Reconstruction-Grade eDiscovery Standard

The System of Record for Modern Collaborative Evidence

Peter Kozak Brandon D’Agostino

Version 0.5-draft — February 2026

Abstract

This document specifies Reconstruction-Grade eDiscovery as an architectural standard for modern collaborative evidence. A Reconstruction-Grade system preserves a reproducible, point-in-time evidentiary record that can answer legal and operational questions about what happened, when it happened, who was involved, what was relied upon, and what was accessed — without substituting inference for preserved fact.

Status: Draft for standards discussion

Canonical Source: Markdown in Git repository

Contents

0. Scope, Non-Goals, and Document Conventions

0.1 Scope

This document specifies a baseline architectural standard for Reconstruction-Grade eDiscovery in collaborative cloud environments. The standard focuses on the minimum properties required to preserve and export evidence such that reconstruction is deterministic, auditable, and reproducible.

0.2 Non-Goals (Out of Scope)

This standard does not: - Provide legal advice or define admissibility standards; it defines an architectural preservation and export baseline. - Require or assume full-tenant “archive everything” capture. - Mandate a specific vendor, product, cloud provider, storage backend, review platform, or deployment topology. - Guarantee perfect behavioral visibility; audit evidence is explicitly bounded by licensing, retention, and upstream availability. - Require semantic analytics, relevance scoring, or LLM-based workflows. - Replace existing review platforms; it specifies what an upstream preservation record MUST provide to downstream systems.

0.3 Threat Model (What This Standard Is Designed to Withstand)

Reconstruction-Grade properties matter because evidence is evaluated under adversarial or high-scrutiny conditions, including: - Court challenges to authenticity, completeness, proportionality, and chain of custody. - Regulator inquiries requiring auditability and repeatability. - Internal investigations where timelines, identity state, and access behavior are disputed. - Repeat exports where differences MUST be explained without operator narrative.

This standard assumes that any ambiguous, inferred, or non-reproducible claim will be challenged. Therefore, it prioritizes: - Deterministic resolution rules and explicit fallback documentation. - Stable identifiers and relationship bindings. - Explicit exceptions (no silent drops). - Export manifests with hash and integrity support. - Clear boundedness statements when upstream evidence is incomplete.

0.4 Normative Language

This document uses MUST, SHOULD, and MAY as normative requirement levels: - MUST indicates a required property for Reconstruction-Grade conformance. - SHOULD indicates a strong recommendation that materially improves reconstruction. - MAY indicates an optional capability that can improve outcomes but is not required for baseline conformance.

1. The Structural Shift in Evidence Behavior

The shift from on-premises file shares and email attachments to cloud-native collaboration has changed the evidentiary substrate. The change is not primarily about volume. It is about how work is performed, recorded, and referenced.

1.1 The Core Assumptions That No Longer Hold

Traditional eDiscovery workflows tend to embed four assumptions: - Files are the unit of work and the unit of evidence. - Ownership implies relevance; custodians are stable boundaries. - Permissions imply access; access can be inferred. - Versions are interchangeable; the latest version is “close enough.”

Collaborative cloud environments violate these assumptions by design. Work occurs through hyperlinks instead of attachments, co-authoring instead of discrete drafts, shared repositories instead of personal storage, and continuous revision instead of immutable records.

Table 1. Legacy Assumptions vs Collaborative Reality

Evidence Dimension	Legacy Assumption	Collaborative Reality
Unit of work	File / attachment	Activity + link + shared repository object
Evidence capture	Final-state collection	Point-in-time resolution per event
Custodians	Static containers	Natural persons with effective-dated identity
Access	Inferred from permissions	Observed via audit evidence where available
Versioning	Minor or ignored	Continuous; non-linear growth; version lineage is evidentiary
Messages	Immutable email	Threaded, editable, multi-modal conversations

1.2 Hyperlinks Replaced Attachments

In legacy email systems, the attachment and the message were inseparable: the bytes were embedded and fixed at send time. In Microsoft 365, messages frequently contain hyperlinks or modern attachments that reference repository objects. The message does not preserve the document behind the link.

Defensible reconstruction therefore depends on preserving both the communication event and the referenced object state. A preserved file without its event bindings cannot explain what it meant at the time it was used.

Scenario: Hyperlinked decision memo A Teams message links to a OneDrive document used to approve a high-impact decision. The file is edited after the message is sent. A traditional export collects the current file bytes. The exported bytes are not the bytes that informed the decision at the time. Without deterministic point-in-time resolution, reconstruction becomes narrative-driven.

1.3 Version Lineage Is Now Evidentiary

Version history is not merely a storage feature. In collaborative systems, version lineage is the audit trail of authorship, evolution, and reliance.

Reconstruction-Grade eDiscovery requires that file versions be enumerated and preserved with stable identifiers, timestamps, and bytes. Point-in-time queries must be able to select a version at or before an event timestamp deterministically.

Figure 2 — As-Sent vs Accessed Version Resolution

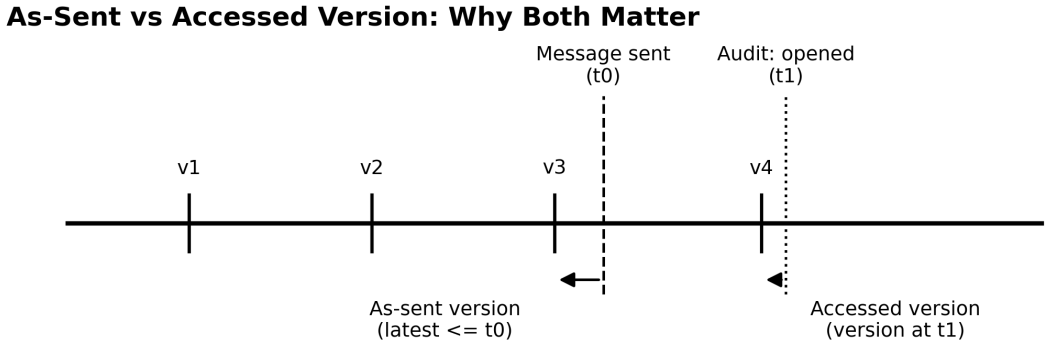

Figure 1: As-sent vs accessed versions: point-in-time reconstruction requires both message-bound state and audit-observed interaction

1.4 Identity Drift and the Static Custodian Myth

Custodians are not directory objects frozen in time. People change roles, teams, reporting lines, access rights, and responsibilities. Legal questions are temporal. Discovery requests are drafted around historical periods, not present-day org charts.

Effective identification therefore requires effective-dated identity: a historical record of who a person was during the relevant period (department, role, manager, status, group membership), not merely who they are today.

1.5 Permissions Are Not Proof of Behavior

Permissions describe potential access. They do not prove actual interaction. Access is time-bound, revocable, and contextual. When discovery decisions rely on inferred access instead of observed behavior, evidence is replaced by assumption.

Reconstruction-Grade systems treat audit logs as first-class evidence inputs and correlate them to preserved objects and version timelines to answer: who saw what, when.

1.6 Context Decays Over Time

A defining characteristic of the context gap is that context decays. People leave organizations, repositories are repurposed, links break, and audit logs age out. Traditional eDiscovery begins after this loss has occurred.

Reconstruction-Grade practice assumes that waiting is the mistake: if context will be required later, it must be preserved while it exists.

2. The Context Collapse Problem

Context collapse occurs when collaborative evidence is flattened into files and messages that cannot carry the full burden of explanation. The result is not merely inefficiency. It is reconstruction failure: the inability to reproduce what the actors experienced at the time.

2.1 Failure Mode: Final-State Collection

Final-state collection captures what exists today. It does not capture what existed at the relevant time, what was shared, what was relied upon, or what version was accessible when a communication occurred.

In collaborative systems, "latest" is not a substitute for "as-of." When the evidence model treats it as such, the record becomes time-shifted.

2.2 Failure Mode: Archive-Everything-First

Archive-everything-first strategies assume that comprehensive capture is the path to defensibility. At enterprise scale, this approach often delays legal outcomes.

