COMPUTER FORENSICS LAB

London digital forensics & e-discovery · Established 2007

Cloud Evidence Is Not Just Files

Versions, Logs, Permissions, Sharing Links, Deleted Objects and Admin Records: the Seven Layers Around Every Cloud Document

Ask for "the files" from a cloud platform and you will get them: current versions, neatly exported, and stripped of almost everything that made them evidence. Around every cloud document sits an invisible estate: version and revision histories recording how it changed; audit events recording who touched it; permission structures recording who could; sharing links recording who else was invited; deletion records outliving the deletions; configuration and admin trails recording how the whole environment was governed. This guide: the capstone of the cloud block: names the seven layers, shows what each proves that files cannot, sets out how to scope, request and collect them across any platform, and works through the disputes where the layers, not the files, carried the case. It is the one-guide answer to the request this series most often corrects: "just get us the documents".

⏱ Estimated reading time: 26 min read

§ ABOUT THE AUTHOR

PREPARED BY COMPUTER FORENSICS LAB E-DISCOVERY TEAM

Computer Forensics Lab is a London-based digital forensics and electronic disclosure practice established in 2007. Its cloud casework across Microsoft 365, Google Workspace and the wider SaaS estate is built on the layered method this guide distils: collecting the records around documents as deliberately as the documents themselves, and correlating them into the chronologies, attribution findings and conduct analyses that decide disputes. Its eDiscovery arm, **e-discovery.uk**, carries the layers intact into review.

ESTABLISHED 2007 · LONDON | ISO 17025-ALIGNED PROCEDURES | CLOUD & SAAS FORENSICS | LAYERED EVIDENCE METHOD | CPR PART 35 EXPERT REPORTS | FULL CHAIN-OF-CUSTODY DOCUMENTATION

§ CONTENTS

In this guide

§ 0 1 · ORIENTATION

Executive summary

THE HEADLINE POINT: IN CLOUD PLATFORMS THE FILE IS ONE LAYER OF SEVEN, AND THE OTHER SIX ARE USUALLY THE CASE

Cloud platforms are record-keeping machines that happen to store files. Around every document sit: **(1) versions and revisions**: the drafting history, per-save or per-edit, with authors and timestamps; **(2) activity and audit events**: access, edits, downloads, shares, deletions as dated acts by accounts; **(3) permissions**: the computed audience: who could, as at any date the audit reach supports; **(4) sharing links and external access**: the invitations beyond the permission structure, with creation and use records; **(5) deletion and recycle records**: staged afterlives with deletion dates and deleting accounts, outliving content in evidential value; **(6) configuration and settings histories**: retention rules, history toggles, sharing policies: the governance whose change dates convert explanation into conduct; and **(7) admin and operations records**: role use, account lifecycle, hold and export operations: the environment's own management diary. The files answer "what does it say"; the layers answer everything else litigation asks: who wrote which sentence and when (1), who saw it and what they did (2-4), what happened to it and at whose hand (5), and whether the estate was governed or gamed (6-7). The practice consequences are three. **Scope in layers:** every cloud disclosure request and specification names the layers wanted per document population, because "the files" delivers layer zero by default. **Preserve in layers:** holds stop content clocks; the audit, configuration and operations layers have their own shorter windows (guide 57's shredder) and need their own week-one exports. **Collect in layers:** routes differ per layer (compliance export for content, API for structures, admin exports for events and settings), and the correlation across them: §7: is where the findings live. This guide is deliberately platform-neutral: the layer model is the transferable method; guides 51-59 supplied the Microsoft and Google mechanics, and the SaaS guides ahead will keep supplying new mechanics for the same seven questions.

- **"Just the files" is an amputation:** the request this guide retires: every specification names its layers.
- **Layers have their own clocks:** audit windows, recycle stages and settings histories expire on schedules unrelated to the content holds: the week-one exports cover the layers, not just the documents.
- **The layers prove in combinations:** a version plus its edit event plus the session context is a finding; any one alone is a fragment: correlation is the method, not a garnish.
- **Governance layers police both sides:** settings and operations records evidence the opponent's conduct and discipline your own: read yours before asserting anything.
- **The model is portable:** new platform, same seven questions: the fastest competent start on any SaaS estate this series knows.

