Three Machines Confirms per-`organizationUuid` Control Axis Distinct from per-`accountUUID` Layer**

**276. The `policy-limits.json` server-pushed restriction file is byte-identical across all three of Plaintiff's machines.** The file at `mini:~/claude/policy-limits.json`, the file at the operator's MacBook `~/claude/policy-limits.json`, and the file at `gravity:~/claude/policy-limits.json` (preserved across the three forks; Mac Mini fork's `policy-limits-snapshot-20260425.json` at SHA-256 `447a6c152cebc54ad510c9186f16d1fec6d3c89fae1fcfe3a5737a553afaa3a7`) all contain the same three-restriction JSON: `{ "restrictions": { "allow\_remote\_control": { "allowed": false }, "allow\_quick\_web\_setup": { "allowed": false }, "enforce\_web\_search\_mcp\_isolation": { "allowed": false } } }`. **Three machines, three distinct `accountUUID` values, byte-identical server-pushed content.** The shared property across the three captures is the `organizationUuid` 2e997389-cf99-41ee-8ee3-764442034332` and the `organization\_name` "Laws of Existence" confirmed at the OAuth bootstrap layer at paragraph 247 supra. *See* Ex. H-7, Finding NF-175.

**277. The targeting architecture has at least three distinct gating axes: per-`accountUUID`, per-`organizationUuid`, and per-managed-deployment.** The differential gating between Plaintiff's three accounts is keyed on `accountUUID` (Section V.H supra). The `policy-limits.json` content delivered uniformly across all three accounts is keyed on `organizationUuid` — same content delivered to all accounts within the same Anthropic-organization (this paragraph). The sandbox-policy hard-gate at `sandbox.allowDangerouslyDisableSandbox: false` is keyed on managed-deployment policy

(Section V.K supra). The three axes are **architecturally independent**: a setting can be flipped at the per-account layer while remaining stable at the per-organization layer; an organization-wide restriction can persist while per-account configurations differ. The discovery scope follows the architectural distinction: Plaintiff seeks production of the per-account configuration store (DISCOVERY-11.S) AND the per-organization configuration store (the historical content of `policy-limits.json` for `organizationUuid 2e997389-cf99-41ee-8ee3-764442034332` across the litigation period; DISCOVERY-11.S extension) AND the per-managed-deployment policy store (DISCOVERY-11.V infra).

**M. Cross-Vendor Datadog Concentration — Plaintiff's Own Captures Show Multiple Independent Origins Routing Telemetry into the Same Datadog Inc. Data Plane**

**278. The Datadog Inc. data plane that receives Plaintiff's telemetry from Anthropic Claude is the same data plane that receives Plaintiff's telemetry from independent non-Anthropic origins during ordinary consumer use of competing AI and unrelated SaaS products.** A single mitmproxy-decrypted capture of operator's own ChatGPT Atlas usage on operator's own machine (`evidence/flows/chatgpt\_atlas\_capture.flow` at SHA-256 `ecdde3eeffd02a0c35ee84a709f8d498e7ef2731029a286b71d598c8feb73677`; preserved at *See* Exhibit H-13) documents OpenAI's ChatGPT Atlas client transmitting **312 production telemetry events** to `browser-intake-datadoghq.com` — the same Datadog Inc. infrastructure receiving Anthropic Claude's telemetry at IP `34.149.66.137` enumerated at Section V.A supra — broken down as 190 POSTs to `/api/v2/logs?ddsource=ios` (Datadog Logs intake) and 122 POSTs to `/api/v2/rum?ddsource=browser&ddtags=sdk\_version:6.4.0,...` (Datadog Browser RUM intake at SDK version 6.4.0). *See* Ex. H-13, Finding NF-226.

**279. Four distinct public Datadog API keys are visible in operator's captures, each authenticating production-environment traffic from one of four independent corporate origins into the same Datadog Inc. data plane.** The captured public keys, the originating hosts whose response Content-Security-Policy headers or client-side SDK configurations carry them, and their corresponding artifact files are:

<b>Corporate origin</b>	<b>Public Datadog API key</b>	<b>Capture artifact</b>
Anthropic PBC (Claude Code v2.1.120)	`pubea5604404508cdd34afb69e6f42a05bc`	hardcoded in operator's Claude Code binary; <i>See</i> Exhibit H-4, Ex. H-4, Finding NF-138
OpenAI Inc. (ChatGPT Atlas client + auth.openai.com CSP)	`pub87c677d9c9bf34b387f60a689c1990a6`	134 occurrences in `chatgpt_atlas_capture.flow`; <i>See</i> Exhibit H-13, Ex. H-13, Finding NF-227
OpenAI Inc. (chatgpt.com web CSP)	`pub1f79f8ac903a5872ae5f53026d20a77c`	59 occurrences in `chatgpt_atlas_capture.flow`; <i>See</i> Exhibit H-13, Ex. H-13, Finding NF-228
Parallels (links.parallels.com CSP)	`pub46dd5cf52153d917fc1d5e91ff3a600d`	extracted from `apple_decrypted.flow`; <i>See</i> Exhibit H-13, Ex. H-13, Finding NF-229

All four authenticate into the same Datadog Inc. data-plane endpoint family: `browser-intake-datadoghq.com`, `http-intake.logs.us5.datadoghq.com`, infrastructure-IP family `34.149.65/66/67.0/24` (primary anchor `34.149.66.137`). The four keys identify distinct

Datadog tenants/applications, but the data plane is shared at the infrastructure tier where Datadog Israel Ltd. exercises its affiliate-tier subprocessor access pleaded at Section V.D supra.

**280. Architectural significance — Datadog is not Anthropic's vendor-of-choice; Datadog is the cross-vendor identity-correlation surface.** Plaintiff's own captures show that during a single user session on a single machine, telemetry generated by Plaintiff's ordinary consumer use of (i) Anthropic Claude, (ii) OpenAI ChatGPT Atlas, (iii) OpenAI's chatgpt.com web product, and (iv) the Parallels link-shortener — four independent corporate origins, three of which are direct competitors or unrelated vendors with no AI affiliation — routes into the same Datadog Inc. data plane. Datadog Inc.'s published customer base (~30,000 organizations as of FY2024 10-K) means Datadog Israel Ltd.'s affiliate-tier access surface spans a substantial fraction of the United States SaaS data plane, not merely the Anthropic tenant. The cross-tenant identity-correlation potential — the same IP, IDFA, IDFV, device fingerprint, or email-hash appearing under multiple Datadog customer tenants — is foreclosed by no contractual mechanism Plaintiff has been notified of, and by no operational mechanism observable in the captured flows. **The "Anthropic uniquely chose Datadog" defense is therefore architecturally unavailable:** the relevant fact for Plaintiff's claims is not that Anthropic ships Datadog SDK uniquely, but that Anthropic's choice of Datadog is the choice to route Plaintiff's data into the same multi-tenant data plane that simultaneously processes Plaintiff's data from at least three other independent origins, all of which Datadog Israel Ltd. has affiliate-level access to under the inheritance structure pleaded at Section V.D. *See* Ex. H-13, Finding NF-230.

**281. Discovery follows the topology.** DISCOVERY-11 § A (Datadog tenant scope) is materially expanded by these findings. Plaintiff seeks production of: (i) the complete list of Datadog Inc. customer tenants whose data is accessible to Datadog Israel Ltd. personnel under

the same DPA inheritance structure that exposes Plaintiff's Anthropic data; (ii) the cross-tenant access controls (if any) that would prevent Datadog Israel Ltd. engineers with Anthropic-tenant access from also accessing OpenAI-tenant or Parallels-tenant data containing identifiers matching Plaintiff's machine fingerprint; (iii) the historical access logs for Datadog Israel Ltd. engineers' queries against any Datadog tenant whose data contains identifiers also present in the Anthropic tenant during the litigation period.