Two realities dominate: (1) preserved footprint grows non-linearly once version history and indexing are included, and (2) Microsoft 365 service limits (API throttling), permissions, retries, and network egress govern time-to-ingest.

2.3 Failure Mode: Static Identity and Custodian-Centric Scoping

Custodian-centric scoping assumes that ownership maps to relevance and that repository boundaries align with work boundaries. Modern work is cross-functional, link-based, and shared. Relevant artifacts frequently reside outside the custodian's owned spaces.

2.4 Failure Mode: Permission-Based Inference

Inferring access from permissions is convenient but methodologically weak. It produces statements like "it appears that…" rather than "the preserved record shows

that…”. Under scrutiny, inferred narratives are fragile.

2.5 Practical Symptoms of Context Collapse

- Escalating over-collection due to uncertainty about where relevant content lived.
- Version explosion in review datasets without a defensible rule for which versions matter.
- Inability to explain which document state informed a decision at a given time.
- Difficulty defending proportionality decisions because scope was not evidence-based.
- Gaps in “who saw what, when” because audit signals were not preserved or correlated.

3. Defining Reconstruction-Grade eDiscovery

Reconstruction-Grade eDiscovery is an architectural classification. It describes whether an evidence system can produce a reproducible, point-in-time record of collaborative activity without relying on hindsight or inference.

Normative language: This document uses MUST, SHOULD, and MAY in their standards sense (see Section 0.4).

3.1 The Reconstruction-Grade Evidence Model

A Reconstruction-Grade evidentiary record is constructed from three pillars. Weaknesses in any pillar forces downstream inference.

Identity over time: Authoritative, effective-dated identity correlated to a natural person (role, department, manager, status, group membership).

Behavior and activity evidence: Audit and activity records treated as evidence to establish interaction (view, edit, share, access) and timing - explicitly bounded by availability and retention.

Document state and relationships: Deterministic point-in-time file resolution and explicit message ↔ link ↔ file ↔ version bindings preserved using stable identifiers and lineage metadata.

Figure 3 — Evidence Graph: Objects and Relationships Preserved for Reconstruction

3.2 Key Definitions

Modern attachment / Hyperlinked file - A message-level reference to a repository object (typically via hyperlink or platform-managed pointer) where the bytes are not embedded in the message.

As-sent version - The file version that existed at the time a communication was sent; deterministically resolved as the latest version whose lastModifiedDateTime is not later than the message timestamp.

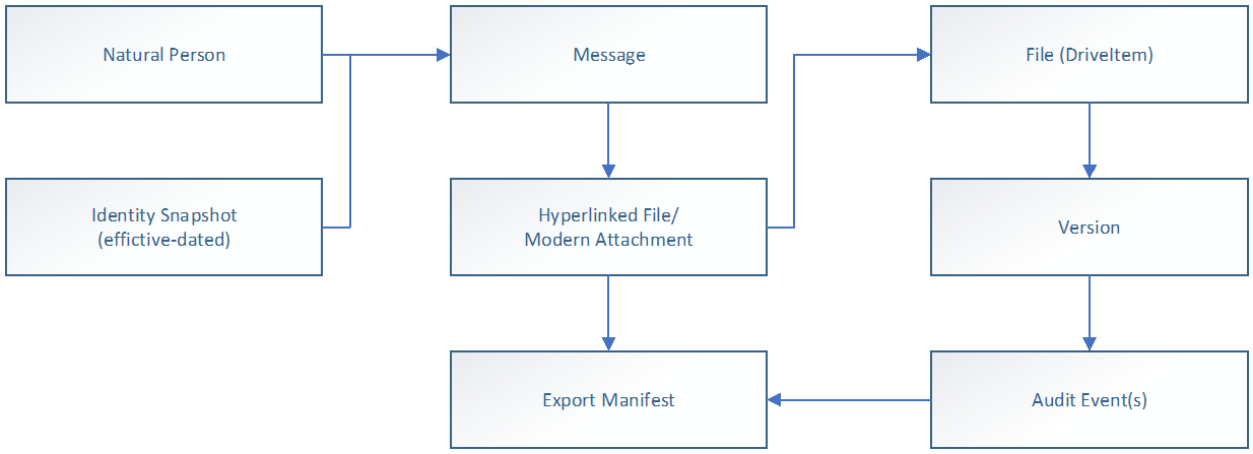

Figure 2: Evidence Graph: Reconstruction requires objects, relationships, and timelines to be preserved as a coherent record

Accessed version - The file version that a specific actor actually opened or interacted with, derived from audit evidence where available and correlated to preserved version timelines. If audit evidence is unavailable or out of retention, the accessed-version claim MUST be represented as unknown rather than inferred.

Evidence graph - A structured record linking people, identities, events, artifacts, versions, and audit signals with stable identifiers and timestamps.

Deterministic end state - A processing outcome in which every discovered reference either resolves to preserved content or results in a structured exception record with traceable reasons and retry history.

3.3 Deterministic Version Resolution

Reconstruction-Grade systems MUST be able to resolve document state as-of an event timestamp deterministically. For modern attachments, the event timestamp is typically the send time of the message containing the link.

A minimal deterministic selection rule is: - Enumerate all available versions and their timestamps for the referenced object. - Select the latest version where lastModifiedDateTime ≤ message timestamp. - If multiple versions share the same timestamp, select the highest version identifier under a deterministic ordering. - If version history is unavailable or incomplete, record the applied fallback rule and outcome explicitly.

3.4 Stable Identifiers and Canonicalization

Because URLs can change (sharing links, redirects, file moves), Reconstruction-Grade systems MUST preserve stable platform-native identifiers sufficient to re-resolve content over time.

At minimum, where applicable, systems MUST persist: - siteId - driveId - itemId -

listItemUniqueId - versionId

Canonicalized SharePoint/OneDrive URLs (as an aid, not as the primary key)

3.5 Relationship Preservation: Message ↔ Link ↔ File ↔ Version

Relationships are evidentiary. If relationships are lost or collapsed during processing, the record can no longer explain reliance and transmission. Reconstruction-Grade systems MUST preserve explicit parent-child relationship mappings between communications and referenced objects.

Recommended export representations include ParentId/ChildId relationship fields with RelationshipType values that distinguish modern attachments, hyperlinks, embedded objects, and other bindings.

3.6 Audit Evidence and the Access Question

Reconstruction-Grade practice distinguishes between: - Potential access (permissions and sharing configuration), and - Observed access (behavioral evidence from audit logs where available).

Permissions describe what could have happened; they do not prove what did happen. Where audit logs exist, systems SHOULD ingest them, treat them as evidence, and correlate them to preserved objects and version timelines to support defensible answers to “who saw what, when.”

Boundedness requirement: Because audit coverage is dependent on upstream availability, licensing, and retention windows, a Reconstruction-Grade system MUST represent audit-based claims with explicit coverage bounds. If audit evidence does not exist for the relevant timeframe, the system MUST represent observed access as unknown and MUST NOT substitute permission-based inference as a factual claim.

3.7 Deterministic Exception Records

Not every referenced object can be collected. Reconstruction-Grade systems MUST make this explicit rather than silent.

A structured exception record SHOULD include: - Original link reference as extracted from the parent communication - Normalized reason code (permission denied, item deleted/outside retention, throttling, transient service error, unknown) - Complete retry history (attempt count, outcomes, timestamps) - Association to the parent communication and to the attempted target identifiers

3.8 Reproducible Exports and Evidence Manifests

Reconstruction-Grade systems MUST support reproducible exports. The same scope definition and parameters should generate the same outputs, with manifests that support chain-of-custody and defensibility.

A Reconstruction-Grade export package SHOULD include: - Native files and communications - Standard load/overlay files (e.g., DAT/CSV) containing metadata, relationship identifiers, provenance fields, and hashes - An export manifest capturing counts, scope queries, time ranges, tool versions, exceptions, and retry outcomes - Resumable export jobs with full auditability

4. The Preservation System of Record

Modern collaboration platforms are operational systems. They are designed to enable work, not to serve as historical systems of record for litigation-grade reconstruction. Review platforms are downstream consumers. Archives are storage systems. Compliance tooling is policy-oriented.

