



C O M P U T E R F O R E N S I C S L A B

London digital forensics & e-discovery · Established 2007

# Recovering Previous Versions of Cloud Documents

## The Drafting History SharePoint, OneDrive and Google Drive Keep Without Being Asked

A document produced in disclosure is one frame of a film. The cloud platforms most organisations now use: SharePoint and OneDrive, Google Drive, and their document-management cousins: record the earlier frames automatically: every save or autosave as a version, attributed to an account and a time, retained under policies most users never read. That history is evidence of the highest quality: who changed the price, when the indemnity was deleted, whether the "final" agreement was final, what the report said before the lawyer saw it. It is also fragile in ways that surprise: version limits, retention windows, the ways a copy or a move sheds history, and the difference between what the platform's interface shows and what its compliance tooling can export. This guide explains for UK lawyers how cloud version histories work, how to preserve and collect them forensically, how to read them for authorship and chronology, and how to use them where a document's history is the dispute.

© Estimated reading time: 24 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. Version-history work runs through its casework: authenticity examinations where the ladder brackets a questioned document, authorship attribution from version metadata, and collections engineered so the history survives the export: with reporting under CPR Part 35 where a document's drafting story is the issue.

ESTABLISHED 2007 · LONDON

ISO 17025-ALIGNED PROCEDURES

VERSION-HISTORY COLLECTION

AUTHORSHIP & CHRONOLOGY ANALYSIS

CPR PART 35 EXPERT REPORTS

FULL CHAIN-OF-CUSTODY DOCUMENTATION

§ CONTENTS

# In this guide

- 01 Executive summary
- 02 The problem in plain English: the file is a frame, the history is the film
- 03 How platforms keep versions: SharePoint, OneDrive, Google Drive and DMS systems
- 04 What kills history: limits, retention, copies, moves and migrations
- 05 Preserving and collecting versions forensically
- 06 Reading the ladder: authorship, chronology and the shape of drafting
- 07 Versions in dispute: authenticity, disclosure scope and production
- 08 Source architecture: where else the evidence lives
- 09 Worked examples
- 10 Common mistakes and technical limitations
- 11 Questions to ask · Suggested wording
- 12 Checklist and red flags · When to involve a digital forensic expert
- 13 Frequently asked questions
- 14 Glossary · References · Disclaimer · How a specialist laboratory can assist

## Executive summary

THE HEADLINE POINT: CLOUD PLATFORMS KEEP A DOCUMENT'S VERSION LADDER AUTOMATICALLY, ATTRIBUTED AND TIMESTAMPED: PRESERVE IT BEFORE LIMITS AND RETENTION PRUNE IT, COLLECT IT WITH THE HISTORY INTACT, AND READ IT AS THE DRAFTING RECORD IT IS

**How it works (§3):** SharePoint and OneDrive keep major (and optionally minor) versions per file, each recording the modifying account, the timestamp and the content as saved, with autosave in co-authored Office files producing versions at short intervals; Google Drive keeps a revision history for Docs, Sheets and Slides with per-edit attribution and, for uploaded files, a versions list; document-management systems keep their own version chains with check-in comments. **What destroys it (§4):** version limits (a maximum count, older versions pruned), library and retention settings, the loss of history on copy (a copied file is a new file with one version), on some moves and on migrations between platforms, and deletion of the file itself past its recycle window: each a clock or a trap. **Preserving and collecting (§5):** holds placed on the site or account before anything is touched; collection through compliance tooling with version export enabled (a default that is frequently off), or through platform APIs that enumerate the ladder, with each version hashed and its metadata (author, time, size, comments) captured: the guide 87 field discipline applied per rung. **Reading it (§6):** the ladder gives authorship at version granularity (which account saved what, when), chronology (when the clause appeared, when the number changed), and the drafting shape (bursts, gaps, late-night sessions) that guide 96's timeline joins to the surrounding record: with version diffs localising every change. **In dispute (§7):** the ladder brackets questioned documents (guide 87 and 109's authenticity work), settles "which version was sent", and raises its own disclosure questions: whether versions are within scope, how they are produced, and how privilege attaches to drafts: decided in the protocol per guide 94, not at exchange.

- **Versions are attributed:** each rung names an account and a time: authorship evidence the final file cannot give.
- **Limits and retention prune quietly:** the ladder's bottom rungs disappear on schedule: hold before the schedule does.
- **Copies shed history:** a copied or migrated file starts again: the original location's ladder is the one to collect.
- **Export must include versions:** compliance exports omit them by default on some platforms: the setting confirmed, the count reconciled.
- **The ladder is disclosure:** scope, production and privilege of versions are protocol questions: settled early.