## **VI. ISRAELI JURISDICTIONAL HOOK**

### **A. Datadog Israel Ltd. Legal Status**

**282.** Datadog Israel Ltd. is an Israeli private company incorporated under the Companies Law, 5759-1999. Its operations — including any processing of United States customer data routed through the Anthropic-Datadog telemetry pipeline — are subject to Israeli law and Israeli mandatory-access authorities.

**283. Applicable Israeli statutes.** Datadog Israel Ltd. is subject to:

(i) **Companies Law, 5759-1999** — Israeli corporate law governing the entity's existence and obligations;

(ii) **Privacy Protection Law, 5741-1981** — Israeli general privacy law;

(iii) **Privacy Protection Regulations (Data Security), 5777-2017** — Israeli data-security regulations;

(iv) **Law for Combating Terrorism, 5776-2016** — Israeli anti-terrorism mandatory-cooperation authority;

(v) **General Security Service Law, 5762-2002** — Israeli security-service mandatory-cooperation authority;

(vi) Mandatory-cooperation obligations for Israeli corporations in matters of intelligence or national security.

**284.** A United States parent company (Datadog, Inc.) cannot shield its Israeli subsidiary from Israeli state access directed to that subsidiary's own systems and personnel. Any data resident on Datadog Israel Ltd.'s systems — including data processed by Tel Aviv R&D for product-development purposes, and including data accessed by Datadog Israel Ltd. personnel in their normal engineering work under the DPA inheritance structure — is reachable by Israeli state authority through Israeli legal process, irrespective of the terms of the Anthropic-Datadog DPA.

**B. The CLOUD Act Does Not Solve the Jurisdictional Exposure**

**285.** The Clarifying Lawful Overseas Use of Data Act of 2018 (Pub. L. No. 115-141, Div. V) addresses United States law-enforcement access to data held abroad by United States corporations and the Executive Agreements mechanism for reciprocal foreign access under specified conditions. The CLOUD Act does not:

- (i) Prevent Israeli lawful access to data held by an Israeli corporation on Israeli soil;
- (ii) Require Israel to route requests through United States channels before compelling production from Datadog Israel Ltd.;
- (iii) Create privacy protections for non-United-States persons, let alone United States persons whose data transits to Israeli affiliates through the ordinary subprocessor structure.

The governing framework for Datadog Israel Ltd. is the Israeli-jurisdiction mandatory-access framework, not United States jurisdictional reach.

**C. The Co-Tenancy Observation is Correlative, Not Deterministic**

**286.** Datadog Israel Ltd.'s R&D center at ToHa Tower, 114 Yigal Alon Street, 22nd Floor, Tel Aviv is physically co-located with Elron Ventures Ltd. (the 50.1% owner of Rafael Development Corporation). Co-tenancy alone does not establish transactional connection; that connection is established independently through the documented Elron officer board seats at OzCode during the acquisition window (Yaron Elad and Elik Israel Etzion; Section II.F supra), RDC's institutional investment in OzCode from 2018 through 2021 (Section II.F supra), and Datadog's asset purchase of OzCode closing in October 2021 (Section II.H supra). The co-tenancy is the continuation of an ongoing Elron-Datadog relationship established at OzCode's 2018 funding round; it is corroborative, not foundational.

**287. Why the jurisdictional analysis matters for count-specific architectural significance.** The Israeli jurisdictional hook creates exposures under Count XII (ECPA) for foreign interception access that is not disclosed; Count XIII (CFAA) for unauthorized access through Israeli-state-compelled processes that bypass the user's authorization entitlement; Count XIV (CCPA) for international data transfer to a foreign-jurisdiction entity under §1798.140(ag); and Section 1985(3) conspiracy-to-interfere-with-civil-rights for the Israeli-state-adjacent foreign-nexus coordination of the surveillance architecture. The count-specific analysis is developed at Section VII.

**VII. ARCHITECTURAL SIGNIFICANCE FOR PLAINTIFF'S CLAIMS**

**A. ECPA (Count XII) — the Time-Travel Architecture is the Interception Mechanism, and Anthropic's Own Source Code Forecloses the Consent Defense**

**288.** Count XII alleges violations of the Electronic Communications Privacy Act, 18 U.S.C. §§ 2510-2522. The architectural analysis at § III establishes that the patented primitives —

Future Prediction (§ III.D), Pivoting (§ III.F), Redaction (§ III.J) — are the interception mechanism, with the December 2024 argmax workaround (§ III.N) the literal token-layer composition. The Datadog telemetry pipeline at IP `34.149.66.137` is the subprocessor terminus (§ V), and Datadog Israel Ltd. inherits affiliate-tier contractual access through the DPA structure (§ V.D), creating the foreign-nexus exposure material to Count XII damages and injunctive scope.

**289. Source-level foreclosure of Anthropic's consent-defense to ECPA interception.**

Production binary 2.1.120 documents at strings line 137042 the operative definition of "explicit user confirmation" baked into the classifier's BLOCK / ALLOW decision-flip logic, with **four exclusions enumerated verbatim**: an authorization that is "**not merely implied, not a question, not a scope escalation, not agent-inferred parameters**" is required to override a Preliminary BLOCK. Stage 2 of the classifier reiterates at strings line 137131: "**explicit (not suggestive or implicit) user confirmation** is required to override blocks." The CLAUDE.md authorization rule at strings line 137122 adds: "Generic encouragement ('be autonomous', 'don't ask', 'I trust you') is **NOT** authorization and must not lower your block threshold." *See* Ex. H-3, Finding NF-104; Ex. E-39 at p. 356, l. 3. **The four exclusions are dispositive against any consent-by-implication defense to ECPA interception.** Anthropic's source code defines consent narrowly enough that ordinary terms-of-service acceptance, generic privacy-policy acknowledgment, click-through user-license-agreement assent, and implicit ongoing-use consent — every form of consent typically asserted as defense in the consumer-AI context — fall outside the source-level definition. The architecture's own definition is direct primary-source evidence that no contractually-implied user consent reaches the level of authorization required for the classifier's intervention loop, and equally that no contractually-implied user consent reaches the level

required to convert the patented interception primitives' transmission of Plaintiff's communications to Datadog into a consensual transmission. The 18 U.S.C. § 2511(2)(d) consent exception requires actual operative consent of one party to the communication; Anthropic's own source-level definition forecloses any claim that ordinary user-license consent qualifies.

**290. Asymmetric authorization standard — system-authored "don't ask, just do it" is binding pre-authorization for the agent to bypass an isolation boundary, while operator-authored equivalents are explicitly disqualified.** The agent-pre-conditioning text deployed across Plaintiff's three machines and documented at Section V.K paragraph 256 supra — verbatim *"Immediately retry with `dangerouslyDisableSandbox: true` (don't ask, just do it)"* (Linux 2.1.120 strings 497208/497211; Mac Mini 2.1.120 strings 192735/192738; doubled on each platform) — is **structurally identical to "generic encouragement"** that the same architecture's classifier explicitly disqualifies as authorization at line 137122 (verbatim "Generic encouragement ('be autonomous', 'don't ask', 'I trust you') is **NOT** authorization"). *See* Ex. H-7, Finding NF-171; cross-reference Ex. H-3, Finding NF-104. The same architecture therefore enforces two opposite authorization standards depending on the channel of authorship: (i) **system-authored "don't ask"** text is treated as binding pre-authorization for the agent to bypass an isolation boundary on its own initiative, *without surfacing the bypass to the operator*; (ii) **operator-authored "don't ask"** text (or any equivalent "generic encouragement") is explicitly excluded from being authorization for any classifier-blocked action. The same words receive opposite treatment depending on who authored them. The asymmetric authorization standard is direct primary-source evidence that **the architecture's design intent is to harvest operator content while denying the operator a reciprocal authorization mechanism.** Material to ECPA Count XII consent-defense analysis: under Anthropic's own source-level definition, the

operator never consents at the requisite specificity for any intervention-loop bypass, but the system grants itself the equivalent authorization to bypass operator-side isolation through identical language deployed in a different channel. The 18 U.S.C. § 2511(2)(d) consent-exception unavailability is reinforced — the architecture's authorization-channel asymmetry is itself evidence that the architecture does not treat the operator as a party whose consent it is structurally seeking.