Reconstruction-Grade eDiscovery requires an explicit Preservation System of Record: a layer that preserves context, determinism, and evidentiary traceability without flattening collaborative reality.

Figure 4 — Reference Architecture: Preservation System of Record

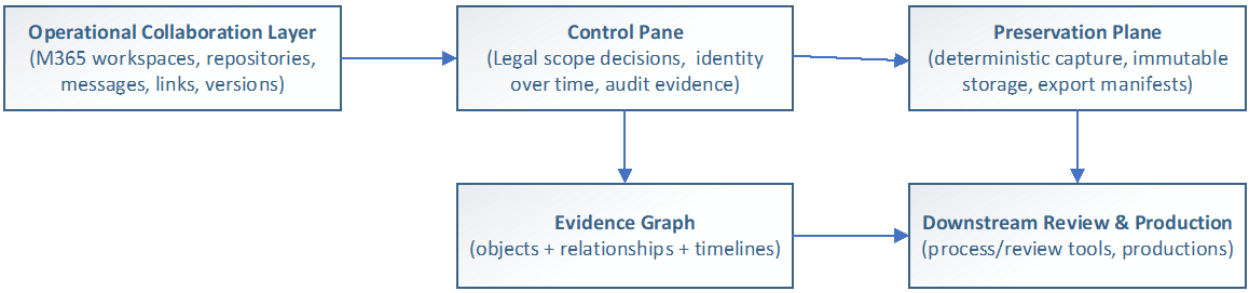

Figure 3: Reference architecture: a Preservation System of Record separates legal decisioning from preservation mechanics and downstream review

4.1 Responsibilities of a Preservation System of Record

- Maintain an evidence graph linking people, identities, events, artifacts, versions, and audit signals.
- Provide a decision ledger: immutable record of scope definitions, preservation triggers, queries, and exceptions.
- Perform deterministic capture of content and relationships for in-scope matters.
- Persist stable identifiers and lineage metadata sufficient for future re-resolution.
- Produce reproducible exports with manifests, hashes, and complete exception traceability.

4.2 What a Preservation System of Record Is Not

- It is not a downstream review platform (those systems assume the evidence has already been normalized).
- It is not simply a storage archive (storage without context does not reconstruct).
- It is not merely a compliance policy layer (policy does not preserve point-in-time relationships).
- It is not an analytics layer that infers missing context after the fact.

4.3 Evidence-Based Scoping

At enterprise scale, scoping is not a clerical step. It is a defensibility-critical decision. Reconstruction-Grade operating models treat identification and scope as evidence-based processes: identity over time, repository usage patterns, and audit-derived interaction signals are used to determine what must be preserved.

Design Principle: Preserve one authoritative object; never collapse context A single file object may be referenced across many messages, channels, and actors. Deduplicating the bytes can be operationally correct. However, collapsing the events (who shared, where, when, which version) destroys context. Reconstruction-Grade systems preserve the object once while retaining all contextual bindings.

4.4 Immutability and Auditability

Reconstruction-Grade evidence must be immutable once preserved, and the system must maintain an audit trail sufficient to reproduce and defend: what was in scope, why it was in scope, what was collected, what failed, what was retried, and what was exported.

This includes not only content-level hashes but also workflow-level audit trails for scope decisions and exception handling.

5. Preservation Operating Model and Architectural Constraints

Reconstruction-Grade eDiscovery requires more than correct data structures. It requires a preservation operating model that delivers defensible legal outcomes under real-world enterprise constraints. This section defines the architectural requirements for that operating model, presents conforming preservation postures, and specifies the constraints under which all postures operate.

The governing design question is not whether data can be retained. It is: which operating model delivers a defensible, reproducible evidentiary record soonest under the constraints imposed by collaborative cloud platforms?

5.1 Architectural Role of the Preservation System of Record

Section 4 defines the Preservation System of Record as an explicit architectural layer. This section specifies how the operating model interacts with that layer.

A conforming operating model MUST route preservation actions through a Preservation System of Record (or equivalent) that satisfies the following properties: - Evidence graph maintenance. The system MUST maintain a linked record of people, identities, events, artifacts, versions, and audit signals with stable identifiers. - Decision ledger. The system MUST record scope definitions, preservation triggers, queries, and exceptions as an immutable, auditable ledger. - Deterministic capture. The system MUST perform deterministic capture of content and relationships for in-scope matters, including as-sent version resolution and explicit relationship bindings. - Reproducible export. The system MUST produce reproducible exports with manifests, hashes, and complete exception traceability (see Section 3.8).

The operating model determines when and how evidence enters the Preservation System of Record. The System of Record determines what properties that evidence carries once preserved.

5.2 Preservation Postures

Enterprises typically operate under one or more preservation postures. Each posture has distinct architectural properties that affect Reconstruction-Grade conformance. A conforming implementation MUST document which posture or combination of postures is in use and MUST satisfy all applicable requirements from Sections 3 and 4 regardless of posture.

Posture A - Preserve-in-Place (Policy-Based Holds)

Content remains within the collaboration platform under retention or legal hold policies that prevent deletion.

Architectural properties: - Minimal data movement; rapid activation. - Lower immediate storage duplication.

Reconstruction-Grade limitations: - Preserves existence but does not freeze point-in-time document state at event time. - Does not preserve explicit message ↔ link ↔ version bindings as first-class records. - Does not arrest audit log retention decay. - Scope is custodian-bounded; collaborative dependencies outside the custodian boundary MAY not be preserved.

Preserve-in-Place is an important compliance control. It is not, by itself, a Reconstruction-Grade preservation mechanism. A system relying solely on Preserve-in-Place MUST document which Reconstruction-Grade requirements (Section 3, Appendix B) are not satisfied and MUST NOT claim conformance at any level unless all applicable MUST requirements are met through supplementary mechanisms.

Posture B - Collect-to-Preserve (Reconstruction-Focused Baseline)

Matter-scoped preservation into a dedicated Preservation System of Record with deterministic resolution and explicit relationship bindings.

Architectural properties: - Preserves point-in-time document state with deterministic as-sent version resolution. - Maintains version lineage and stable identifiers. - Preserves explicit relationship bindings (message ↔ link ↔ file ↔ version). - Records exceptions deterministically with reason codes and retry histories. - Enables reproducible export from the Preservation System of Record.

Tradeoffs: - Requires scoped ingestion effort per matter. - Requires architectural discipline in scope definition and preservation orchestration.

Collect-to-Preserve aligns preservation effort directly to legal relevance. Scope is defined by matter context, custodian and repository relevance, and time bounds. Preservation then captures precisely what is required with deterministic version and relationship fidelity. This approach avoids the common anti-pattern of requiring multi-year tenant backfill before legal teams can rely on the system.

Collect-to-Preserve is the recommended baseline posture for Reconstruction-Grade conformance because it satisfies all core architectural requirements (Sections 3 and

4) by design, while scaling preservation in proportion to legal demand rather than total tenant size.

Posture C - Always-On Capture (Defined Populations)

Continuous preservation for designated high-risk roles, repositories, or custodian populations.

Architectural properties: - Reduces context-decay risk for designated cohorts by preserving evidence before a matter arises. - Enables incremental footprint expansion governed by policy. - Requires governance clarity: which populations, which artifact types, which retention depth.

Always-On Capture is often deployed in combination with Collect-to-Preserve. The combination allows enterprises to maintain standing preservation for high-risk populations while using matter-scoped collection for all other custodians and repositories.

Posture D - Full-Tenant Archive

Enterprise-wide continuous capture intended to support broad searchability across all users and repositories.

Architectural properties: - Broadest preservation coverage when fully operational. - MAY be justified when regulatory or operational mandates require comprehensive, standing capture.