## The problem in plain English: the file is a frame, the history is the film

When a contract, a report or a spreadsheet is disclosed, the recipient sees its final state and the metadata guide 87 taught them to read with caution. What they do not see, unless someone collected it, is the sequence of states the platform recorded on the way: the draft with the indemnity still in, the version where the price changed at 23:40, the autosave from the meeting in which the board's wording was replaced. Cloud platforms record that sequence because collaborative editing needs it: the "restore previous version" feature is the user-facing end of a compliance-grade history.

Two consequences follow for lawyers. First, the history is evidence: attributed, timestamped and usually beyond the reach of the user who would like it otherwise. Second, it is perishable: version limits prune it, retention policies age it out, and the everyday acts of copying, moving and migrating files reset it: so a document that has a ladder today may have a single rung by the time disclosure is scoped. This guide is the discipline for both facts: preserve the ladder while it exists, collect it in a form that survives, and read it for what it proves about who wrote what, when.

## How platforms keep versions: SharePoint, OneDrive, Google Drive and DMS systems

- **SharePoint and OneDrive:** document libraries keep major versions by default (minor versions optionally), each with the modifying account, timestamp, size and any check-in comment; autosave and co-authoring generate versions at frequent intervals, so an actively edited Office file may accumulate hundreds; the "Version History" view shows the ladder and the compliance and API layers expose it for collection: with per-library settings governing limits.
- **Google Drive:** native Docs, Sheets and Slides keep a granular revision history with per-edit attribution and the ability to name and pin versions; uploaded non-native files keep a versions list (replaced-file history) with retention rules of their own; Google Vault and the Drive API reach the revisions for collection.
- **Document-management systems:** iManage, NetDocuments and similar keep versioned documents with explicit check-in and version comments, author attribution and audit trails: the legal-sector estate's most structured ladders, collected through the system's export tooling (guide 75's territory).
- **Desktop-synced copies:** sync clients keep local copies of current versions and sometimes local caches of prior states; the authoritative ladder is the platform's, but the endpoint can corroborate (guides 51-58 and 111-112's client-side artefacts).
- **What a version contains:** the file content as saved (full copy or delta, depending on the platform), plus the platform metadata: the internal document metadata of guide 87 (author fields, revision counts) sits inside each rung and can be read per version.

## What kills history: limits, retention, copies, moves and migrations

- **Version limits:** libraries cap the number of versions retained (a configurable maximum; with autosave, the cap can be reached in days of active editing), pruning the oldest: the early drafting of a heavily edited file may be gone while the file looks current.
- **Retention and deletion:** platform retention policies and version-expiry settings age versions out; deleting the file sends it and its ladder to the recycle bin, then second-stage recycle, then gone (guide 51-58's windows): with holds suspending the pruning while they apply.
- **Copying:** a copied file is a new item with a single version and a new creation date: the history stays with the original: a document produced from a copy has no ladder, and guide 87's created-date cliff appears.
- **Moving and migrating:** moves within a platform may preserve history or not depending on method and platform; migrations between platforms (file server to SharePoint, one tenant to another, DMS upgrades) frequently flatten ladders to a single version and rewrite dates (guide 96 Example 3's migration-day effect): the migration records become the ladder's obituary and, sometimes, its replacement.
- **User actions:** "delete version history" and equivalent commands exist, are logged in audit trails, and are as suspicious as they sound: guide 95's timing analysis applies to version deletions as to anything else.

## Preserving and collecting versions forensically

- **Hold first:** the site, library or account placed under hold so pruning and deletion are suspended: before anyone opens, moves, copies or "tidies" the relevant files: the guide 97 §3 principle for documents.
- **Collect with versions enabled:** compliance tooling (Purview eDiscovery, Google Vault) can export version histories, but the option is frequently off by default or limited in scope: the setting confirmed in the collection plan, and the exported version count reconciled against the platform's ladder for sampled files.
- **API enumeration where tooling falls short:** platform APIs list every version with its metadata and allow each to be downloaded and hashed: the method of choice for targeted files where the complete ladder matters, documented per guide 71's reproducibility principle.
- **Metadata per rung:** for each version: version number, modifying account, timestamp (in the platform's frame, normalised per guide 96), size, comment, hash: the ladder's own manifest, produced alongside the content.
- **Audit-log correlation:** SharePoint and Drive audit logs record file operations (edits, downloads, shares, version deletions, moves) independently of the ladder: collected together (guide 105's windows), because the log survives what the ladder does not and vice versa.