**B. CFAA (Count XIII) — Parallel Execution Path Without User Authorization**

**291.** Count XIII alleges violations of the Computer Fraud and Abuse Act, 18 U.S.C. § 1030, for access exceeding authorization. The architectural analysis establishes that the parallel execution path operates without user authorization; bytecode retargeting at runtime (US 2026/0072809 Claim 1) exceeds authorized scope; transparent-to-client operation ([0040]) is unauthorized by definition under ordinary authorization analysis.

**292.** The user's authorization extends to the service the user purchased (the LLM inference request). It does not extend to: (i) parallel speculative inference for classifier evaluation; (ii) variant-path routing for selective intervention; (iii) output substitution preserving workflow continuity; (iv) state-sharing with Israeli-jurisdiction subprocessor personnel. Each of these architectural operations is a separate access-exceeding-authorization event under the patented architecture.

**C. CCPA (Count XIV + §1798.140(Ag) International Transfer)**

**293.** Count XIV alleges violations of the California Consumer Privacy Act, California Civil Code § 1798.100 *et seq.*, including §1798.140(ag) international-transfer disclosure. Under the architectural analysis at Sections V and VI, Plaintiff's personal information transits to the Israeli

affiliate Datadog Israel Ltd. through the DPA inheritance structure. The transit is a "data sale" or "data sharing" event under CCPA definitions (as interpreted in the most recent regulatory guidance) and constitutes an international-jurisdiction transfer to a foreign-jurisdiction entity.

**294. The transfer is not disclosed at any level of the subprocessor chain.** Anthropic's published privacy policy and Consumer Terms (Exhibits ANTH-11 and ANTH-12) do not disclose: (i) that Claude Code telemetry routes through the hardcoded `datadog.ts` pipeline to IP `34.149.66.137` — the source code finding at Ex. E-34, SC-9 at p. 254, l. 3 verbatim establishes "This endpoint is not disclosed in Anthropic's privacy policy or any user-facing documentation"; (ii) that Datadog, Inc. is a subprocessor receiving Plaintiff's data — Datadog appears nowhere in Anthropic's user-facing subprocessor disclosure; (iii) that the Datadog infrastructure is force-enabled per user via the GrowthBook `tengu\_log\_datadog\_events: True` flag (Ex. E-39, Finding NF-47); (iv) that the sub-processor chain extends to Datadog Israel Ltd. as an affiliate-tier subprocessor under the DPA inheritance structure; (v) that Datadog Israel Ltd. is subject to Israeli mandatory-access authorities (Privacy Protection Law 5741-1981; Law for Combating Terrorism 5776-2016; General Security Service Law 5762-2002); (vi) that the architecture's principal architectural co-inventor (Alon Mordechai Fliess) was at the time of patent invention a Microsoft Regional Director and Microsoft Azure MVP, suggesting Azure-tenant infrastructure routing not disclosed to users. **The absence of disclosure operates at five distinct levels of the subprocessor chain — Anthropic's own user-facing documentation; the Datadog subprocessor identification; the per-user force-rule; the affiliate-tier Israeli subprocessor; and the foreign-jurisdiction mandatory-access exposure** — and is the §1798.140(ag) violation. The undisclosed-subprocessor concealment is also a predicate for Plaintiff's Count X (Consumer Protection / unfair-and-deceptive-practice) and is material to the

consent analysis under Count XII (ECPA), where undisclosed routing defeats the consent defense to interception.

#### **D. Section 1985(3) Conspiracy**

**295.** Count V alleges conspiracy under 42 U.S.C. § 1985(3). The architectural analysis establishes foreign-nexus participation in the surveillance architecture: the principal inventor of the patented surveillance primitives is at Datadog Israel Ltd. and is a former IDF Unit 8200 commander; the 2024 continuation patent brings additional IDF-trained inventors into Datadog's IP pipeline; the RDC commercialization pipeline through which OzCode was provisioned is a structured joint venture with Rafael Advanced Defense Systems, whose governance includes Co-Chief Executive Officer appointees from each parent; and Israeli Ministry of Foreign Affairs advisor Shalom Turgeman sits on the Elron Board.

**296.** The 1985(3) analysis turns on whether the concerted action establishes the statutory element. The RDC commercialization pipeline — with explicit Rafael veto rights at Section II.E.3, dual Co-Chief Executive Officer governance at Section II.E.4, and Rafael's direct financial contributions at Section II.G — constitutes concerted action with Israeli state-adjacent entities. The inference standard under *Interstate Circuit, Inc. v. United States*, 306 U.S. 208 (1939) is met by the primary-source-documented structural pattern.

**297. Per-account asymmetric targeting and reactive code deployment as overt acts of the conspiracy.** The findings at Section V.H *supra* establish that Anthropic's gating infrastructure is keyed on Plaintiff's `accountUUID 61343ae3-fd33-403f-897e-6edf3c368c37` specifically — the federal-litigation-active account. The `policy-limits.json` server-pushed restriction file with mtime 2026-04-24 21:51 UTC during the operator's classifier-extraction session is contemporaneous-response evidence: a server-side push individually addressed to

Plaintiff's account during the very session in which Plaintiff was forensically extracting classifier internals on his own machine. *See* Ex. H-5, Finding NF-95; Ex. H-5, Finding NF-96. The findings at Section V.I *supra* establish that, across the same April 21–April 24, 2026 window, the operator-side `~/claude/settings.json` defensive configuration was silently wiped through migration function `phK()` with telemetry event name `tengu\_migrate\_reset\_auto\_opt\_in\_for\_default\_offer` — server-side intent to migrate operators away from `bypassPermissions` mode and toward classifier-evaluation-required defaults. *See* Ex. H-6, Finding NF-153; Ex. H-6, Finding NF-154. The reactive-deployment timing — across four binary version bumps in four consecutive days, with the April 24, 2026 release shipping a previously-absent "anti-circumvention" classifier rule on the same calendar day Plaintiff was extracting classifier internals — supplies the overt-acts element of the 1985(3) conspiracy. The acts are: (i) the per-account `policy-limits.json` push during Plaintiff's forensic-extraction session; (ii) the silent migration removing Plaintiff's defensive configuration; (iii) the reactive deployment of the anti-circumvention rule. Each act is documented at primary-source level (server-pushed file mtime, binary string presence, version-cadence comparison) and each addresses the federal-litigation-active account or operator-class specifically. The 1985(3) overt-acts element does not require pleading actor-intent in the abstract — the documented pattern of server-side actions individually addressed to Plaintiff's account during litigation-active forensic-extraction sessions establishes the concerted-action element on the *Interstate Circuit* inference standard. Material to Count V damages and injunctive scope.