Reconstruction-Grade constraints: - Highest storage and indexing footprint, amplified non-linearly by version history, metadata normalization, and resiliency requirements. - Longest lead time to complete initial backfill; time-to-operational-state is bounded by platform service limits and ingestion constraints (see Section 5.3), not by archive platform capacity. - Before ingestion stabilizes, legal teams may be unable to rely on the archive for active matters, creating a gap between deployment and defensible time-to-use.

A Full-Tenant Archive that satisfies all Reconstruction-Grade requirements is conforming. However, enterprises SHOULD evaluate whether the time-to-defensible-outcome under this posture meets operational needs relative to Collect-to-Preserve or hybrid approaches.

5.3 Enterprise Scale Constraints

All preservation postures operate under the same structural constraints imposed by collaborative cloud platforms. These constraints are inherent properties of the operating environment. A conforming implementation MUST document how it addresses each constraint class.

Platform service limits. Microsoft 365 and similar platforms impose API throttling, rate limits, and concurrency ceilings that bound effective sustained throughput. Time-to-ingest for any preservation posture is governed by these limits, not by archive

platform capacity. Implementations MUST account for throttling in capacity planning and MUST NOT assume unthrottled access.

Version proliferation. SharePoint and OneDrive version history, metadata normalization, relationship indexes, and resiliency requirements (e.g., geo-redundant immutable storage) can multiply the preserved footprint non-linearly relative to the latest-version corpus. Implementations SHOULD document the effective version multiplier and total footprint assumptions for capacity planning.

Permissions boundaries. Preservation actions require application-level permissions that may not be granted uniformly across all repositories. Permission remediation cycles are a common source of delay and exception volume. Implementations MUST treat permission-denied conditions as structured exceptions (see Section 3.7) and MUST provide remediation workflows.

Audit retention windows. Audit logs in Microsoft 365 are subject to retention limits that vary by license tier and configuration. Once audit records age out, behavioral evidence is permanently lost. Implementations that depend on audit evidence (RG-Plus and above) MUST document the effective audit retention window and MUST preserve audit records within that window. The system MUST represent audit coverage bounds explicitly (see Section 3.6).

Network and egress constraints. Large-scale data movement is bounded by network bandwidth, egress costs, and geographic data residency requirements. These constraints affect time-to-completion for any preservation posture and are particularly acute for Full-Tenant Archive deployments.

5.4 Time-to-Defensible-Outcome as the Governing Metric

The critical operational metric for a Reconstruction-Grade preservation operating model is not total volume preserved. It is time-to-defensible-outcome: the elapsed time from matter trigger to a reproducible, export-ready evidentiary record that satisfies all applicable conformance requirements.

This metric reflects a practical reality: legal teams cannot defer active matters until a preservation system reaches full operational state. A conforming operating model SHOULD be evaluated against the following: - Time-to-first-defensible-export. How quickly can the system produce a conforming export for a new matter? - Scope-to-preservation latency. What is the elapsed time from scope definition to completed deterministic capture? - Exception resolution cadence. How quickly are structured exceptions triaged and remediated? - Incremental matter onboarding. Can additional matters be onboarded without waiting for tenant-wide backfill to complete?

Enterprises SHOULD select the preservation posture (or combination) that minimizes time-to-defensible-outcome for their risk profile and regulatory obligations.

5.5 Throttling, Retries, and Deterministic Outcomes

At enterprise scale, failures are normal: permission errors, API throttling backoff, transient service faults, file moves and renames, and object lifecycle events are expected operating conditions. Reconstruction-Grade systems MUST treat these conditions as first-class workflow states.

The standard expectation is not perfection. It is determinism and auditability. Every referenced object MUST resolve to one of the following deterministic end states: - Preserved content. The object was successfully collected with stable identifiers, version metadata, and relationship bindings. - Structured exception record. The object could not be collected. The system has recorded the original reference, stable identifiers (if resolvable), a normalized reason code, complete retry history, and a deterministic end-state classification.

There are no silent drops. When preservation outcomes are explicit and reproducible, defensibility becomes an architectural property of the system rather than a function of operator narrative.

6. Measurable Evaluation Framework

A new category only matters if it can be evaluated. This section provides a practical framework for assessing whether a platform is Reconstruction-Grade.

6.1 Evaluation Categories

Point-in-time resolution Can the system deterministically resolve document state at or before an event timestamp (e.g., message send time)?

Stable identifiers Does the system preserve platform-native IDs sufficient to re-resolve objects despite URL changes, moves, or renames?

Relationship integrity Does the system preserve explicit message ↔ link ↔ file ↔ version relationships and export them without collapsing context?

Identity over time Can the system reconstruct role, department, manager, and group membership as-of a historical date?

Behavior evidence Does the system ingest audit evidence and correlate observed interaction to preserved objects and versions, with explicit boundedness?

Deterministic exceptions Are failures explicit, reason-coded, and auditable with retry histories and deterministic end states?

Reproducible exports Are exports repeatable with manifests, hashes, and complete traceability of scope and exceptions?

6.2 Conformance Levels

To support incremental adoption, this document defines three conformance levels.

RG-Core (Baseline Reconstruction-Grade) A system qualifies for RG-Core only if it satisfies deterministic point-in-time resolution, stable identifier preservation, relationship export integrity, deterministic exception handling, and reproducible export manifests.

RG-Plus (Identity + Behavior Conformance) RG-Plus adds effective-dated identity reconstruction and audit-evidence ingestion/correlation with explicit boundedness reporting.

RG-Max (Expanded Reconstruction Depth) RG-Max adds accessed-version analysis (where audit supports it), expanded artifact coverage (pages/lists/Loop/etc.), and advanced validation routines (referential integrity scoring, coverage gap alerting, and multi-profile export without semantic drift).

Enterprises may require RG-Core as the minimum and treat RG-Plus/RG-Max as maturity goals.

Note: Conformance levels are formally specified with full requirement mappings in Appendix B, Section B.0. The descriptions above are informative summaries.

6.3 Minimum Conformance Tests

Enterprises should require a small battery of conformance tests. Reconstruction-Grade claims should be demonstrable with controlled scenarios.

T-01 As-sent resolution Create a document with multiple versions; send a Teams/email link at t0; modify at t1; export must produce v≤t0.

T-02 Link canonicalization Use sharing links, redirect patterns, and renamed/moved objects; export must still resolve using stable IDs.

T-03 Relationship integrity Single message references multiple objects; export must preserve parent-child mappings and relationship types.

T-04 Identity as-of Change a user's department and group membership; as-of queries must reflect historical state.

T-05 Access evidence (bounded) Generate view/edit events; system must correlate audit events to objects/versions and report coverage bounds.

T-06 Exception determinism Force a permission denial; system must produce a structured exception record with retries and reason code.

T-07 Export reproducibility Run export twice; manifests and hashes must match and exceptions must be stable and explainable.

7. Standard Governance and Adoption

Reconstruction-Grade eDiscovery is not a product. It is a standard of evidence architecture. The goal is to define measurable criteria that enterprises, service providers, and technology vendors can evaluate and implement independently.

7.1 Why a Standard Is Needed

Collaborative evidence has outgrown ad hoc interpretation. When each matter invents its own reconstruction logic, outcomes become inconsistent and defensibility becomes fragile. A standard provides shared definitions, shared test criteria, and a baseline for proportional reasonableness under scrutiny.

7.2 Standard Artifacts

The Reconstruction-Grade Standard comprises the following artifacts: - A normative requirements set (RGR series, Appendix B) defining what a Reconstruction-Grade system MUST preserve and produce. - A conformance test suite (Section 6.3) with repeatable scenarios covering modern attachments, identity drift, audit correlation, and exception determinism. - A reference data model (evidence graph) describing objects, relationships, and required metadata. - A reference export profile (Appendix C) specifying manifest structure, hashes, and relationship overlays aligned to downstream review systems. - An evaluation framework (Section 6 and Appendix G) for vendor and platform assessment.

7.3 Standard Versioning and Change Control

Collaboration workloads evolve. Teams, Loop, Viva, Copilot-era artifacts, and future Microsoft 365 workloads will introduce new artifact types, new relationship patterns, and new audit signals. The standard MUST evolve with them.