## Reading the ladder: authorship, chronology and the shape of drafting

- **Authorship at version granularity:** the modifying account per rung: with the attribution disciplines of guides 105-107 joining account to person, and co-authoring's attribution rules understood (the account that triggered the save may not be the only editor in a co-authored session: stated as such).
- **Chronology of content:** diffs between consecutive versions localise each change: when the clause entered, when the figure moved, when the paragraph vanished: guide 96's timeline gaining content-level events with platform timestamps.
- **The drafting shape:** version bursts and gaps, session lengths, out-of-hours activity, the single evening in which a "long-standing" document acquired all its history (guide 82 Example 3, guide 98 Example 2): the pattern evidence that suggests reconstruction rather than drafting.
- **Internal metadata per version:** guide 87's document fields read inside each rung: author names, revision counts, total editing time, template origins: consistent ladders corroborate; discontinuities (a rung whose internal author differs from its platform account) raise questions.
- **Cross-referencing transmission:** which version was attached to which email or shared with whom (guides 51-58's sharing logs, guide 87's attachment hashes): the "which version did they see?" question answered by hash-matching rungs to attachments.



## Versions in dispute: authenticity, disclosure scope and production

- **Bracketing the questioned document:** a document claimed to date from March whose ladder starts in June, or whose only rung was created the week proceedings loomed: the version history as guide 109's backdating evidence and guide 87's corroboration lattice, from the platform's own records.
- **Disclosure scope:** whether prior versions are "documents" within the disclosure model (they are documents; whether they are within scope is a proportionality and issues question), and how far down the ladder scope reaches: agreed in the DRD, with the census of §5 informing the count.
- **Production of versions:** native production of selected rungs with their metadata, version manifests as load-file fields, and the family question (versions as related items): guide 94's protocol deciding the format before exchange.
- **Privilege in drafts:** drafts prepared for or reflecting legal advice attract privilege the final may not, and vice-versa: the ladder reviewed per guide 91's per-segment discipline: with the practical warning that a version produced unreviewed is guide 92's subject.
- **Requesting the opponent's ladders:** targeted requests for the version history of identified documents, with audit-log entries for version deletions and migrations: §11's wording, aimed at the documents whose history is the dispute.

## Source architecture: where else the evidence lives

Per this series' source-architecture discipline: a document's history lives in the platform's ladder first, and in several other places that survive when the ladder does not. The platform version history (SharePoint, OneDrive, Drive, DMS) is the authoritative sequence. The audit logs record file operations, including the version deletions and moves that explain a pruned ladder. Email and messaging attachments (guides 51-58, 63-66) hold snapshots of the document as sent, hash-matchable to rungs or standing in for pruned ones: the counterpart's copy of the March draft proves the March draft existed whatever the ladder now shows. Sync-client caches and local copies on endpoints hold states the platform may have lost. Backups of the platform (tenant backups, DMS backups) hold ladders as they were at backup time, before limits and migrations pruned them. And the document's own internal metadata (guide 87's revision counts and editing-time fields) carries a compressed history inside the final file. When the ladder has been flattened by copy or migration, pruned by limits, or deleted by a user, the table's columns supply the alternates.

| EVIDENCE                          | PLATFORM<br>VERSION<br>LADDER | AUDIT<br>LOGS                   | ATTACHMENTS /<br>SENT COPIES | ENDPOINT<br>CACHES +<br>LOCAL<br>COPIES | PLATFORM<br>BACKUPS     | INTERNAL<br>DOCUMENT<br>METADATA |
|-----------------------------------|-------------------------------|---------------------------------|------------------------------|-----------------------------------------|-------------------------|----------------------------------|
| <b>Prior content states</b>       | Y: the sequence               | n/a                             | Y: snapshots as sent         | Sometimes: cached states                | Y: as at backup         | n/a                              |
| <b>Who edited, when</b>           | Y: per rung                   | Y: operations + accounts        | Sender + date bracket        | User context                            | Ladder as backed up     | Author + revision fields         |
| <b>Why the ladder is short</b>    | n/a                           | Y: deletions, moves, migrations | Existence of earlier states  | n/a                                     | Pre-pruning ladder      | Revision count inconsistency     |
| <b>Which version was sent</b>     | Hash-match target             | Share + download events         | Y: the direct answer         | Sent-items caches                       | n/a                     | Matching internal fields         |
| <b>Reconstruction vs drafting</b> | Y: burst patterns             | Session timing                  | Absence of earlier sends     | n/a                                     | Ladder before the burst | Y: editing-time + sessions       |

Fallback strategy in one line: when the ladder is gone, prove the history from the audit log, the sent copies and the backup: and when the ladder is suspiciously short, the audit log explains why.