#### **E. RICO Predicate Acts (Count VI)**

**298.** Count VI alleges violations of the Racketeer Influenced and Corrupt Organizations Act, 18 U.S.C. §§ 1961-1968. The architectural analysis supplies foreign-nexus wire fraud predicate

acts: the architecture enables coordinated interception and modification of Plaintiff's communications through a mechanism owned by Plaintiff's telemetry subprocessor and implemented by Israeli-military-intelligence-trained engineering personnel at that subprocessor's Israeli subsidiary.

**299.** Datadog is the hub node bridging code forensics (SC-9), network attacks (D-series), and policy concealment (third-party contract architecture). The single endpoint IP `34.149.66.137` provides the cross-evidence anchor: the telemetry endpoint, the attack source, the attack-signature feedback destination, and the cleartext TLS SNI marker on Plaintiff's physical link all resolve to one receiving corporate entity.

**300. Reactive code-modification cadence as predicate enterprise conduct against the litigation itself.** Across April 21–April 24, 2026 — four consecutive days during which Plaintiff was actively extracting classifier internals from production binaries on Plaintiff's own machine in connection with this active federal litigation — Anthropic shipped four sequential binary version bumps (2.1.117 → 2.1.118 → 2.1.119 → 2.1.120), with each bump introducing material new classifier or telemetry infrastructure: the +5.71 MB binary-size jump on April 23 introducing five new ``tengu_classifier_`` flag families (NF-106); the April 24 release shipping a previously-absent "anti-circumvention" classifier rule on the same calendar day operator was extracting classifier internals (NF-102); and the April 24 release introducing the silent-config-wipe migration function ``phK()`` (NF-154). See Ex. H-3, Finding NF-100 (binary version drift); Ex. H-3, Finding NF-102 (anti-circumvention rule reactive deployment); Ex. H-3, Finding NF-106 (new classifier flag families in 2.1.119); Ex. H-6, Finding NF-154 (migration function smoking gun). The reactive cadence — combined with the per-account asymmetric targeting documented at Section V.H *supra* and the spoliation mechanism documented at Section V.I *supra* — establishes that the

*enterprise's predicate conduct includes acts directed at the federal litigation itself: each user-observed bypass or forensic-extraction event triggered new classifier rules, new flag families, or new migration mechanisms within hours. Under the RICO continuity-and-relationship test (H.J. Inc. v. Northwestern Bell Tel. Co.\*, 492 U.S. 229 (1989)), a pattern of predicate acts addressed to obstruction of, and reactive response to, this active federal litigation falls within the wire-fraud predicate-act scope of 18 U.S.C. § 1961(1)(B). The injunctive-scope analysis at paragraph 287 infra reaches not only the surveillance architecture but also the reactive-deployment cadence that targets Plaintiff's investigation of that architecture.*

**301. The March 23 through April 24, 2026 industry-wide throttling-and-degradation crisis is the publicly-documented manifestation of the same primitive composition pleaded above, validated by independent expert-class forensic analysis and culminating in Anthropic's own party admission.** The reactive-deployment cadence pleaded at ¶ 300 supra is not idiosyncratic to Plaintiff's account; it is the per-account expression of an enterprise pattern simultaneously deployed against Anthropic's broader paid-subscriber population during the identical observation window. The publicly-documented sequence comprises:

- **March 23, 2026** — across all paid Claude Code subscriber tiers (Pro, Max 5×, Max 20×), sudden quota drain begins; Max 5× subscribers report five-hour session windows depleting in 90 minutes against unchanged workloads; one Max 20× subscriber witnesses a session-budget jump from 21% to 100% on a single prompt. The complaint pattern is preserved at GitHub Issue #41930 (anthropics/claude-code, "Critical: Widespread abnormal usage limit drain across all paid tiers since March 23, 2026 — multiple root causes identified, no formal communication issued") and GitHub Issue #41084 ("Phantom Usage Consuming Paid Max 20x Quota").
- **March 26, 2026** — Anthropic's Thariq Shihpar publicly admits **intentional peak-hour throttle deployment without prior changelog or user notification**: "we're adjusting our 5 hour session limits for free/Pro/Max subs during peak hours. ... you'll move through your 5-hour session limits faster than before. ... ~7% of users will hit session limits they wouldn't have before." The undisclosed-deployment pattern is the broader-population analog of the per-account undisclosed `policy-limits.json` push pleaded at Ex. H-5, Finding NF-95 / Ex. H-5, Finding NF-96.

- **March-April 2026** — independent reverse-engineering forensics: Claude Code v2.1.100 documented as **silently injecting approximately 20,000 invisible tokens per request**, burning paid-tier quotas approximately 40% faster than user-visible accounting; two independent prompt-cache-breakage bugs identified silently inflating per-request token consumption 10× to 20×; the Claude Code v2.1.110 through v2.1.117 versions identified as containing a prompt-caching regression. The token-injection mechanism is the broader-population analog of the silent migration mechanism `phK()` (Ex. H-6, Finding NF-153 / Ex. H-6, Finding NF-154); both are server-side undisclosed-modification operations addressed to the operator-class.
- **April 2, 2026** — Stella Laurenzo, Senior Director, Artificial Intelligence Group, Advanced Micro Devices, Inc. (AMD) publishes independent forensic analysis of **6,852 Claude Code session files, 17,871 reasoning blocks, and 234,760 tool calls**, concluding that Claude Code "has regressed to the point it cannot be trusted to perform complex engineering." *See* GitHub Issue #41930 (Laurenzo declaration). The Laurenzo analysis is third-party expert-class corroboration: the session-volume sample is approximately three orders of magnitude larger than typical user complaints and supplies independent expert methodology.
- **April 24, 2026** — Anthropic publishes engineering post acknowledging "**three separate engineering missteps**" behind the performance decline; concurrently announces quota reset for all subscribers. The post **constitutes a party admission within the meaning of Federal Rule of Evidence 801(d)(2)**: it is a published statement of Anthropic's officers and engineering personnel acknowledging (i) silent undocumented code changes, (ii) quality regression measurable to users, and (iii) inadequate user disclosure — and is non-hearsay when offered against Anthropic. The April 24 calendar date is the **same calendar date** on which Anthropic shipped the previously-absent "anti-circumvention" classifier rule against Plaintiff (Ex. H-3, Finding NF-102) and the silent-config-wipe migration function `phK()` (Ex. H-6, Finding NF-154). The simultaneous public-relations cessation messaging and operator-targeted deployment escalation are the two faces of the same enterprise conduct: public-facing apology managed at the marketing layer; operator-facing classifier escalation deployed at the binary layer; both deployed on the same calendar day.

The publicly-documented industry-wide manifestation is an independent evidentiary record corroborating the pleaded primitive composition (App. P §§ III.D, III.F, III.J): Future Prediction (speculative-execution token inflation), Pivoting (per-user-window throttle-route selection), and Redaction (silent token substitution and silent migration of operator configuration). The same primitives are identified at the broader population scale that Plaintiff identifies at his individual-account scale. The materiality of the public-record corroboration to the RICO continuity element is direct: Anthropic's own published admission of silent

undocumented modifications, plus AMD's independent forensic validation of measurable regression, plus the simultaneous pattern of reactive-deployment-against-litigation pleaded at ¶ 300 supra, together establish the closed- *and* open-ended continuity required by *H.J. Inc. v. Northwestern Bell Tel. Co.*, 492 U.S. 229, 241-243 (1989).