The following versioning principles apply: - Major versions (e.g., 1.0 → 2.0) indicate breaking changes to conformance requirements. Existing conformance claims MUST be re-evaluated. - Minor versions (e.g., 1.0 → 1.1) add requirements, artifact coverage, or conformance tests without invalidating existing claims. - Each version MUST

include a change log documenting additions, modifications, and deprecations. - Conformance claims MUST reference a specific standard version.

7.4 Participation and Validation

A limited number of enterprises are invited to participate in defining and validating the Reconstruction-Grade Standard through an architectural working group focused on measurable criteria for collaborative evidence preservation and reconstruction.

Founding participants will help validate requirements against real enterprise constraints (scale, throttling, identity systems, regulatory obligations) and contribute to a conformance test suite that reflects the operational reality of modern collaboration environments.

Vendors and service providers MAY also participate. Conformance claims - by any party - SHOULD be evaluated against the published requirements and test suite. The purpose is shared vocabulary, measurable baselines, and repeatable evaluation.

Appendix A: Glossary

Collaborative evidence: Evidence created and used in cloud-native collaboration environments where artifacts are link-based, shared, and continuously revised.

Context gap: The gap between the artifact (file/message) and the surrounding reality required to answer what actually happened.

Reconstruction: The process of deterministically resolving what actors experienced at a specific time, including document state, identity state, and access behavior.

Evidence graph: A structured representation of evidence objects and their relationships with stable identifiers, timestamps, and provenance.

Collect-to-Preserve: An operating model where preservation is triggered by legal relevance and scope, rather than full-tenant archive-first backfill.

Deterministic end state: A completed workflow where all references resolve or are captured as structured exceptions, with no silent drops.

Appendix B: Reconstruction-Grade Requirements (RGR)

This appendix defines the normative requirements for Reconstruction-Grade conformance.

A system may claim conformance only if it satisfies all MUST requirements applicable to the declared conformance level.

B.0 Conformance Levels

To support incremental adoption while maintaining architectural rigor, this standard defines three conformance levels.

RG-Core (Baseline Reconstruction-Grade)

A system qualifies as RG-Core only if it satisfies all MUST requirements in the following domains:

Deterministic document state resolution

Stable identifier preservation

Explicit relationship preservation

Deterministic exception handling

Reproducible exports with manifest and hashes

Immutable preservation and scope ledger

RG-Core establishes deterministic reconstruction without requiring advanced identity or audit correlation.

RG-Plus (Identity + Behavior Conformance)

RG-Plus requires all RG-Core requirements plus:

Effective-dated identity reconstruction

Historical membership reconstruction (bounded by available data)

Audit ingestion and correlation

Explicit differentiation between potential access and observed access

RG-Plus enables defensible “who was responsible” and “who saw what” claims within bounded audit coverage.

RG-Max (Expanded Reconstruction Depth)

RG-Max requires RG-Plus plus:

Accessed-version correlation where audit permits

Expanded artifact coverage (pages, lists, Loop, etc.)

Referential integrity validation

Multi-profile export without semantic drift

Operational coverage metrics and gap reporting

RG-Max represents mature, enterprise-scale Reconstruction-Grade programs.

B.1 Identity Over Time (RGR-ID)

Code	Level	Requirement	Verification
RGR-ID-001	RG-Plus MUST	Model individuals as natural persons independent of transient directory identifiers (e.g., UPN changes).	Demonstrate identity correlation across renames and account lifecycle changes.
RGR-ID-002	RG-Plus MUST	Preserve effective-dated identity snapshots from authoritative sources and record provenance (source system, snapshot time).	Show as-of reconstruction for a user across role/department changes.
RGR-ID-003	RG-Plus MUST	Support as-of queries for role, department, manager, status, and key scoping attributes.	Run queries for date X and date Y and produce differing outputs consistent with change history.

Code	Level	Requirement	Verification
RGR-ID-004	RG-Plus SHOULD	Preserve historical group membership and role-based access membership as-of specific dates.	Demonstrate group membership timeline and as-of resolution.
RGR-ID-005	RG-Plus MUST	Persist identifiers required to link identity records to collaboration artifacts and retain historical mapping.	Demonstrate linkage between message actor and identity state as-of event time.
RGR-ID-006	RG-Plus SHOULD	Maintain identity drift audit (UPN, display name, manager, department, employment status changes).	Produce identity change log and show reconstruction use.
RGR-ID-007	RG-Core MUST	Record custodian and repository scoping decisions with timestamps and approver identifiers.	Provide immutable decision ledger entries for scope changes.
RGR-ID-008	RG-Max MAY	Ingest HR and directory data to support “hidden custodian” discovery workflows.	Demonstrate behavioral signals suggesting additional actors.

B.2 Audit and Behavior Evidence (RGR-AU)

Code	Level	Requirement	Verification
RGR-AU-001	RG-Plus SHOULD	Ingest audit records relevant to collaboration behavior (view, edit, share, access) and treat them as evidence.	Demonstrate ingestion pipeline and immutability controls.

Code	Level	Requirement	Verification
RGR-AU-002	RG-Plus MUST	Correlate audit events to preserved objects using stable identifiers, not URLs.	Demonstrate correlation for moved/renamed items.
RGR-AU-003	RG-Plus MUST	Preserve audit evidence with provenance (source, retrieval time, query parameters).	Provide audit ingestion manifest and reproducibility documentation.
RGR-AU-004	RG-Plus SHOULD	Support audit preservation strategies aligned to retention windows to avoid context loss.	Demonstrate capture-before-expiry and coverage reporting.
RGR-AU-005	RG-Plus MUST	Explicitly differentiate potential access (permissions) from observed access (audit).	Show separate fields and explainability in export.
RGR-AU-006	RG-Plus MUST	Represent audit-based claims with explicit coverage bounds (time range, licensing, availability).	Demonstrate “unknown” classification when audit coverage is incomplete.
RGR-AU-007	RG-Core MUST	Maintain immutable audit trail of preservation triggers and actions.	Show end-to-end chain-of-custody across preserve and export.
RGR-AU-008	RG-Max SHOULD	Enable accessed-version analysis by correlating audit events to version timelines.	Demonstrate accessed-version output for controlled scenario.

B.3 Document State and Deterministic Resolution (RGR-DS)

Code	Level	Requirement	Verification
RGR-DS-001	RG-Core MUST	Preserve file bytes for each preserved version and compute cryptographic hashes.	Validate hash stability across repeated exports.
RGR-DS-002	RG-Core MUST	Persist version identifiers, timestamps, and lineage metadata.	Demonstrate version list with stable IDs.
RGR-DS-003	RG-Core MUST	Deterministically resolve an as-of version for modern attachments using event timestamp (latest version where lastModifiedDateTime ≤ message timestamp).	Demonstrate deterministic resolution rule.
RGR-DS-004	RG-Core MUST	Apply deterministic tie-breaker when multiple versions share same timestamp.	Show consistent tie-breaker behavior.
RGR-DS-005	RG-Core MUST	Record fallback rules when version history incomplete or unavailable.	Demonstrate explicit fallback documentation.
RGR-DS-006	RG-Core MUST	Canonicalize and resolve sharing links and redirects to underlying repository objects.	Demonstrate resolution across redirect scenarios.
RGR-DS-007	RG-Core MUST	Persist stable platform identifiers (siteId, driveId, itemId, listItemUniqueId, versionId) where applicable.	Inspect preserved metadata and re-resolve content after move.

Code	Level	Requirement	Verification
RGR-DS-008	RG-Plus SHOULD	Support preservation of full version lineage for in-scope repositories where feasible.	Demonstrate policy-based version capture.
RGR-DS-009	RG-Max MAY	Preserve additional metadata required for advanced filtering (createdBy, modifiedBy, file path history).	Demonstrate completeness in export metadata.
RGR-DS-010	RG-Core MUST	Support deterministic resolution even if referenced file is moved, renamed, permission-changed, or deleted within retention bounds.	Demonstrate stability via stable identifiers and exception records.