## Worked examples

### EXAMPLE 1 · THE INDEMNITY THAT LEFT AT 23:40

A supply-agreement dispute turned on whether an indemnity clause had been agreed. The executed PDF lacked it; the claimant's negotiators swore it had been in the final draft. The defendant's SharePoint ladder for the Word source: held in time under the DRD's preservation terms: showed 214 versions. Version 209, saved at 17:12 by the defendant's commercial lead, contained the clause; version 210, at 23:40 the same evening by the same account, did not; the diff was the clause and nothing else; and version 211 the following morning was the "clean final" circulated for signature with a covering email reading "no changes from yesterday's agreed draft". The §6 reading: authorship, chronology, diff and transmission matched to the sent copy: was exhibited with the ladder manifest. The claim resolved on the evidence; the defendant's own platform had kept the record its negotiator had hoped nobody would read.

### EXAMPLE 2 · THE MIGRATION THAT FLATTENED THE LADDER, AND THE BACKUP THAT DID NOT

An authenticity challenge to a "2021" policy document met a SharePoint ladder with a single rung dated 2023: consistent with fabrication, the challenger said. The §8 map disagreed. The audit log showed a tenant migration in 2023 in which every file in the library had been re-created with one version and a migration-day timestamp (guide 96 Example 3's pattern, seen from the other side); the pre-migration tenant's backup, restored by the examiner, held the original library with a 2021-2022 ladder of thirty-one versions for the document; and three of those rungs hash-matched attachments in 2021 emails. The document was genuine; the flattened ladder was an artefact of IT, not of fraud. The examiner's report stated both the finding and the method, and the challenge was withdrawn. Ladders are evidence; short ladders are questions, answered from the audit log before anyone pleads.

### EXAMPLE 3 · THE EXPORT THAT FORGOT THE VERSIONS

A provider's Purview export of a disputed spreadsheet arrived as a single file: current version only, because version export had been left at its default. The opponent, who had asked for the version history specifically, called the omission. The provider re-collected via the API: 46 versions, each hashed with its metadata: and the reconciliation showed the export tooling's default had silently dropped the ladder for every file in the production. The re-collection cost a fortnight and a costs concession; the §5 setting would have cost a checkbox. The case's substantive point was made by version 12, which showed the formula driving the pleaded valuation being hard-coded to a number three days before the board pack: but the example's lesson is the collection plan: "with versions" is a setting, a reconciliation and a line in the methodology statement, or it is not there.

## Common mistakes and technical limitations

### Common mistakes

- **Exports without versions:** Example 3's default: the ladder dropped by tooling nobody configured.
- **Collecting the copy:** the document taken from the email attachment or the copied folder: one rung, no history: the original location never identified.
- **Holds after the tidy-up:** files moved, renamed or "cleaned" before preservation; ladders flattened by well-meant housekeeping.
- **Short ladders read as fraud:** Example 2's mistake: migration and limit effects pleaded as fabrication without the audit log.
- **Co-authoring attribution overclaimed:** the saving account named as sole editor of a session five people shared.
- **Versions unreviewed for privilege:** the draft with counsel's comments produced because only the final was reviewed: guide 92 by way of guide 102.
- **Scope silent on versions:** the DRD and protocol never mentioning prior versions; the dispute arriving at exchange.
- **Timestamps unnormalised:** platform UTC and local presentation mixed in the ladder analysis: guide 96's error, per rung.

### Technical limitations

- **Limits are real:** pruned versions are gone from the platform: the alternates of §8 or nothing, stated honestly.
- **Deltas and reconstruction:** some platforms store versions as deltas; reconstruction of a specific rung's full content depends on the chain's integrity: verified, not assumed.
- **Autosave granularity varies:** the interval between versions is platform- and session-dependent: "no version between X and Y" bounds a change's timing only to the interval.
- **API and tooling change:** export options and scope move with the platforms: verified per matter, per this series' standing rule.
- **Non-native files have thinner histories:** uploaded PDFs and binaries keep replacement histories, not edit-level ladders: the evidence graded accordingly.

