

Claude Enterprise Integration Guide

Anthropic Compliance API

The Above Anthropic Claude integration reads a person's out-of-browser Claude activity from your own Anthropic organization using the Anthropic Compliance API, covering claude.ai chats and Claude Code / Cowork session transcripts, so the Above insider-threat investigator can use it during an investigation. The data flow is one way: Above never writes to Claude, calls a delete endpoint, or requests a delete scope.

Document

Integration guide, v3

Updated

29 September 2026

Audience

Security and IT administrators

Contents

1. Overview
2. What Above Collects
3. What Above Stores
4. Stored evidence snapshots
5. Required Scopes
6. Prerequisites
7. Connecting
8. Summarizing a Claude File
9. When Content Is Not Available
10. Validation Checklist
11. Security and Least Privilege
12. Disconnecting
13. Out of Scope
14. Official references
15. AI agent identities created from this source

Overview

Nothing from this integration trains a model, not a vendor's, not a shared one, not Above's own. Your people's Claude prompts are never used to train, fine-tune, or improve any model, and are never disclosed to a model vendor for training. What Above reads is processed inside Above's own AWS account, scoped to your company, and stays in your tenant's applicable US or EU region. The only places a model is involved at all are the short per-conversation summary and the file description an analyst can request, both described in *What Above Stores*.

The Above Anthropic Claude integration reads a person's out-of-browser Claude activity from your organization's own Anthropic account, using the Anthropic **Compliance API**, so the Above insider-threat investigator can fold it into an investigation. It covers claude.ai chats and the session transcripts produced by Claude Code, Cowork, and remote Cowork, and it attributes each of them to an Above identity by email address.

The data flow is **one way**, and it is **read only**. Claude activity flows from your Anthropic organization into Above. Above never writes to Claude, never modifies or deletes a chat, a session, a file, or any Anthropic configuration, and never calls a Compliance API DELETE endpoint. The delete scope that would make such a call possible is deliberately never requested, so a compromise of Above cannot destroy your own compliance record.

From the Compliance API, Above keeps what the person typed and a record of each **tool call** Claude made on their behalf. Claude's reply text, thinking content, and system prompts are never stored. A tool call's own arguments and result text are kept only under the same capture decision as the typed prompts. The **metadata** of files attached to or generated in a conversation is kept, and an analyst can explicitly ask Above to describe one such file's contents. The file is read once and never stored. See *What Above Stores*.

Auth model Compliance Access Key A single API key sent in the x-api-key header, prefix sk-ant-api01-, created in claude.ai organization settings	Data direction One-way ingest Claude activity into Above; nothing is ever written back to Anthropic
Scopes Two read scopes read:compliance_user_data and read:compliance_activities; no delete scope	Used for Insider-threat investigations Out-of-browser Claude prompts, session context, and account activity
Coverage 30 days of chats, 7 days of sessions Reached back on first connection; file contents only when an analyst asks for a summary	

A Compliance Access Key is not an Admin API key

The two credentials look similar and are not interchangeable. A Compliance Access Key carries the prefix `sk-ant-api01-`. An Admin API key carries `sk-ant-admin01-`, has no compliance authority, and is refused by Above at connect time rather than stored and left to fail later.

What Above Collects

Above matches provider actor emails against existing identities' primary emails and aliases, with primary email taking precedence. The provider actor email is retained as attribution metadata. Conversations without a matched identity or stored messages are hidden, including from administrators and direct links. Collection continues under your capture policy. This does not create identities for directory-only users. Older unmatched conversations may remain hidden until a provider sync revisits them.

Anthropic Compliance API data Above reads

Data type	Purpose in Above
claude.ai chats and their messages	Read the conversation and its messages. Only the user's own text is retained.
Local Claude Code and Cowork session transcripts	Read the session and its transcript for sessions run on a person's own machine, for the same purpose.
Remote Cowork session transcripts	Read the session and its transcript for remotely hosted sessions. A remote session belongs either to a user or to an agent, and an agent-owned session can carry turns from several people.
Uploaded and generated file metadata	Read the file name, media type, size, and checksum where the provider reports them, so an analyst can see what was attached without Above holding the contents.
Account activity feed	Read, on demand while Above analyzes a person's activity, the account activity recorded for the analyzed time window. Activity Anthropic attributes to that person can become evidence on the incident, and the person's activity events the lookup returns are kept as stored evidence snapshots, within their limits, the evidence behind incidents and Insights (see Stored evidence snapshots). Activity from API keys and other automation is not attributed to any person, and Above never presents it as something the subject did.
File contents, on an analyst action only	Read the bytes of one uploaded or generated file when an analyst asks for a summary of it, capped at 25 MiB. The bytes are read once, described, and never stored.