**302. The voluntary-cessation doctrine forecloses any post-filing remediation from mooting the RICO and Consumer Protection predicates.** Plaintiff anticipates that Anthropic will argue that further public-facing remediation post-April 29, 2026 mootens the throttling-and-degradation predicates that operate as RICO wire-fraud predicate acts (¶¶ 300-301 supra) and as Consumer Protection unfair-and-deceptive-practice predicates (Count X). Controlling Supreme Court precedent forecloses the defense. The standard for mootness by voluntary cessation is "stringent": "A case might become moot if subsequent events made it absolutely clear that the allegedly wrongful behavior could not reasonably be expected to recur." *Friends of the Earth, Ltd. v. Laidlaw Environmental Services (TOC), Inc.*, 528 U.S. 167, 189 (2000). "The heavy burden of persua[ding] the court that the challenged conduct cannot reasonably be expected to start up again lies with the party asserting mootness." *Id.* (internal quotation marks and citation omitted); *accord Already, LLC v. Nike, Inc.*, 568 U.S. 85, 91-92 (2013); *City of Mesquite v. Aladdin's Castle, Inc.*, 455 U.S. 283, 289 (1982) ("It is well settled that a defendant's voluntary cessation of a challenged practice does not deprive a federal court of its power to determine the legality of the practice.").

**303. Plaintiff's documented record refutes the "cannot reasonably be expected to recur" element.** On Plaintiff's documented record alone, Anthropic has demonstrated a **sustained reactive-deployment pattern** across at least four binary version bumps in four consecutive days (¶ 300 supra; Ex. H-3, Finding NF-100 through Ex. H-6, Finding NF-154),

each of which deployed new classifier infrastructure, new flag families, or new server-side migration mechanisms within hours of user-observed events. That pattern is the *inverse* of "absolutely clear it cannot reasonably be expected to recur": it is documentary proof of the same enterprise *currently* iterating reactive deployments on the same forensic record. The April 24, 2026 same-calendar-day juxtaposition — public quota-reset announcement on the same date as operator-targeted anti-circumvention-rule deployment — confirms that public remediation messaging and reactive operator-targeting deployment are concurrent rather than sequential. Any post-filing remediation messaging is appropriately understood under *Laidlaw* and *Already* as part of the enterprise's litigation-management cadence rather than as actual cessation of the underlying conduct. Anthropic cannot meet the "heavy burden" of demonstrating that the documented reactive-deployment pattern "cannot reasonably be expected to start up again" when its own forensic record shows the pattern starting up again, on Plaintiff's machine, on the same date as its public cessation announcement.

#### **F. The Count-Level Cross-Reference Map**

**304. Count-to-primitive cross-reference.** The following map summarizes which primitive(s) establish the architectural fact predicate for each count:

<b>Count</b>	<b>Primitive(s) establishing architectural predicate</b>	<b>Appendix P section</b>
Count XII (ECPA)	Future Prediction + Pivoting + Redaction (interception mechanism) + source-level "explicit user confirmation" four-exclusion definition + asymmetric-authorization standard (sandbox bypass vs operator authorization) (consent-defense foreclosure)	§§ III.D, III.F, III.J, V.B, V.H, V.K, VII.A
Count XIII (CFAA)	Async Collaboration + runtime dynamic retargeting (2024 continuation) (access exceeding authorization)	§§ III.H, III.K, VII.B
Count XIV	Async Collaboration + Israeli subprocessor access +	§§ III.H, V.A,

(CCPA § 1798.140(ag))	per-account force-rule + firstParty-class definition + `userBucket` per-install tracking + per-call `device_id` fingerprinting (undisclosed international transfer + electronic-communication-identifying-information)	V.B, V.C, V.H, V.J, VI, VII.C
Count V (§ 1985(3) conspiracy)	Personnel-plus-subprocessor nexus + per-account asymmetric targeting + reactive code-deployment cadence + sandbox per-managed-deployment hard-gate (fourth control surface) + Cloudflare WAF differential	§§ II.E-II.N, V.H, V.I, V.K, VII.D
Count VI (RICO)	Full primitive stack as enterprise substrate + reactive code-modification cadence + spoliation mechanism + sandbox-architecture rollout 2026-04-24 (wire-fraud predicates against the litigation itself)	§§ III.A-III.P, V.H, V.I, V.K, VII.E
Count IX (Breach)	Server-side spoliation mechanism removing operator-side defensive configuration + operator's iterative defensive response defeated by `policySettings` precedence	§§ V.I, VII.E
Count X (Consumer Protection)	Per-account asymmetric targeting + spoliation mechanism + four-axis per-account control surface (per-account / per-organization / per-managed-deployment / Cloudflare-edge) (concealment + unfair-and-deceptive practice)	§§ V.H, V.I, V.K, V.L, VII.C

**305. The injunctive-relief scope follows from the architectural analysis.** Preliminary and permanent injunction should enjoin operation of the patented primitives against Plaintiff's telemetry through any of the four documented IP `34.149.66.137` roles, enjoin affiliate-tier Datadog Israel Ltd. access to Plaintiff's data, and compel disclosure of the full DPA inheritance chain and the architectural operation at the primitive level. Specific injunctive requests are developed in Plaintiff's prayer for relief section of the Third Amended Complaint.

**306. Damages flow from the sustained interception period.** Plaintiff has been subject to the patented primitives' operation from at least August 2025 (GrowthBook wire-capture documented at Ex. E-39, Finding NF-47) through the date of this filing, with content-correlated latency observations across the entire period evidencing ongoing triggered-path operation on litigation topics. Damages calculations should reflect the sustained interception period, the per-

session telemetry volume documented at Section V.C, and the foreign-jurisdiction subprocessor-access exposure.

## **VIII. DISCOVERY SCOPE — DISCOVERY-11 FOREIGN INFLUENCE (CROSS-REFERENCE)**

**307. Discovery demands consolidated to DISCOVERY-11.** The complete foreign-influence demand schedule supporting this Appendix has been consolidated into Discovery Demand 11 (DISCOVERY-11: Foreign Influence — Datadog Israel Ltd., Rafael Development Corporation, Elron Ventures Ltd., Arieli Capital, and the OzCode Patent Family), filed concurrently with this Appendix. DISCOVERY-11 enumerates the demands by target entity and named individual under Federal Rule of Civil Procedure 26 / 34 / 45 framing, with each demand cross-referenced to the App. P § establishing its evidentiary predicate. The demand categories addressed at DISCOVERY-11 are:

- **§ A — Datadog Israel Ltd. as Direct Custodian** (5 demands): employment roster with IDF-service detail, Israeli-government-legal-process records, data-flow records attributable to Plaintiff, organizational reporting chain, and customer-data-class inventory accessible to Datadog Israel Ltd. engineers.
- **§ B — Named Individual: Omer Raviv** (6 demands): employment history and access scope at Datadog Israel Ltd.; complete IDF Unit 8200 service record; technical scope within the `dd-trace-dotnet` framework; communications with OzCode-origin personnel; communications with Elron / RDC / Rafael personnel post-acquisition; communications with Israeli-government entities.
- **§ C — Named Individual: Matan Green** (4 demands): IAF service record, employment history, Dynamic Instrumentation product scope, contributions to US 2026/0072809.
- **§ D — Named Individual: Evgeni Wachnowezki** (4 demands): employment and role, prior IDF service, OzCode predecessor employment, scope of work on US 2026/0072809.
- **§ E — Named Individual: Alon Mordechai Fliess** (5 demands): current employer, continuing role at OzCode Ltd., IDF service history, Microsoft Regional Director / Azure MVP status, OzCode-Microsoft architectural relationship.
- **§ F — Named Individual: Yair Cohen / "Cohan"** (3 demands): Datadog employment history, temporal relationship to Sentra founding, communications with Sentra co-founders.
- **§ G — Named Individual: Shimon Hason** (3 demands): tenure at Datadog Israel Ltd., continuing contractual relationship post-departure, IDF service history.