§ 0 2 · FIRST PRINCIPLES

The problem in plain English: the estate around the document

On paper, a document was just a document: what surrounded it: who read it, who could have, what earlier drafts said: lived in memories and, occasionally, file-room registers. Cloud platforms inverted that world without telling the profession: now the surroundings are recorded more durably than memory ever managed, by machinery no witness controls: and the habits of paper disclosure ("send us the documents") persist into estates where the documents are the least revealing layer present. The result is a systematic under-asking: productions technically complete and evidentially hollow, disputes litigated on files while the answers sit unrequested in the layers around them.

The fix is a habit of mind as much as a procedure: see every cloud document as the centre of a small estate, and every request, preservation step and collection plan as a decision about how much of the estate to take. This guide gives the estate a fixed map: seven layers, platform-independent: so the decision is made deliberately every time, in scoping meetings, DRD entries, specifications and requests, rather than discovered as a gap at trial.

§ 0 3 · THE MAP

The seven layers, named

LAYER	WHAT IT IS · WHERE THE BLOCK MET IT
1 · Versions / revisions	Per-save series (SharePoint/OneDrive, guide 54-55) or per-edit streams (Google Docs, guide 58): the drafting record, collected as series via structural routes.
2 · Activity / audit events	Access, edit, download, share, delete as dated acts by accounts: guide 57's unified log; guide 58's admin reports: the chronology spine.
3 · Permissions	The computed possibility map with inheritance and group resolution: guide 55's discipline: reconstructable as-at-date within audit reach.
4 · Sharing links / external access	Link types, audiences, expiries, first-access events: guide 54's leak machinery: the audience beyond the org chart.
5 · Deletion / recycle records	Staged bins, deletion metadata, restore events: guides 53-55's marches: the acts that outlive their objects.
6 · Configuration / settings histories	Retention rules, history toggles, sharing policies, version limits, with change dates: guides 56-59's conduct seams.
7 · Admin / operations records	Role use, account lifecycle, hold/search/export operations: guides 51, 57 and 59's governance diary.

§ 0 4 · THE PROOFS

What each layer proves that files cannot

- **Layer 1 answers authorship and timing at sentence grain:** who inserted the clause, when the figure changed, whether the "final" predates the signature: the questions files answer only with their weakest field, the last-modified date.
- **Layer 2 answers knowledge and conduct:** the download before resignation, the read before the denial, the deletion after the letter: acts, dated, by accounts: the material of every chronology this block has built.
- **Layer 3 answers exclusivity and opportunity:** "only she could" tested by computation; the pool named before attribution begins: the fairness layer.
- **Layer 4 answers audience:** who outside the structure was invited, when, and whether they came: the leak inquiry's backbone and the confidentiality claim's test.
- **Layer 5 answers spoliation:** what went, when, at whose hand, against which duty date: with the recycle metadata often surviving the content it describes.
- **Layer 6 answers "routine":** the retention shortened, the history toggled, the policy rescoped: dated against the dispute, converting settings into conduct or vindicating them as genuine housekeeping.
- **Layer 7 answers process:** who held the keys, what they did with them, and whether the disclosure exercise itself was run straight: the layer that audits the audit.

§ 0 5 · ASKING PROPERLY

Scoping and requesting the layers, platform-neutrally

- **The layer schedule:** every cloud specification carries a per-population layer table: which document sets need versions as series, which need events, which need the possibility map: the scoping dial (guide 37) applied vertically as well as horizontally.
- **Platform-neutral request language:** ask for the layer, let the platform name the mechanism: "version or revision histories as series, with authorship and timestamps", "activity records for [population] including access, download, sharing and deletion events", "permission and sharing states as structured data, as at [dates]", "retention and settings histories with change dates", "administrative and preservation-operations records": wording that works against any estate and forecloses the "our system calls it something else" deflection.
- **Preservation names the layers too:** the week-one letter and instruction preserve events, settings histories and operations records by name: content holds alone leave the shorter clocks running.
- **Proportionality by layer:** layers cost differently: events and settings exports are cheap; version series are targeted; full possibility maps are occasional: the schedule matches depth to what each population must prove.
- **Reciprocity offered:** your own layer disclosure volunteered where sought: the cooperation record that makes the requests above hard to resist.