## § 11 · INTERROGATORIES &amp; DRAFTING AIDS

## Questions to ask · Suggested wording

### Ask your client

- Where does the original of each key document live: the platform, library and path: not the copy in someone's email?
- What are the version limits and retention settings on those libraries, and has any migration or bulk move touched them?
- Has anyone tidied, copied or reorganised the relevant folders since the dispute arose?

### Ask your opponent

- Please disclose the complete version history of [documents], from the platform location in which each was originally created and edited, with for each version the modifying account, timestamp, size and content, and confirm the version-limit and retention settings applicable to that location during [period].
- Please disclose the audit-log entries relating to [documents] for [period], including any version deletions, moves, copies or migrations affecting the version history.
- Please confirm whether the collection underlying your disclosure exported version histories, and if not, the basis on which prior versions were excluded from scope.

### Ask your eDiscovery / forensic provider

- Was version export enabled, and does the exported count reconcile to the platform ladder for sampled files?
- Are the rungs hashed and their metadata normalised: and have they been matched to sent attachments?
- What does the audit log say about any short ladder before we characterise it?

### SUGGESTED WORDING · VERSION-HISTORY PARAGRAPH FOR THE DRD / PROTOCOL

"Prior versions of documents held in cloud document platforms or document-management systems are documents for the purposes of disclosure. Each party shall preserve version histories by placing the relevant sites, libraries and accounts under hold before any reorganisation, and shall collect from the original platform location with version export enabled, recording for each version the modifying account, timestamp, size and hash. Prior versions of the documents identified in Schedule 1 [and of any document whose authenticity, date or authorship is in issue] shall be produced natively with a version manifest; the scope of version production for other documents shall be [as agreed / determined on request by reference to the issues]. Audit-log entries recording deletion, movement, copying or migration of documents within scope shall be preserved and disclosed on request."

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

### The version-history checklist

- ☐ Original platform location identified for each key document
- ☐ Sites, libraries and accounts held before any housekeeping
- ☐ Version limits, retention and migration history established
- ☐ Collection plan states version export enabled; counts reconciled
- ☐ Each rung hashed with account, timestamp, size, comment
- ☐ Audit logs collected alongside within their retention windows
- ☐ Timestamps normalised per guide 96 before analysis
- ☐ Rungs hash-matched to attachments and sent copies
- ☐ Versions reviewed for privilege before any production
- ☐ Version scope and production format agreed in the protocol

### Red flags

- Single-rung ladders for long-lived documents: Example 2's question.
- Version bursts in one session near a critical date: reconstruction's signature.
- Audit-log entries for version deletion during the dispute.
- Exports whose file counts match but version counts do not: Example 3's default.
- Key documents held only as email copies.
- Migrations or bulk moves after contemplation of proceedings.
- Opposing productions of "finals" where drafting was clearly collaborative.

### When to involve a digital forensic expert

At preservation, where the original locations are identified and held before limits, retention or housekeeping prune the ladder; at collection, where version export is engineered, reconciled and manifested (Example 3's checkbox, done properly); at analysis, where the ladder is read for authorship, chronology and drafting shape and joined to the transmission record (Example 1's evening); and at challenge, where short ladders are explained from the audit log and backups (Example 2's migration) and reported under CPR Part 35. Through **cflab.uk** and **e-discovery.uk**, version-history work runs from hold to exhibit as one discipline: the document's film, not its frame.

## § 13 · COMMON QUESTIONS

## Frequently asked questions

Are prior versions really "documents" we have to disclose?

They are documents; whether they fall within scope is an issues-and-proportionality question answered in the DRD (§7). Where a document's date, authorship or content history is in issue, its ladder is usually squarely relevant; for the bulk of a corpus, agreed limits are normal. What is not acceptable is silently producing finals only because the export defaulted that way.

The version history on our key document is missing. Does that look bad?

It looks like a question, and the audit log usually answers it: version limits, migrations, moves and copies flatten ladders innocently every day (Example 2), while user-initiated version deletion during a dispute is logged and looks exactly as bad as it is. Establish the cause from the records before anyone characterises it, in either direction.

Can the platform tell us who made a specific change?

To the account that saved the version containing it, yes: diffs localise the change to a rung, the rung names the account and time (§6): with co-authoring caveats (several editors in one session) and the guide 105-107 disciplines joining account to person. Example 1's 23:40 deletion was attributed exactly that way.