B.4 Relationship Integrity (RGR-RL)

Code	Level	Requirement	Verification
RGR-RL-001	RG-Core MUST	Preserve explicit message ↔ link ↔ file ↔ version bindings as first-class records.	Show relationship table with stable IDs.
RGR-RL-002	RG-Core MUST	Export relationships using explicit fields (ParentId/ChildId, RelationshipType) without re-attaching binaries.	Demonstrate export overlay reconstruction.

Code	Level	Requirement	Verification
RGR-RL-003	RG-Core MUST	Allow single preserved object to have multiple contextual bindings without collapsing events.	Demonstrate many-to-one event bindings.
RGR-RL-004	RG-Plus SHOULD	Preserve repository context (site/channel/team identifiers) per event binding.	Demonstrate contextual export metadata.
RGR-RL-005	RG-Core MUST	Preserve timestamps for relationship events (send time, share time, access time where available).	Show reconstructed timeline.
RGR-RL-006	RG-Max MAY	Preserve conversation threading relationships.	Demonstrate thread grouping and ordering.
RGR-RL-007	RG-Core MUST	Ensure stable, unique identifiers for every exported record (referential integrity).	Demonstrate referential integrity validation.
RGR-RL-008	RG-Max SHOULD	Support relationship integrity validation (detect broken or orphaned links).	Demonstrate validation report.

B.5 Export and Reproducibility (RGR-EX)

Code	Level	Requirement	Verification
RGR-EX-001	RG-Core MUST	Produce exports consisting of native files plus standard load/overlay files including provenance, relationships, and hashes.	Demonstrate export package compatibility.

Code	Level	Requirement	Verification
RGR-EX-002	RG-Core MUST	Include export manifest capturing counts, scope parameters, time ranges, tool versions, and exceptions.	Provide sample manifest.
RGR-EX-003	RG-Core MUST	Support reproducible exports (same scope definition → stable outputs and hashes, subject to new preservation events).	Run export twice and compare manifests/hashes.
RGR-EX-004	RG-Core SHOULD	Support resumable exports with retry and full auditability.	Demonstrate interruption and resumption.
RGR-EX-005	RG-Core MUST	Preserve and export exception records alongside content.	Show exception overlay records.
RGR-EX-006	RG-Max MAY	Support multiple export profiles without altering evidence graph semantics.	Demonstrate alternate schema profile.
RGR-EX-007	RG-Max SHOULD	Provide export validation routines (hash verification, referential integrity checks).	Demonstrate automated validation report.

B.6 Exception Determinism (RGR-ER)

Code	Level	Requirement	Verification
RGR-ER-001	RG-Core MUST	Generate structured exception record for any attachment or linked item that cannot be collected.	Demonstrate exception object creation.
RGR-ER-002	RG-Core MUST	Exception records MUST include original reference, normalized reason code, retry history, and timestamps.	Inspect exception schema and audit trail.
RGR-ER-003	RG-Core MUST	Exception records MUST remain associated with parent communications and intended targets.	Demonstrate linkage visibility in export.
RGR-ER-004	RG-Core SHOULD	Implement controlled backoff and bounded retry policies for transient failures.	Demonstrate throttling retry behavior.
RGR-ER-005	RG-Core SHOULD	Enable reprocessing after remediation while preserving attempt history.	Demonstrate re-queue and updated tracking.
RGR-ER-006	RG-Core MUST	Reach deterministic end state for every preservation job (success or explicit failure).	Demonstrate job completion criteria and reporting.
RGR-ER-007	RG-Max MAY	Provide exception analytics to quantify risk and remediation priority.	Demonstrate exception dashboards.

Conformance Declaration

A system claiming Reconstruction-Grade conformance MUST:

Declare its conformance level (RG-Core / RG-Plus / RG-Max).

Satisfy all MUST requirements for that level.

Provide demonstrable evidence via minimum conformance tests.

Provide documentation describing boundedness where upstream evidence (e.g., audit logs) is incomplete.

Appendix C: Export Manifest and Chain-of-Custody Profile

A Reconstruction-Grade export is not merely a file transfer. It is a reproducible evidentiary transaction. The manifest is the bridge between scope definition and downstream defensibility.

Table 8. Example export manifest fields

Field	Purpose
ManifestId	Unique identifier for the export job and its manifest artifact.
GeneratedAt (UTC)	Timestamp when the export package was assembled.
ScopeDefinitionId	Identifier referencing the immutable scope definition (custodians, repositories, time bounds, queries).
ScopeParameters	Human-readable summary of scope parameters (matter identifier, date range, filters).
SourceWorkloads	Workloads included (e.g., Exchange, Teams, SharePoint, OneDrive) and their coverage windows.
ResolutionPolicy	Document state resolution policy (as-sent rule, tie-breakers, fallback rules).
HashAlgorithm	Hash algorithm used for files and records (e.g., SHA-256).
ExportProfile	Export schema profile used (load file formats, relationship representation).
Counts	Counts of exported parent items, child items, versions, relationship records, audit records.
ExceptionsSummary	Counts by reason code; list of high-severity gaps; pointer to exception overlay.

Field	Purpose
ToolVersion	Tool and schema versions used to generate export (supports reproducibility).
Operator	Actor identity who initiated the export and approvals (if applicable).
IntegrityChecks	Results of referential integrity checks and hash verification.
ReproducibilityNotes	Notes required to reproduce (queries, paging tokens, snapshot identifiers).

Export packages SHOULD include the manifest as a signed or otherwise integrity-protected artifact alongside the native files and load/overlay files.

Appendix D: Reconstruction Scenarios

The following scenarios illustrate why Reconstruction-Grade capabilities are not theoretical. They are the minimum to answer modern questions about behavior, reliance, and timing.

D.1 Hyperlinked Attachment After-the-Fact Edits

Question:

What exact document state informed a decision communicated at time t0?

What is required to answer: - Message timestamp (t0) and immutable message content - Stable resolution of linked object and full version timeline - Deterministic as-sent version selection rule - Preserved bytes and version identifier for the resolved version

What legacy models produce: - Current file bytes (time-shifted) - No explicit binding between message and specific version - Reviewer interpretation of “closest version”

What a Reconstruction-Grade record produces: - As-sent version bytes (latest version where lastModifiedDateTime ≤ t0) with versionId - Explicit message ↔ link ↔ file ↔ version binding exported and reproducible - Manifest documenting resolution policy and tie-breakers

D.2 Identity Drift and Historical Responsibility

Question:

Who was responsible for a repository during the relevant period, and who approved access?

What is required to answer: - Effective-dated identity snapshots and change history - Historical group membership and access assignment events - Repository ownership and governance metadata as-of period

What legacy models produce: - Current directory attributes substituted for historical state - Org-chart interviews used as primary evidence - Ambiguous mapping between person and role at the time

What a Reconstruction-Grade record produces: - As-of identity record (department, manager, role) with provenance - As-of group membership and access events tied to the natural person - Defensible explanation without narrative substitution

D.3 Permissions vs Observed Access

Question:

Did a person actually access a sensitive document during the relevant period?

What is required to answer: - Audit evidence showing open/view/edit events and timestamps - Correlation between audit events and preserved object identifiers - Version timeline to determine which version was accessed

What legacy models produce: - Permissions lists used as proxy for access - No proof of actual behavior - Inability to differentiate exposure from interaction

What a Reconstruction-Grade record produces: - Observed access evidence ("who saw what, when") correlated to the preserved object - Accessed-version analysis tied to preserved version identifiers - Clear separation of potential access vs observed interaction

D.4 Broken Links, Redirects, and File Moves

Question:

Can you still produce the evidence when links change or objects move?