§ 0 6 · GETTING THE LAYERS OUT

Collecting the layers: routes, fidelity, integrity

- **Routes per layer:** compliance exports carry content and some metadata; API routes carry series and structures (layers 1, 3, 4); admin consoles export events, settings and operations (layers 2, 6, 7); recycle enumerations come from the platforms' own stores (layer 5): the routing table of guide 59, drawn vertically.
- **Fidelity per layer stated:** what each export preserves and flattens, layer by layer: the honesty discipline, applied to the whole estate.
- **Integrity as everywhere:** documented queries, complete result handling, hashes, manifests: layers are challengeable evidence and answer with the same paperwork as content.
- **Time-alignment at collection:** layers timestamp in their own semantics (event time, ingestion time, platform time zones): captured and normalised deliberately, because §7's correlations live or die on the alignment.
- **Interested operators, same rule:** layer exports by perimeter custodians get the guide 59 verification patterns: the diary's curator problem does not improve when the diary is about them.

§ 0 7 · THE METHOD

Correlation: turning layers into findings

- **The join, generalised:** version (what changed) + event (who acted, when) + session (from where) + device (at whose hands) + conduct (why it coheres): the five-element chain this series has built case by case, stated once as the method.
- **Chronologies assemble across layers:** the deletion spree (5+2) against the duty (correspondence) under the settings change (6) by the admin (7): findings are sentences whose words come from different layers.
- **Exculpation uses the same grammar:** the absent event with coverage proven, the pool that was never one person, the housekeeping with a genuine signature: guide 57's Example 3 and guide 55's Example 3, as method.
- **Alternatives canvassed structurally:** shared credentials, engine-initiated activity, migration artefacts: each has a layer signature, and the report tests them by signature rather than assertion.
- **Findings sized to grain:** each layer's resolution (save-grain, event-grain, licence-bound reach) caps what the correlation may claim: the honesty that survives the opposing expert.

§ 0 8 · THE WIDER MAP

Source architecture: where else the evidence lives

Per this series' source-architecture discipline: the seven layers themselves have counterparts beyond the platform. Layer content echoes in device mirrors and caches; events echo in SIEM ingestion and counterpart tenants' logs; permissions and sharing echo in recipients' own records; deletions echo in backup platforms' point-in-time sets; settings histories echo in change tickets and IT correspondence; operations echo in provider notifications and export artefacts. The architecture walk of every previous guide is, in this model, the horizontal move: same layer, different holder: and the layer model is the vertical one: same document, different record. Competent cloud evidence work moves in both directions, and the matrix below crosses them.

LAYER	PLATFORM (PRIMARY)	DEVICE SIDE	SIEM / EXTENDED LOGS	COUNTERPARTS / RECIPIENTS	BACKUP LAYER	DELETED / RECOVERABLE
1 · Versions	Y: series via structural route	Shadow copies, caches	n/a	As-shared generations	Generational sets	Pruning/trimming noted
2 · Events	Y, window-bound	Local artefacts corroborate	Y: longer retention	Their event layer	Backup's own logs	Windows absolute; export first
3 · Permissions	Now-state + as-at via events	n/a	Change events	Membership context	Point-in-time states	Reconstructable within reach
4 · Sharing	States + use events	Browser/ session traces	Access records	The recipients themselves	States in sets	Varies
5 · Deletions	Recycle metadata, window-bound	Mirror divergence	Deletion events	n/a	Bracketing sets	Acts outlive objects
6 · Settings	Current + change history	n/a	Config-change events	n/a	Historic states	The conduct layer
7 · Operations	Y: the platform's own log	Admin workstation artefacts	Where ingested	n/a	n/a	Window-bound; tickets corroborate

Fallback strategy in one line: move horizontally before conceding vertically: the layer the platform lost usually persists with a device, a counterpart, a SIEM or a backup: and the operations layer tells you which to try first.