How far back collection reaches:

- **Chats.** claude.ai chats updated in the previous **30 days** when you first connect, and every new or updated chat from then on.
- **Local sessions.** Claude Code and Cowork sessions run on a person's own machine are listed by their **last activity** (the Compliance API's `updated_at.gte` filter): on the first collection, every session active in the previous **seven days**; after that, every session active since the previous completed collection, with a 15-minute overlap so a session updated during a collection is not missed. A long-running local session is therefore re-read whenever it has new activity, however long ago it started.
- **Remote sessions.** Cloud Cowork sessions started in the previous **seven days**. The Compliance API offers no activity filter for remote sessions, so a remote session started more than seven days ago is not revisited, and messages added to it after that point are not collected.

Two further limits come from your Anthropic configuration rather than from Above. Claude Code sessions authenticated through **Amazon Bedrock**, **Google Vertex AI**, or a **Console API key** are outside this source entirely and never appear. And an organization configured for **HIPAA** does not have its local Claude Code sessions captured, so those lists are legitimately empty rather than failing.

What Above Stores

The boundary here is deliberate and narrow. Above stores the **verbatim text the person typed**, for the prompts your capture configuration admits, the metadata of the files involved, and a record of each tool call. Claude's own replies, its thinking, and the system prompt are read past and never written.

Tool calls. Every `tool_use` block in a chat or a session transcript (for example a web search, a file read, or an email sent through a connected MCP server) becomes one tool-call record, whether or not prompt text is captured. The record always holds: the tool name; the MCP server's host name, where Anthropic reports the server (claude.ai chats only); when it was invoked and its position in the conversation; whether the call succeeded, failed, or has no result yet; a coarse action class derived from the tool name (`read`, `mutate`, `egress` for a call that sends data out, or `other`); and, where the call's complete arguments name a destination or a resource under a well-known key (`to`, `url`, `file_path`, `document_id`, and similar), the destination's domain plus a **keyed digest** of the destination and of the resource. A keyed digest is an HMAC-SHA256 under a key derived for your company alone from a server-side secret that is never stored in the database, so it lets Above recognize the same address or file again without storing the address or path itself, and the same address never yields the same digest in two companies. Above sets no digest from arguments Anthropic truncated.

Each tool call is attributed to the person its conversation or session is attributed to. In an agent-owned remote session, which can carry turns from several people, a tool call belongs to the person whose prompt started the turn that made it, and to the person who started the session when Anthropic names no sender for that turn. The tool calls of a turn whose sender Above cannot match to exactly one of your identities are skipped with that turn's prompts, never filed under someone else.

The call's **arguments and result text** are stored only when the conversation's prompt text is captured under the rules below, and are not copied from a conversation Anthropic already reports as deleted or unavailable when Above reads it. A call first read while its prompt text was not captured gains its arguments and result text if a later read of the same conversation is captured; a later read that is not captured never removes what is already stored. When they are stored they are encrypted at rest: under your own tenant envelope key where Tenant Encryption is enabled for your workspace, and under Above's platform key otherwise. Above asks Anthropic for at most 65,536 characters (chats) or 65,536 bytes (session transcripts) of each tool input and each tool result; a longer one arrives cut short, and the record is marked truncated. A binary tool result, such as an image, is never stored. A session whose transcript holds more than 5,000 tool calls keeps the first 5,000 and is flagged as truncated rather than dropping the rest silently.