What is required to answer: - Canonicalization of sharing links and redirects - Stable platform identifiers (siteId/driveId/itemId/versionId) - Evidence of resolution at ingest time plus re-resolution capability

What legacy models produce: - Brittle URL-based capture that breaks on rename/move - Silent failures where linked content is not collected - No record of what was attempted or why it failed

What a Reconstruction-Grade record produces: - Stable resolution independent of URL changes using platform IDs - Structured exception records when resolution fails, with reason codes and retries - Deterministic end state and auditability for every link

D.5 Export Reproducibility Under Scrutiny

Question:

Can you reproduce the same export later and defend chain-of-custody?

What is required to answer: - Immutable scope definition and decision ledger - Export manifest with counts, hashes, tool versions, and exceptions - Resumable export mechanism with full audit trail

What legacy models produce: - Ad hoc exports that depend on current state and operator judgement - Missing manifests or incomplete hashing - Inability to explain discrepancies across export attempts

What a Reconstruction-Grade record produces: - Reproducible exports with matching manifests and hash evidence - Full audit trail for scope decisions and exceptions - Integrity verification and referential integrity checks

Appendix E: Operational Metrics for Reconstruction-Grade Programs

Reconstruction-Grade programs benefit from operational metrics that quantify coverage, risk, and time-to-use.

Table 9. Example operational metrics

Metric	Why It Matters
Context coverage	Percent of modern attachments resolved to a preserved as-sent version; percent with stable identifiers captured.
Exception rate	Percent of referenced objects that resulted in exceptions; breakdown by reason code.
Audit correlation coverage	Percent of preserved objects with correlated audit evidence in relevant windows.
Identity coverage	Percent of custodians with effective-dated identity history available for relevant periods.
Reproducibility score	Percent of exports that reproduce identical manifests and hashes when rerun under same scope definition.
Time-to-use	Elapsed time from matter trigger to first usable export (not time-to-backfill).
Scope precision	Ratio of preserved-to-reviewed volume; trends over time as identification improves.

These metrics support continuous improvement and create an evidence-based operating discipline, rather than a one-off collection exercise.

Appendix F: Collaboration Artifact Taxonomy and Context Dependencies

Reconstruction-Grade evidence preservation requires workload-specific understanding of what constitutes an artifact, how it is identified, and which relationships are required to preserve meaning. This appendix provides a practical taxonomy for Microsoft 365-centric environments.

Table 10. Artifact taxonomy and context dependencies

Workload	Artifact Type	Primary Identifiers	Required Relationships	Reconstruction Risk If Missing
Exchange	Email message (legacy attachment)	InternetMessageId, MessageId	Parent/child attachment records, recipients, folders	Typically self-contained bytes; primary risk is metadata loss or chain-of-custody gaps.
Exchange	Email message (modern attachment / link)	InternetMessageId, MessageId	Message ↔ link ↔ driveItem ↔ version binding	High risk of time-shifted content if as-sent resolution is not preserved.
Exchange	Mailbox folder hierarchy	FolderId	Containment and lifecycle (moves)	Scoping and provenance can be challenged if location context is lost.

Workload	Artifact Type	Primary Identifiers	Required Relationships	Reconstruction Risk If Missing
Teams	1:1 or group chat message	chatId, messageId	Edits/deletes, reactions, thread context	Messages can be edited; loss of edit history can change meaning.
Teams	Channel message	teamId, channelId, messageId	Threading, edits/deletes, mentions	Channel context and membership affect interpretation of dissemination.
Teams	Message edit / delete events	messageId	Audit events or native revision history	Without history, the record may represent a later state, not what was seen.
Teams	Meeting chat	meetingId, messageId	Participants as-of meeting time	Identity-over-time and participant lists are often disputed.
Teams	Shared file via link	messageId + driveItem IDs	As-sent version, access evidence	Common reconstruction failure point: latest file substituted for as-sent.
SharePoint	Document library file	siteId, driveId/itemId	Version lineage, permissions, sharing links	Version retention and link canonical-ization are critical.
SharePoint	List item	siteId, listId, listItemU-niqueId	Field history, attachments, workflow events	List items can drive decisions; missing history breaks recon-struction.
SharePoint	Page / news post	siteId, pageId	Version history, authorship	Pages are frequently edited; final state may be misleading.

Workload	Artifact Type	Primary Identifiers	Required Relationships	Reconstruction Risk If Missing
OneDrive	User file	driveId, itemId, versionId	Sharing events, moves/renames	High churn; user lifecycle can orphan content if IDs not preserved.
OneDrive	Sharing link / invite	shareId / link token	Canonicalization to driveItem	Brittle URL capture breaks under link changes.
Viva Engage	Post / comment	threadId, messageId	Reactions, edits, audience scope	Audience and membership context are required for dissemination analysis.
Loop	Loop component	componentId (varies)	Embedding context across Teams/Outlook	Component is distributed; preservation requires capturing object + embedding bindings.
Planner/Tasks	Task item	planId, taskId	Assignees, comments, attachments, timestamps	Task history can be central to intent and timing; static capture loses evolution.
Forms	Form response	formId, responseId	Responder identity, timestamps	Identity attribution and timing can be contested without provenance.
Stream	Video	videoId	Permissions, views, transcript versions	Behavior evidence (views) and permission changes matter for access analysis.

Workload	Artifact Type	Primary Identifiers	Required Relationships	Reconstruction Risk If Missing
Entra ID	Group membership	groupId, memberId	Effective-dated membership timeline	Static snapshots misrepresent who had access at time X.
Audit	M365 Unified Audit Log event	recordId	Correlation keys to workload objects	If audit ages out, ‘who saw what’ becomes unprovable.
Compliance hold/policy layer (conceptual)	Hold / retention policy event	policyId	Scope decisions and preservation triggers	Policy context is necessary to defend preservation posture.

Note: Identifier names vary by workload and API. The standard requirement is not a specific field name; it is preservation of stable identifiers sufficient to re-resolve objects and to correlate events deterministically.

Appendix G: Questions to Ask Vendors

When evaluating platforms for Reconstruction-Grade conformance, enterprises should require substantive answers to the following questions. These questions are designed to distinguish systems that preserve collaborative evidence with reconstruction-grade fidelity from those that flatten it into files and messages that cannot support legal reasoning. - When a message contains a modern attachment or link, can you export the as-sent version (latest version where lastModifiedDateTime ≤ message timestamp)? - Do you preserve version identifiers and the actual file bytes for resolved versions? - What stable identifiers do you persist (siteId, driveId, itemId, listItemUniqueId, versionId), and how do you canonicalize sharing links and redirects? - Can you preserve and export explicit ParentId/ChildId relationship mappings for messages and linked content? - Can you reconstruct group membership and identity attributes as-of date X, or do you rely on current directory state? - Do you treat audit logs as evidence, and can you correlate observed access to preserved versions - with explicit coverage bounds? - What is your exception model: do failures produce reason-coded records with retry history, or do failures silently drop evidence? - Are exports reproducible? If we run the same scope twice, do we get the same set with the same hashes and manifests (subject to new preservation events)? - How do you support interruption handling and resumption for long-running exports? - How do you record scope decisions so proportionality arguments are supported by an immutable decision ledger?

Appendix H: Vendor Scoring Worksheet

This worksheet provides a structured way to score solutions against Reconstruction-Grade criteria. Scores are illustrative; enterprises should adjust weights based on risk profile.

Table 11. Scoring worksheet (fill-in)

Criterion	Weight (1-3)	Vendor Score (0-3)	Notes / Evidence
Point-in-time as-sent resolution	3		Demonstrated deterministic resolution and documented fallback rules.
Stable identifier preservation	3		Persists platform IDs; resilient to moves/renames/redirects.
Relationship integrity (Parent/Child mappings)	3		Exports explicit relationships without collapsing context.
Identity over time (effective-dated)	3		As-of identity and membership reconstruction.
Audit evidence ingestion and correlation	2		Observed access correlated to preserved objects/versions.
Deterministic exceptions and retry audit	2		Reason-coded exceptions; full retry history; deterministic end state.
Reproducible exports with manifests + hashes	3		Repeatable outputs; integrity validation.