§ 0 9 · IN THE WILD

Worked examples

EXAMPLE 1 · THE PRODUCTION THAT WAS TECHNICALLY COMPLETE

A warranty claim's defendant produced every document requested: thousands of files, immaculate, current-state. The claimant's expert asked one scoping question: which layers had been collected: and the answer (none beyond content) reshaped the case. The layer requests that followed produced: version series showing the disclosure schedule's key figures edited after the SPA's data-room cutoff (layer 1); download events showing the CFO pulling the originals the same week (2); and a version-limit setting changed to five, from five hundred, the day before collection (6): which had silently truncated most series and was, once dated, indefensible. The files had said nothing; the layers said everything; and the settlement followed the second production, not the first.

EXAMPLE 2 · THE LEAK WITH A FOUR-LAYER SIGNATURE

A confidential tender bundle reached a rival. The files proved only that it existed. The layers convicted: sharing analysis found an external link created on the bundle's folder (4); its first-access event resolved to the evening of creation (2); the possibility map confirmed the creating account was one of three with folder rights (3); and the creating account's session context plus the imaged laptop's browser artefacts supplied the join to a person: who had, the operations layer added (7), asked IT to purge his account "for GDPR reasons" the following week: a request logged, dated and declined. Four layers plus a device: the standard this series calls proof: and a defence that never got past the first interview.

EXAMPLE 3 · THE LAYERS THAT CLEARED THE ESTATE

Liquidators alleged a director had systematically stripped a company's cloud estate before insolvency: folders were sparse and suspicion was cheap. The layer review told the other story: deletion records (5) showed volumes within the account's multi-year norms; the settings history (6) showed retention rules untouched for four years; the events layer (2) showed no download clusters, no unusual sessions; and the operations record (7) showed no holds released, no exports run, no admin anomalies: while the backup layer's quarterly sets matched the estate's decline to ordinary business contraction, quarter by quarter. The report's conclusion was the rare, valuable negative: the estate was thin because the business was, and every layer said so consistently. The allegation was withdrawn before it was pleaded.

§ 10 · WHERE IT GOES WRONG

Common mistakes and technical limitations

Common mistakes

- **Content-only requests:** Example 1's first production, invited by the request that shaped it.
- **Content-only preservation:** holds placed, layer windows left running: the audit export missed again.
- **Layers collected but never correlated:** exports shelved separately, the joins nobody built: findings left as fragments.
- **Grain overrun:** save-grain layers claimed at keystroke grain; event windows claimed past their reach.
- **Settings assumed innocent, or guilty:** layer 6 read without dates: housekeeping and conduct confused in both directions.
- **Your own layers unread:** assertions ("routine deletion", "nobody accessed it") made before checking the records that will check them.
- **Platform-specific requests:** wording that names one vendor's mechanism and licenses the "we use a different system" evasion.
- **Time semantics ignored:** cross-layer chronologies built on unaligned clocks: guide 252's subject, previewed.

Technical limitations

- **Every layer is window-bound differently:** the layer schedule carries a clock column, and completeness statements respect it.
- **Reach varies by licence and configuration:** the entitlement facts precede every promise, per guides 57-58.
- **Layers can conflict:** event streams, version series and caches occasionally disagree: the discrepancy is investigated and stated, not smoothed (guide 258 ahead).
- **Grain is a property, not a defect:** each layer proves at its resolution; the report claims accordingly.
- **The model maps platforms, not people:** the join to a human remains the correlation's final, non-optional element.

§ 11 · INTERROGATORIES & DRAFTING AIDS

Questions to ask · Suggested wording

Ask your client

- For each platform in the estate: which of the seven layers exist, at what windows: and are the week-one layer exports run?
- For the documents that matter: what would layers 1-2 show, and have we looked before pleading?
- What do our own settings and operations records say about the period: read before assertions go out?

Ask your opponent

- Please state, per platform, which layers your collection included, and produce the layer schedule applied.
- For [population]: please produce version/revision histories as series; activity records (access, download, sharing, deletion); permission and sharing states as at [dates]; and retention/settings histories with change dates for [scope].
- Please produce preservation- and export-operations records for the disclosure exercise itself.