Where you see them: a Claude conversation's drawer in Shadow AI > Conversations lists its tool calls after its prompts, and each Claude surface (Claude Code, claude.ai chats, Cowork, or a Claude agent) appears in Shadow AI > AI Agents as a **Chat** row whose sessions list them. Both follow the conversation's own visibility: nothing is shown for a conversation Shadow AI would not show. Each view lists the first 500 tool calls in transcript order and says so when there are more; this display limit is separate from the 5,000 calls a session keeps.

You decide whether prompt text is kept at all, and the control is one you already own: **Business account prompt capture** under **Settings > Shadow AI**. Claude activity arrives with no personal-or-business classification, so it is always governed by the Business setting. Turning it off stops Above storing **any new Claude prompt text** from that point on. It acts forward only: text captured while it was on is retained rather than deleted, so use a deletion request to remove that history. Three further cases withhold text on their own, deliberately erring toward keeping less: a capture policy scoped by location withholds every prompt in the run, because a scheduled collection has no location to judge and Above will not invent one; a conversation Above could not attribute to a person is withheld whenever your policy narrows by person or risk, so the prompts nobody can attribute are never the ones that get kept; and a conversation Above did attribute, to an identity that no longer resolves in your company, is withheld rather than admitted on a subject that is no longer there. Where your policy narrows by neither person nor risk, a conversation Above could not attribute is still captured: nothing was excluded, and withholding there would drop evidence your own configuration admits. In each case the conversation itself is still recorded, with its times, counts, and file metadata; only the typed text is withheld.

Above never stores any of the following, from any of the four streams:

- Claude's responses, in whole or in part.
- Tool arguments and tool results, unless the conversation's prompt text is captured (see **Tool calls** above).
- Extended thinking content.
- System prompts.
- Images and other non-text message content.
- Artifact content.
- File contents. A file an analyst explicitly asks about is read once to produce a description; the description is kept, the file is not.

Local Claude Code session transcripts from the Compliance API also carry text Claude Code adds on the person's behalf. Above does not store it as a prompt: its system reminders, the expanded body of a slash command or skill (the typed command itself is kept), and its own background calls such as permission checks, status updates, next-prompt suggestions, and compaction. Claude Code sends the whole conversation again on every call; from the Compliance API, Above stores each typed prompt once, from the call that sent it. Recognizing Claude Code's own text is best-effort: text it adds in a shape Above does not yet recognize is kept as a prompt.

Data direction

One-way ingest from your Anthropic organization into Above. Above issues only GET requests against the Compliance API. It never creates, modifies, or deletes a chat, a session, a file, or any Anthropic configuration, and never calls a DELETE endpoint.

What Above reads

claude.ai chats and messages, local Claude Code and Cowork session transcripts, remote Cowork session transcripts, the metadata of files attached to or generated in them, and, on demand during an investigation, your organization's account activity for that investigation's time window.

What Above stores

The Compliance Access Key, encrypted at rest with AES-256-GCM: under your own tenant envelope key where Tenant Encryption is enabled for your workspace, and under Above's platform key otherwise. It is never returned

to a browser. Per conversation and per session: its times, its retained turn count, a summary when one was generated, and, where the provider reports them, the product surface it came from, the model it ran against, your Anthropic workspace id, and a deep link back to the transcript on Anthropic's side. Per tool call: the tool name, the MCP server host, times, outcome, action class, destination domain, and keyed (HMAC) digests of the destination and resource, plus the arguments and result text only when the conversation's prompt text is captured, encrypted at rest the same way the key is. Per admitted prompt: the verbatim user text, which message of the conversation that text came from, and whether Anthropic reported it as truncated. Per file: name, media type, size, checksum, and kind, each of which the provider may report as unknown. Plus, when an analyst has asked for one, a model-written description of the file's contents, encrypted the same way the key is. The file's own bytes are never stored. When an analyst cites account activity in an investigation, that activity is also kept with the incident as part of its evidence record. Prompt rows are held under Above's existing AI-chat storage policy, the same policy that governs the Gemini integration and browser-captured prompts. Everything is scoped to the connecting company.

What Above never does

Store Claude's responses, thinking content, system prompts, images, or artifact content. Store a tool call's arguments or result text for a conversation whose prompt text is not captured. Store the bytes of any file. Request a delete scope or call a delete endpoint. Read activity outside the time window of the investigation asking for it. Attribute account activity from API keys or other automation to a person. Fetch file bytes outside an explicit analyst request for a summary.