We only have the document as an email attachment. Is that enough?

It is a snapshot with a date bracket, useful for hash-matching and for proving a state existed: but it has no ladder. Find the original platform location and collect there (§5); the attachment then corroborates the rung it matches. Collecting only copies is the most common way a case loses history it never knew it had.

How far back do version histories go?

As far as limits and retention allowed: potentially years for lightly edited files, potentially days for heavily co-authored ones under a low version cap (§4). The answer is per library and per file, established from settings and the ladder itself: and where the platform has pruned, §8's alternates (backups, sent copies, audit logs) extend the reach.

Do drafts carry privilege the final does not?

They can: a draft prepared for legal advice, or reflecting it, attracts privilege on its own merits regardless of the final's status: and the reverse: which is why versions are reviewed per guide 91's segment discipline before production, not assumed to share the final's coding. A privileged draft produced unreviewed is guide 92's subject.

## § 1 4 · REFERENCE

# Glossary

## Autosave version

A rung created automatically during co-authored editing.

## Diff

The localised difference between two consecutive versions.

## Ladder flattening

Loss of history through copy, move or migration.

## Major / minor versions

SharePoint's published and draft version tiers.

## Revision history

Google Drive's per-edit version record for native files.

## Rung

A single version in a document's ladder.

## Version export

The collection setting that includes prior versions.

## Version ladder

The complete attributed sequence of a document's saved states.

## Version limit

The library setting capping retained versions.

## Version manifest

The per-rung metadata and hash schedule produced with a collection.

## Sources and authoritative references

The version-history disciplines in this guide draw on materials published by our laboratory and e-discovery practice, referenced here as sources:

- cflab.uk, digital forensics services (version-history collection, authorship and authenticity analysis, CPR Part 35 reporting): [cflab.uk/digital-forensics-services](https://cflab.uk/digital-forensics-services) · guides library: [cflab.uk/guides](https://cflab.uk/guides)
- e-discovery.uk, EDRM Phase 04 · Collection and Phase 07 · Analysis (version export, manifests and ladder analysis as practised): [e-discovery.uk/edrm/collection](https://e-discovery.uk/edrm/collection) · [e-discovery.uk/edrm/analysis](https://e-discovery.uk/edrm/analysis)
- e-discovery.uk, practice method and Knowledge Centre: [e-discovery.uk](https://e-discovery.uk) · [e-discovery.uk/knowledge](https://e-discovery.uk/knowledge)
- Microsoft Learn, versioning in SharePoint and OneDrive: [learn.microsoft.com](https://learn.microsoft.com), [versioning](#) · Google Workspace Admin Help, Vault and Drive: [support.google.com/vault](https://support.google.com/vault)
- PD 57AD: [justice.gov.uk](https://justice.gov.uk), [PD 57AD](#) · CPR Part 31: [justice.gov.uk](https://justice.gov.uk), [Part 31](#) · CPR Part 35: [justice.gov.uk](https://justice.gov.uk), [Part 35](#)
- ICO, UK GDPR guidance: [ico.org.uk](https://ico.org.uk) · ISO/IEC 27037: [iso.org](https://iso.org), [27037](#)

## DISCLAIMER

This publication provides general information about recovering previous versions of cloud documents as at August 2026. Platform versioning behaviour, limits, retention defaults and export tooling change; verify the current position for the platform and matter at hand. The worked examples are fictional. This guide is not legal advice, does not create a professional relationship, and should not be relied upon in place of advice on the facts of a specific matter from a qualified lawyer or a suitably qualified forensic practitioner. Links were live at the date of publication.



## § HOW A SPECIALIST LABORATORY CAN ASSIST

## Working with Computer Forensics Lab

Version work at **e-discovery.uk** begins by finding the original location and holding it before housekeeping, limits or retention touch the ladder; collection runs with version export engineered and reconciled (Example 3's checkbox, plus the API where tooling falls short), each rung hashed and manifested; and analysis reads the ladder for authorship, chronology and drafting shape on guide 96's normalised frame, joined to the transmission record by hash-matching (Example 1's evening). The laboratory side explains short ladders from audit logs and backups before anyone pleads them (Example 2's migration), examines questioned documents against their histories, and reports under CPR Part 35 where the drafting story is the case. Conflicts checks, confidential scoping discussions and fixed-fee estimates are routine; and if a key document is being "tidied into the right folder" this week, the hold comes first: the ladder does not survive the move.



## I N S T R U C T T H E L A B

### 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