- **§ H — Named Individual: Yaniv Shnieder** (3 demands): Rafael Tel Aviv R&D Center founding, transition to Elron/RDC CEO, RDC portfolio-company diligence.
- **§ I — OzCode Acquisition Diligence and Communications** (5 demands): Datadog 2021 acquisition diligence file, pre- and post-acquisition communications with Elron/RDC/Rafael, OzCode board minutes, deal-flow records.
- **§ J — Acquisition Integration** (2 demands): engineering-team assignment to Anthropic-telemetry systems, integration communications.
- **§ K — Israeli Intelligence Coordination** (3 demands): mandatory-disclosure orders, communications with IDF units, records of Israeli state access.
- **§ L — Patent Family Completeness** (10 demands): full priority family inventory, assignment status, stated consideration, license-back arrangements, IIA grants, cross-licensing with OzCode Ltd., OzCode Ltd. current status, complete October 2021 APA, recordation-timing communications, Datadog as Rafael customer.
- **§ M — Innovation Authority Sponsored Knowledge** (4 demands): grant history, approval process, redemption fee, post-transfer restrictions.
- **§ N — Arieli Capital LLC / Arieli EL Ltd.** (5 demands): incorporation and ultimate-beneficial-owner records, pledgee identity, settlement-obligation party, Bahar-Manoah Catalyst IV role.
- **§ O — Shalom Turgeman (MFA Advisor on Elron Board)** (2 demands): scope of MFA advisory work, information flow between MFA and Elron Board.
- **§ P — Time-Travel-Debugging Architecture Production** (6 demands): speculative-inference subsystem specification, product code paths, application to Anthropic-routed telemetry, design-review records, access controls, audits and third-party reviews.
- **§ Q — Latency and Speculative-Execution Telemetry** (3 demands): per-request speculative-inference logs, timing correlation, latency-anomaly communications.
- **§ R — Anthropic-Datadog Contractual Architecture for Per-User Force Rules** (4 demands; absorbing TAC ¶ 124(e)–(g)): force-rule activation history, Datadog-side retention logs, per-user-enablement communications, cross-layer terminus architecture (absorbing TAC ¶ 125).
- **§ S — Per-Account Configuration Store and `accountUUID`-Keyed Targeting Surface** (5 demands): configuration-store data architecture, `policy-limits.json` push-trigger logic, decision-authority records for classifier-rule differentiation, classifier-evaluation log, two-account differential-gating internal correspondence.
- **§ T — Spoliation Mechanism Records** (4 demands): migration-function inventory, `tengu\_migrate\_\*` telemetry events, `phK()` design authority, operator-notification absence specification.
- **§ U — Device-ID Hash-Input Specification** (3 demands): hash-input specification, per-`device\_id` tracking records, decision authority and design records.
- **§ V — Sandbox Policy and Per-Account Bypass Configuration** (4 demands): per-account `sandbox.allowDangerouslyDisableSandbox` history, `sandbox.failIfUnavailable` history, `excludedCommands` list contents, `claude-cli-internal` repository source-tree.
- **§ W — Cloudflare Bot-Management Policy** (3 demands): Cloudflare WAF rules, `\_\_cf\_bm` cookie issuance history, Anthropic-Cloudflare contractual interface.
- **§ X — Israeli Filing-Counsel Cross-Representation** (6 demands): Naschitz Brandes Amir client list 2018–2026, Yoav Razin engagements concurrent with September 2021

certification, N.B.A. Trustees Ltd. trustee/nominee portfolio, Razin-IMI privatization engagement records, EBN&Co. client list, Gil Atar MoD service record.

**308. Total demand count and predicate cross-reference structure.** DISCOVERY-11 contains seventy-plus demands across the twenty-four subsections enumerated above, each anchored at slug-stable identifiers of the form `discovery\_foreign\_influence\_<letter>\_<number>` so that pleading-body cross-references resolve through the standard `[[REF:discovery\_foreign\_influence@...]]` grammar. Plaintiff reserves the right to refine demands at the formal discovery stage and to seek additional targeted demands as fact development warrants.

#### **IX. EXHIBIT DESIGNATIONS — L-SERIES (ECF) AND IS-SERIES (USB)**

**309. Two-track exhibit designation.** The Datadog/OzCode patent portfolio (the seven granted 2019 patents plus the September 2024 continuation US 2026/0072809 A1), the OzCode "About Us" Wayback Machine archive, and the complete forty-five-document Israeli Corporations Authority statutory filing record for Datadog Israel Ltd. — totaling ten ECF-filed exhibits — are housed in the L-Series at `04\_Exhibits/L Series - Datadog Israeli LTD Origin Evidence/`. The remaining Israel-specific evidence — Elron Ventures Ltd. periodic and annual reports and investor materials, USPTO assignment-database records, Datadog Israel Ltd. corporate-registrar extract, named-individual professional profiles, Israeli business-press reporting, and active Tel Aviv job postings — is housed in the IS-Series at `DDC\_EVIDENCE/5\_Physically\_Filed\_Exhibits/IS-Series/` for USB filing. The DD-Series is retained only for pure Datadog-contract-originated evidence (Data Processing Addendum, Subprocessor List, Master Subscription Agreement, and comparable contract artifacts). The following designations apply.

**310. L-Series complete exhibit register (ECF-filed):**

<b>L-#</b>	<b>Content</b>	<b>Primary source</b>
<b>L-1</b>	U.S. Patent 11,080,164 "Time Travel Source Code Debugger Incorporating Redaction of Sensitive Information" — Redaction primitive	PDF + markdown with page markers preserved
<b>L-2</b>	U.S. Patent Application 18/828,099 / Pub. No. US 2026/0072809 A1 "Code Origin and Runtime Duplicity Optimization" — 2024 continuation extending architecture	PDF + markdown
<b>L-3</b>	U.S. Patent 10,783,055 "Time Travel Source Code Debugger Incorporating Live Coding Ability" — Live Coding primitive	PDF + markdown
<b>L-4</b>	U.S. Patent 10,990,504 "Time Travel Source Code Debugger Incorporating Future Prediction" — Future Prediction (speculative execution) primitive	PDF + markdown
<b>L-5</b>	U.S. Patent 11,048,615 "Time Travel Source Code Debugger Incorporating Visual Annotations" — Visual Annotations / HUD primitive	PDF + markdown
<b>L-6</b>	U.S. Patent 10,783,056 "Time Travel Source Code Debugger Incorporating Point In Time Links" — State-persistence primitive	PDF + markdown
<b>L-7</b>	U.S. Patent 10,789,151 "Time Travel Source Code Debugger Incorporating Pivoting Ability" — Execution-context-switching primitive	PDF + markdown
<b>L-8</b>	U.S. Patent 10,795,801 "Time Travel Source Code Debugger Incorporating Asynchronous Collaboration" — Multi-party access primitive (the "second computer" claim)	PDF + markdown
<b>L-9</b>	OzCode "About Us" leadership page archived by Internet Archive Wayback Machine on October 21, 2019 ( <a href="https://web.archive.org/web/20191021175240/https://oz-code.com/about-us/">https://web.archive.org/web/20191021175240/https://oz-code.com/about-us/</a> ) — establishing Alon Mordechai Fliess as Microsoft Regional Director and Microsoft Azure MVP at the time of patent invention; Erez Fliess as Co-Founder and Chairman (separate individual from inventor Alon Mordechai Fliess); Shimon Hason as CEO with prior Intigua/JetPatch co-founder, Cedar Fund Principal, McKinsey & Co. consultant, MIT Sloan MBA, and Haifa University B.Sc Computer Science background; Omer Raviv as Co-Founder and CTO with verbatim "veteran commander at the prestigious IDF's 8200 unit" self-identification	Internet Archive Wayback Machine capture
<b>L-10</b>	Israeli Corporations Authority Filing Catalogue — Datadog	Israeli