Ask your eDiscovery / forensic provider

- Draw the layer schedule and its clock column: what expires when, and what must export this week?
- For the pleaded findings: which layer combinations support them, at what grain, with what alternatives tested?
- Where layers conflict: what does the discrepancy itself indicate?

SUGGESTED WORDING · LAYER SCHEDULE CLAUSE FOR THE DISCLOSURE SPECIFICATION

> "Cloud-platform collection shall proceed by layers per the Schedule: for each document population listed, the Schedule states whether collection includes (1) version or revision histories as series with authorship and timestamps; (2) activity and audit records including access, download, sharing and deletion events; (3) permission structures as data, current and as at the dates listed; (4) sharing links and external-access records with creation and use events; (5) deletion and recycle records with deletion metadata; (6) retention, settings and policy histories with change dates; and (7) administrative and preservation-operations records. Each layer shall be exported by a route preserving its structure, with queries documented, results hashed and fidelity stated, and the parties shall exchange layer schedules with their productions."

§ 12 · QUICK CONTROL

Checklist and red flags · When to involve a digital forensic expert

The layers checklist

- ☐ Layer schedule drawn per platform and population, clock column included
- ☐ Week-one exports cover the expiring layers (events, operations), not just content holds
- ☐ Requests and specifications name layers in platform-neutral language
- ☐ Structural routes assigned to layers 1, 3 and 4 where the issues need series and states
- ☐ Your own layers 5-7 read before any assertion about deletion, access or routine
- ☐ Correlations built and documented: version+event+session+device+conduct
- ☐ Grain and reach stated per layer in every finding
- ☐ Conflicts between layers investigated, not smoothed
- ☐ Opponent's layer schedule requested with their production
- ☐ Time semantics normalised before cross-layer chronologies are drawn

Red flags

- A production with no layer schedule and no route statement.
- Settings changes (retention, version limits, history toggles) dated inside the dispute: Example 1's truncation.
- "Nobody accessed it" with no events layer produced.
- Deletion explanations unaccompanied by layer 5-6 paperwork.
- Findings pleaded from single layers at impossible grain.
- Operations records refused in a process dispute.
- An estate suspiciously thin whose layers were never asked (Example 3's inverse: suspicion never tested).

When to involve a digital forensic expert

At scoping, where the layer schedule and its clock column are drawn before anything expires; at collection, where routes preserve each layer's structure and integrity; at correlation, where the five-element joins are built, alternatives tested by signature, and grain policed: the work Examples 1-3 turn on in three directions (attack, proof, exculpation); and at the conflicts, where disagreeing layers need the discrepancy read rather than smoothed. Reported under CPR Part 35 with windows, grain and reach stated: and through **e-discovery.uk**, the layers travel intact into review, so the estate around each document is evidence, not an afterthought.

§ 13 · COMMON QUESTIONS

Frequently asked questions

Isn't this just metadata by another name?

Broader and more organised: "metadata" usually means fields on files (layer zero's own properties); the layer model covers the platform's separate record systems: series, events, states, histories, operations: most of which are not on or in the file at all and are missed precisely because they are not. Naming seven layers gives requests, schedules and reports a fixed structure that "please include metadata" never achieved.

Which layers should a standard commercial dispute actually collect?

By default: layer 2 exported estate-wide in week one (it expires; it is cheap); layers 5-6 wherever deletion or "routine" is in issue; layer 1 as series for the documents the case turns on; layers 3-4 where access or audience is disputed; layer 7 where process or conduct is. The schedule scales with the issues: the discipline is deciding per population, on the record, rather than defaulting to zero.

The opponent says their platform "doesn't keep that". How do we test it?

Platform-neutral request first (§5's wording), then the documentation: every major platform's layer capabilities are publicly documented, and "doesn't keep" usually means "didn't look", "didn't license" or "let it expire": three different answers with three different consequences. Ask which, in terms; request the settings and operations layers that would show it; and remember that a lapsed window after a preservation letter is not a technical answer but a conduct one.

Can layers be fabricated or edited the way documents can?