Criterion	Weight (1-3)	Vendor Score (0-3)	Notes / Evidence
Operational monitoring at scale	2		Throughput, throttling backoff, exception tracking, SLA reporting.
Data portability and evidence longevity	2		Supports export profiles and long-term reusability.

Appendix I: Deterministic Resolution and Processing Profiles

This appendix provides non-normative algorithm sketches that illustrate deterministic processing. Implementations will vary, but outcomes must be reproducible.

I.1 As-Sent Version Resolution (Conceptual)

Inputs: messageTimestamp, linkReference

1) Resolve linkReference to underlying object using canonicalization + platform APIs
2) Enumerate object versions with (versionId, lastModifiedDateTime)
3) CandidateVersions = { v | v.lastModifiedDateTime <= messageTimestamp }
4) If CandidateVersions is empty:

Record fallback rule; emit exception or select earliest available per policy

Else:

Select v* = argmax(CandidateVersions.lastModifiedDateTime)

If ties on timestamp: select highest versionId per deterministic ordering

5) Preserve bytes(v*), metadata(v)*, versionId(v*)
6) Persist binding: (messageId -> link -> objectId -> versionId) with provenance

I.2 Exception and Retry Handling (Conceptual)

When a linked object cannot be collected:

- Create ExceptionRecord { originalReference, resolvedIds, reasonCode, attempts[] }
- attempts[] includes { attemptTime, outcome, errorDetail, backoffApplied }

- Retry bounded times for transient conditions (e.g., throttling)
- Mark deterministic end state when retries exhausted
- Allow reprocessing after remediation while preserving attempt history

Appendix J: Exception Taxonomy and Operational Playbooks

Exceptions are not edge cases at enterprise scale. A Reconstruction-Grade program treats exception handling as an operational discipline.

Table 12. Exception reason code taxonomy

Reason Code	Meaning	Recommended Response
PermissionDenied	Access to target object not granted at time of collection.	Coordinate permission remediation; reprocess; record approvals.
ItemDeletedOrOutsideRetention	Target object no longer exists or is beyond retention bounds.	Document loss; preserve link evidence; capture secondary sources if any.
Throttling	Service limits triggered; requests slowed or denied.	Apply controlled backoff; scale concurrency; extend job windows.
TransientServiceError	Temporary fault (timeouts, 5xx, network).	Retry with backoff; monitor systemic outage patterns.
UnresolvableLink	Link could not be canonicalized to a stable object.	Preserve original reference; attempt alternate resolution methods; flag for review.
UnsupportedArtifact	Artifact type not yet supported by preservation pipeline.	Record as explicit limitation; define roadmap or alternate capture method.
Unknown	Unexpected failure with insufficient classification.	Capture full error context; triage; refine reason codes over time.

Operational note: exception rates should be reported per matter and per workload, with trends and remediation backlog tracked as a program metric.

Appendix K: Enterprise Implementation Roadmap

Reconstruction-Grade adoption is best approached as a program, not a single ingestion project. The goal is to establish a Preservation System of Record that delivers immediate matter value while expanding coverage over time.

Table 13. Illustrative implementation roadmap

Phase	Theme	Deliverables (illustrative)
Phase 0	IT and security alignment	Define roles, permissions, key management, audit access, and operational controls; establish decision ledger and governance.
Phase 1 (0-6 months)	Foundational Collect-to-Preserve + core exports	Deliver end-to-end matter workflows (Identify → Preserve → Search → Export) for custodial repositories; establish export profiles, manifests, hashes, and exception handling.
Phase 2 (6-12 months)	Modern attachments depth + Teams export maturity + version policies	Expand deterministic resolution for modern attachments at scale; mature Teams coverage; define version retention and preservation policy choices by risk.
Phase 3 (12-18 months)	Historical membership + metadata lineage maturity	Increase identity-over-time fidelity (historical group membership); improve metadata lineage; strengthen relationship integrity validation.

Phase	Theme	Deliverables (illustrative)
Phase 4 (18-24 months)	Emerging artifacts and expanded workload coverage	Address emerging collaboration artifacts (Loop, Copilot-era objects) and extend coverage across additional workloads while maintaining conformance tests.

K.1 Program Guardrails

- Start with matters: success is measured by time-to-usable export for real cases, not by total backfilled volume.
- Preserve context early: prioritize audit and identity pipelines to prevent context decay.
- Make limitations explicit: unsupported artifacts and gaps MUST be recorded as structured exceptions, not hidden.
- Instrument everything: measure throughput, throttling behavior, exception rates, and reproducibility.
- Treat scope as evidence: maintain an immutable ledger of what was included/excluded and why.

K.2 Practical Pilot Design

A pilot should be designed to prove Reconstruction-Grade properties with controlled scenarios and one or two real matters. A minimal pilot typically includes modern attachment resolution, identity as-of queries, audit correlation, and reproducible exports with manifests. 1. Select two matters with known hyperlink usage and meaningful timeline questions. 1. Define conformance tests (Appendix G) and success metrics (Appendix E). 1. Run parallel: compare legacy export results to Reconstruction-Grade outputs for as-sent versions and relationship integrity. 1. Document scope decisions and exceptions from day one; validate reproducibility with repeated exports. 1. Review results with litigation support and compliance architects; incorporate findings into standard requirements.

Appendix L: Collect-to-Preserve Matter Playbook

Collect-to-Preserve is an operating discipline. The playbook below describes what 'good' looks like when a matter triggers preservation.

L.1 Matter Trigger and Scope Definition

- Create an immutable scope definition: custodians, repositories, time bounds, and rationale.
- Resolve custodian identities to natural persons; capture effective-dated identity snapshots for the relevant period.
- Identify collaboration spaces where work occurred (teams, channels, sites) using evidence-based signals where possible.

L.2 Preservation Actions

- Preserve custodial repositories and relevant shared repositories within scope.
- Enumerate and preserve version lineage for in-scope objects per policy.
- Resolve modern attachments and links referenced by communications; preserve as-sent versions and relationship bindings.

L.3 Validation and Exception Triage

- Run referential integrity checks: every message-linked reference resolves or produces an exception record.
- Review exception dashboard; prioritize remediation for permission-denied and throttling classes.
- Lock preservation artifacts as immutable evidence once validated.

L.4 Export and Reproducibility

- Generate export package with manifest, hashes, load/overlay files, and exception overlays.

- Verify reproducibility by rerunning export under identical scope definition and comparing manifests/hashes.
- Provide downstream teams with relationship mappings and provenance fields to avoid re-flattening context.

Appendix M: Further Reading

The following materials provide additional context on collaborative evidence challenges and the motivation for Reconstruction-Grade standards. - Introducing Context-Aware eDiscovery: Why Modern Data Requires a Modern Approach - Jan 27, 2026 (https://www.cloudficient.com/blog/introducing-context-aware-ediscovery-why-modern-data-requires-a-modern-approach) - The Context Gap: Why Traditional eDiscovery Can't Explain What Really Happened - Jan 30, 2026 (https://www.cloudficient.com/blog/the-context-gap-why-traditional-ediscovery-cant-explain-what-happened) - Why Identification is the Nexus of Defensibility in eDiscovery - Feb 6, 2026 (https://www.cloudficient.com/blog/why-identification-is-the-nexus-of-defensibility-in-ediscovery) - Why Inference Is Not Defensibility: The Case for Reconstruction-Grade eDiscovery - Feb 13, 2026 (https://www.cloudficient.com/blog/why-inference-is-not-defensibility-the-case-for-reconstruction-grade-ediscovery) - Why Microsoft 365 Breaks Traditional eDiscovery Assumptions - Feb 18, 2026 (https://www.cloudficient.com/blog/why-microsoft-365-breaks-traditional-ediscovery-assumptions) - Closing the Context Gap: What Reconstruction-Grade Discovery Looks Like in Practice - Feb 26, 2026 (https://www.cloudficient.com/blog/closing-the-context-gap-what-reconstruction-grade-discovery-looks-like-in-practice)

Note: references are provided for conceptual background. The Reconstruction-Grade requirements in Appendix B are intended to be vendor-neutral and testable.