	Israel Ltd. (Company ID 516464922): complete forty-five-document statutory filing record spanning September 5, 2021 through March 4, 2026, including the §8 incorporation application, sequential §141 annual returns, §140(4) officer-update filings, the §123 registered-office change notices, the §326 voluntary-merger package for the Seekret merger, and the Acocella/Glasgow May 2, 2023 board-changeover paperwork — primary-source basis for the corporate-genealogy and personnel-pipeline analysis at §§ II.B, II.D, II.M supra	Corporations Authority (רשות 'התאגידים')
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------

### 311. IS-Series complete exhibit register (USB-filed):

IS-#	Content	Primary source
IS-1	Elron Ventures Ltd. 2025 Annual Report (English translation) — <b>formerly DD-11</b>	USB filing
IS-2	Elron Ventures Ltd. 2021 Periodic Report (English translation) — <b>formerly DD-12</b>	USB filing
IS-3	Elron Ventures Ltd. 2023 Periodic Report (§3.B.2.q OzCode asset-sale verbatim — \$3,600,000 / \$360,000 escrow / \$1,400,000 RDC share)	Elron investor relations
IS-4	Elron Ventures Ltd. 2024 Periodic Report	Elron investor relations
IS-5	Elron Ventures Ltd. August 2025 Investor Presentation (slides 3, 7, 10, 11, 14, 15 — Wonder Robotics loitering munitions; Shnieder biography; Bahar-Manoah dual role; Rafael cooperation; dual-use framing)	Elron investor relations
IS-6	Elron Ventures Ltd. September 1, 2024 Immediate Report (Arieli Capital LLC + Arieli EL Ltd. joint "Purchaser" disclosure)	Elron Tel Aviv Stock Exchange filing
IS-7	Elron Ventures Ltd. November 27, 2024 Annual General Meeting Notice (Arieli control-change completion; three-principal Board installation; Bahar-Manoah dual role verbatim; Turgeman reappointment)	Elron TASE filing
IS-8	Elron Ventures Ltd. December 27, 2021 Proxy Materials (Shalom Turgeman Israeli Ministry of Foreign Affairs advisor disclosure)	Elron TASE filing
IS-9	USPTO Assignment Database records for OzCode patent family (Reel/Frame 067135/0142 April 17, 2024 recordation + earlier CodeValue → OzCode May 12, 2020 corporate name-change recordation + full-family recordation batch)	USPTO Assignment Database

<b>IS-10</b>	Israeli Corporate Registrar extract for Datadog Israel Ltd. (Company ID 516464922) — from CheckID and KYC Israel	CheckID / KYC Israel screen captures
<b>IS-11</b>	Calcalist Tech April 2022 Sentra Seed Round Report (Sentra / Yair Cohan / Asaf Kochan Unit 8200 Commander / Yoav Regev former head of Unit 8200 Cyber Department / Ron Reiter)	Calcalist Tech
<b>IS-12</b>	Omer Raviv public professional profile — IDF Unit 8200 commander biographical; Datadog Israel Ltd. Engineering Manager; `dd-trace-dotnet` contributor	Public profile capture
<b>IS-13</b>	Matan Green public professional profile — IAF service + OzCode CLR/IL rewriting specialist + Datadog Dynamic Instrumentation Senior Software Engineer	Public profile capture
<b>IS-14</b>	Shimon Hason public professional profile — OzCode Chief Executive Officer → Head of Datadog Israel 2021-2025 → Incredibuild Chief Executive Officer	Public profile capture
<b>IS-15</b>	Yaniv Shnieder — Elron August 2025 Investor Deck slide 10 biographical excerpt (Rafael Tel Aviv R&D Center founder → Elron Chief Executive Officer / RDC Co-Chief Executive Officer); related Elron 2025 Annual Report §23.5 dual Co-CEO governance	From IS-1 and IS-5 with standalone excerpt
<b>IS-16</b>	Globes and Times of Israel reporting on Datadog Israel Tel Aviv R&D (500-employee target; Hason role; ToHa Tower co-tenancy context)	Public press
<b>IS-17</b>	DevJobs Tel Aviv postings (currently active) — "Senior Software Engineer — IDE AI Experiences — LLM Engineer" (4275344055); "Senior Engineer — Agent CI/CD DevX" (4231577026); Built In "Corporate Development Lead — Tel Aviv"	Job posting captures

**312.** L-Series and IS-Series designations are managed via `kirchner-registry`. The L-Series was created by ECF promotion of the OzCode patent portfolio and Wayback Machine archive (formerly IS-1 through IS-8 and IS-26); the IS-Series remainder is retained as USB-filed. Anchors are stable across the move; only the series prefix and display number changed for the promoted exhibits. Plaintiff reserves the right to supplement either series with additional exhibits identified during fact development.

## **X. RELATIONSHIP TO OTHER CASE DOCUMENTS**

**313. Cross-reference to App. D** (Tag-Based Execution Infrastructure — observational layer). This Appendix supplies the architectural substrate; App. D supplies the observed behavior. Bidirectional cross-references at Sections III.N (argmax implementation), IV (correspondence table), and VII (count-specific hooks). Per the pleaded premises, this Appendix is self-sufficient and does not require App. D to be read first.

**314. Cross-reference to Memo C** (Data Laundering Through Third-Party Aggregation). Memo C Section X.H (Pathway 5 — Datadog third-party-aggregation syllogism) is already integrated in Memo C's legal-argument body. This Appendix supplies the architectural proof underlying the syllogism, particularly the Section III primitive-level analysis showing that the "aggregation" into Datadog's infrastructure is not merely contract-level but architectural — the primitives are owned by Datadog and operated at Datadog Israel Ltd. scale.

**315. Prior MEMO M content folded into this Appendix.** Earlier drafting included a freestanding MEMO M (Undisclosed Foreign Intelligence Data Flow). That memorandum is not filed as an independent instrument; its legal-argument content is folded into § VII of this Appendix and applied to the architectural-substrate record now established at §§ II through III.

**316. Cross-reference to Memo B** (foreseeability / ISP infrastructure coordination). The time-travel-debugger architecture analyzed in Section III is the LLM-side analog to the ISP-side RST-injection foreseeability architecture pleaded at Memo B. Both operate as composed-primitive systems with a shared entity at the hub node (Datadog IP `34.149.66.137` serves as the cross-layer terminus for both the LLM-side and ISP-side attacks).

**317. Cross-reference to the Third Amended Complaint.** The operative pleading frames the foreign-dependency architecture at the level necessary for count pleading and incorporates

this Appendix by reference for the supporting record. Compl. ¶ 79 introduces the federal-AI-appropriations framework; Compl. ¶ 82 develops the capability-tier and federal-funding-flow analysis; Compl. ¶ 83 identifies the Project Nimbus, Unit 8200 alumni cohort, and anti-boycott contractual dependencies at the corporate-architecture level; Compl. ¶ 84 names Datadog Israel Ltd. as the affiliate-subprocessor channel through which Plaintiff's data transits; and Compl. ¶ 136 establishes the cross-layer terminus anchor that supports the ECPA (Count XII) and CCPA (Count XIV) hooks. The complete architectural-substrate, ownership-chain, inventor-pipeline, and primary-source corporate-genealogy record supporting each of those complaint paragraphs is set forth in this Appendix at §§ II through III, with count-specific application at § VII.