Far less readily: platform-maintained series, events and operations records are not user-editable, which is much of their weight: the realistic vulnerabilities are at export (curation, scope) and interpretation: met by documented queries, retained self-describing exports, verification patterns and grain honesty. In practice the layers are where fabrication of documents goes to die, not a new place for it to live: Example 1's version series being the standing illustration.

How does this model apply to platforms outside Microsoft and Google?

Directly: it is the interview you conduct with any SaaS estate: does it version, does it log, how are permissions computed, what are the sharing mechanisms, what survives deletion and for how long, what settings histories exist, what does the admin layer record: seven questions that scope Dropbox, Slack, Salesforce or a bespoke system in an afternoon. The SaaS guides ahead (112-114) are this interview, conducted platform by platform.

What is the single highest-value habit this guide recommends?

The week-one layer export: events and operations, full window, hashed and shelved, every cloud matter, before anyone knows whether they will matter: guide 57's instruction generalised to the whole estate. It is cheap, it is quick, it expires if skipped, and Examples 1-3 all rest on someone having done it: or on the other side having not.

§ 14 · REFERENCE

Glossary

As-at reconstruction
A layer's state computed for a past date from events and histories.

Clock column
The layer schedule's expiry entries; what must export this week.

Five-element join
Version + event + session + device + conduct; the attribution chain.

Grain
A layer's resolution; the cap on what findings may claim.

Layer schedule
The per-population table of layers to preserve, collect and produce.

Layer zero
The files themselves; necessary, and least revealing.

Operations layer
The platform's record of its own administration and preservation acts.

Platform-neutral wording
Requests naming layers, not vendor mechanisms.

Settings history
Configuration changes with dates; the routine-versus-conduct layer.

Vertical / horizontal moves
Across layers of one document; across holders of one layer.

Sources and authoritative references

The layered method in this guide draws on materials published by our laboratory and e-discovery practice, referenced here as sources:

- cflab.uk, digital forensics services (layered cloud collection, correlation and attribution workups, CPR Part 35 reporting): cflab.uk/digital-forensics-services · guides library: cflab.uk/guides
- e-discovery.uk, EDRM Phase 01 · Identification, Phase 02 · Preservation and Phase 03 · Collection (layer schedules, week-one exports, structural routes): e-discovery.uk/edrm/identification · e-discovery.uk/edrm/preservation · e-discovery.uk/edrm/collection
- e-discovery.uk, practice method and Knowledge Centre: e-discovery.uk · e-discovery.uk/knowledge
- Platform documentation referenced through this block: learn.microsoft.com, audit solutions · learn.microsoft.com, retention and deletion · support.google.com/vault
- PD 57AD and CPR Part 35: justice.gov.uk, PD 57AD · justice.gov.uk, Part 35
- ICO, UK GDPR guidance: ico.org.uk

§ HOW A SPECIALIST LABORATORY CAN ASSIST

Working with Computer Forensics Lab

The layered method is the laboratory's standing approach to every cloud estate: layer schedules drawn at scoping with the clock column governing week one; routes assigned per layer and integrity carried throughout; correlations built to the five-element standard with alternatives tested by signature; and the three directions of Examples 1-3: the under-asked production reopened, the multi-layer proof assembled, the suspected estate cleared: delivered under CPR Part 35 with grain and reach stated. The model travels: Microsoft, Google, and the SaaS estates the series turns to next, interviewed with the same seven questions. Through **e-discovery.uk**, layers survive processing into review rather than dying in exports. Conflicts checks, confidential scoping discussions and fixed-fee estimates are routine; and the first deliverable of any engagement is usually the same: the layer export that expires if it waits.

INSTRUCT THE LAB

Speak to a forensic examiner, not a salesperson.

Describe your matter in confidence and an examiner will respond the same working day. Conflicts checks and scoping calls with US counsel are routine.

NEW ENQUIRIES
+44 (0)20 7164 6971

EMAIL
info@cflab.uk

E-DISCOVERY
e-discovery.uk

Computer Forensics Lab Ltd · Euro House, 133 Ballards Lane, Finchley, London N3 1LJ, United Kingdom
cflab.uk · Customer service +44 (0)20 7164 6915 · ICO ZB395023 · NPCC · ISO 17025 aligned practice · UK Gov Digital Marketplace