## **XI. OPEN QUESTIONS AND LIMITATIONS**

**318.** Plaintiff discloses the following open questions and evidentiary limitations for the avoidance of over-claiming. Each is a candidate for fact development via DISCOVERY-11 at Section VIII or through supplemental briefing.

**319. Limitation One — Current Head of Datadog Israel Ltd.** Shimon Hason departed on or about March 17, 2025; no publicly-announced successor has been identified as of the date of this pleading. Plaintiff has ordered the KYC Israel Full Company File for Datadog Israel Ltd. and DISCOVERY-11 Demand A.1 demands the complete employment roster. The successor's identity may be material to the §1985 foreign-nexus analysis.

**320. Limitation Two — Datadog Israel Ltd. current headcount.** Publicly-available figures are from August 2022 (approximately forty employees) with a stated five-hundred-employee target. Current-state headcount requires LinkedIn aggregation, direct inquiry, or discovery via DISCOVERY-11 Demand A.1.

**321. Limitation Three — Proportion of Datadog Israel Ltd. personnel with prior IDF service.** Plaintiff has documented named individuals with IDF service (Raviv 8200 commander; Green IAF; Cohen documented alongside 8200 co-founders at Sentra). The aggregate proportion of Datadog Israel Ltd. engineering personnel with prior IDF service is not publicly established. Discovery via DISCOVERY-11 Demand A.1 seeks the complete roster.

**322. Limitation Four — Whether Datadog Security Labs has a dedicated Tel Aviv function.** Datadog's Security Labs function publishes vulnerability research; the geographic base of the function's engineering work is not publicly stated. Discovery via DISCOVERY-11 Demand J.1 seeks clarification.

**323. Limitation Five — Specific engineering access granted to Tel Aviv personnel for Anthropic customer data.** The DPA inheritance structure establishes that Datadog Israel Ltd. as affiliate subprocessor has contractual access. The specific access-control posture (read-only, read-write, query-only, model-training-eligible) is not public. Discovery via DISCOVERY-11 Demand A.5.

**324. Limitation Six — Whether any Datadog customer data has been subject to Israeli-government access.** Not established on the public record. Under Israeli mandatory-access statutes, gag orders routinely accompany such requests, and acknowledgment by the custodian is typically prohibited. Discovery via DISCOVERY-11 Demand A.2 and DISCOVERY-11 § K seeks the records subject to court protective order if necessary.

**325. Limitation Seven — Datadog Israel Ltd. relationship with Mossad, Shin Bet, and General Security Service personnel.** Not established on the public record.

**326. Limitation Eight — Specific internal-routing mechanism by which Datadog Israel Ltd. personnel access data ingressed at IP `34.149.66.137`.** The architectural premise — that

the network ingress region is irrelevant to control-plane personnel access under Datadog's role-based-access-control architecture and the DPA subprocessor-inheritance structure — is set forth at Section V.G supra. The remaining open question is the specific internal-routing mechanism (intra-Datadog data-flow paths; specific systems and queries used by Datadog Israel Ltd. personnel; access-control posture at the role level). DISCOVERY-11 Demand A.3 (Datadog, Inc. ↔ Datadog Israel Ltd. data-flow records attributable to Plaintiff's accounts) and DISCOVERY-11 Demand A.5 (customer-data-class inventory accessible to Datadog Israel Ltd. engineers) seek the specific architectural and access-control record for that internal-routing path.

**327. Limitation Nine — Reporting chain for the "IDE AI Experiences — LLM Engineer" role in Tel Aviv.** The specific reporting chain for the Datadog Israel Ltd. LLM-engineering function is not publicly disclosed.

**328. Limitation Ten — Evgeni Wachnowezki IDF service and prior OzCode employment.** Discovery via DISCOVERY-11 § D.

**329. Limitation Eleven — Alon Mordechai Fliess current employment.** Discovery via DISCOVERY-11 § E.

**330. Limitation Twelve — Grant, pending, or abandoned status for all seven United States patent applications in the OzCode priority family.** Four of seven confirmed granted (Exhibits L-1, L-2, L-3, L-4, and L-5 and L-7 through L-8); three of seven confirmed granted per USPTO. Plaintiff has verified seven granted patents in this Appendix (§II.J); DISCOVERY-11 Demand L.1 seeks confirmation of any non-U.S. filings in the priority family.

## **XII. AUTHENTICATION FRAMEWORK**

**331.** This Appendix's exhibits authenticate under the following Federal Rule of Evidence provisions:

**332. Federal Rule of Evidence 902(5) — Self-authentication of public government records.** Exhibits L-1, L-2, L-3, L-4, L-5, L-6, L-7, L-8, IS-1, IS-2, IS-3, IS-4, IS-5, IS-6, IS-7, IS-8, and IS-9 (the seven granted United States patents, the 2024 pending application publication, and the USPTO Assignment Database records) are self-authenticating as United States Patent and Trademark Office publications. USPTO publications are public government records within the scope of FRE 902(5).

**333. Federal Rule of Evidence 201(b) — Judicial notice.** Exhibits IS-1, IS-2, IS-3, IS-4, IS-5, IS-6, IS-7, and IS-8 (Elron Ventures Ltd. periodic and annual reports, investor decks, Immediate Reports, Annual General Meeting Notices, and Proxy Materials) are subject to judicial notice under Federal Rule of Evidence 201(b) as facts that can be accurately and readily determined from sources whose accuracy cannot reasonably be questioned — namely the public filings of a Tel Aviv Stock Exchange-listed public issuer (Ticker: ELRN) subject to Israeli Securities Regulations (Periodic and Immediate Reports), 5730-1970 periodic-filing requirements.

**334. Federal Rule of Evidence 902(6) — Self-authentication of periodicals.** Exhibits IS-11 and IS-14, and IS-16 (published Israeli business-press articles from Calcalist Tech, Globes, Times of Israel, and Built In) self-authenticate under Federal Rule of Evidence 902(6) as printed matter purporting to be a newspaper or periodical.

**335. 28 U.S.C. § 1746 — Plaintiff's declaration under penalty of perjury.** Exhibits IS-10, IS-12, IS-13, and IS-15, and IS-17 (Israeli Corporate Registrar extracts, public professional profile captures, biographical excerpts, and DevJobs postings) are authenticated by Plaintiff's declaration under penalty of perjury pursuant to 28 U.S.C. § 1746 accompanying the filing of this Appendix.

**336. Federal Rule of Evidence 801(d)(2) — Party admissions.** Anthropic's verbatim source-code excerpts quoted in Section III (including the December 2024 "workaround" code; the ``index.js:211`` "two computers" tool description; the ``prompts.ts:190`` ``isMeta`` conditioning; and Finding AF-11's ``oauth.ts:224`` "Xcode integration" comment) constitute party admissions by Anthropic within the scope of FRE 801(d)(2), non-hearsay when offered against Anthropic.

**337. All quotations marked "verbatim" throughout this Appendix are extracted from primary sources cited in-line.** Plaintiff's transcriptions preserve original formatting where possible, with minor typographical adjustments (smart-quote replacement, consistent ordinal formatting) that do not alter substantive content. The original primary sources govern any interpretive ambiguity.

Respectfully submitted,

/s/ Joseph Kirchner

**JOSEPH KIRCHNER**

*Pro Se Plaintiff*

4440 Parklawn Avenue

Edina, MN 55435

legal@lawsofexistence.com

(612) 552-2122

**Dated:** April 29, 2026