Tenant and identity scoping

Every read of this data is filtered to your company, and then filtered again by the per-user identity-visibility rules configured in Above. An analyst who cannot see a person elsewhere in the platform cannot see that person's Claude activity, list their Claude files, or ask for one to be summarized.

Your Claude prompts never train a model

Your Claude prompts do not train, fine-tune, or improve any model, a model vendor's or Above's own, and are never disclosed to a model vendor for training. Above may generate and store a short summary of a conversation and, when an analyst asks for one, a description of a file; both run on Amazon Bedrock, the AWS-hosted managed inference runtime Above operates in its own AWS account, against your tenant's data only and in the applicable US or EU region. Per [AWS's published Bedrock data-protection terms](#), and under Above's configuration, Amazon Bedrock does not use inputs or outputs to train models and model providers do not receive them; AWS's terms describe the operational exceptions (such as abuse monitoring and legal compliance) that apply to any provider. Summaries are intended to describe retained employee content, treating employee-authored instructions as part of the conversation. The conversation text and file list are treated as untrusted input and sanitized as the summarization prompt is rendered, before that prompt is sent to Bedrock. A conversation with no available text is recorded as having no content and is never sent to a model.

Stored evidence snapshots

When Above analyzes activity and queries Anthropic (Claude) Compliance live, the records the query returns are saved as evidence snapshots, up to the per-query and per-analysis limits under Bounds below: one row per record in Above's database, in the same deployment region as the rest of your company's data (US: `us-east-2`, EU: `eu-west-1`). The row is written when the query result arrives and holds, as JSON, the fields the Anthropic (Claude) Compliance API returned for that record; a text field over the size limit is truncated and the row is marked truncated. A record past those limits, or whose snapshot write fails, has no snapshot. Snapshots are an additional copy, alongside any other evidence this guide describes Above keeping, and are written only by Above's analysis runtime. In this release the portal does not display them; they are the stored record of the evidence behind incidents and Insights.

What is stored

- `query_anthropic_activity`: Compliance API account activity events, limited to the analyzed person's own activity events, one row per event the lookup returned.

Bounds

At most 8 KB per record (longer text fields are truncated and the row is marked truncated), at most 500 records per query, and at most 2,000 records per analysis run. Attachment and file bytes are never stored: only the fields the API returned as the record, and these queries never download file contents. A snapshot write that fails is logged, and the analysis continues without those rows.

Encryption

The record itself is stored in `payload_encrypted`, encrypted with your company's tenant encryption key, the same tenant encryption as Above's other encrypted evidence columns. When your company has configured a customer-managed key, that key applies (see Tenant Encryption (BYOK)). Stored in plaintext next to it: the query tool name, the provider's record identifier, or a SHA-256 digest of the record when the provider returns no id, the record's occurrence time, the identity ids of the person analyzed, the identity ids of the people in your company the record names as its author and its recipients (matched by email address when the record is stored; a Microsoft Entra object id, such as a Teams sender, is matched through the email addresses your synced Microsoft Entra directory holds for that user; a party that matches no one, or more than one person, is left unmatched), a request hash, the truncation flag, and the screening flag.

Retention

Kept for as long as the analysis record behind the incident or Insight it supports is kept; deleted with it, and deleted when your company's data is purged.

Tenant isolation

Every row carries your company id, and the table is protected by row-level security.

Prompt-injection screen

Each stored query result is screened for instruction-shaped text by Anthropic Claude Haiku 4.5 on Amazon Bedrock (in batches of about 24,000 characters, at most 60 screening calls per analysis run), and the yes/no result is stored on each snapshot row (`injection_suspected`); a row the screen did not reach, or whose screening call failed, keeps an empty value. The screen never blocks or changes what Above's analysis reads, and never deletes evidence. In the US deployment the screen uses Bedrock's global cross-Region inference profile, so requests may be processed in any supported commercial AWS Region (the same routing Above's analysis model already uses in the US); in the EU deployment it uses the EU cross-Region inference profile, so

processing stays in EU Regions. Amazon Bedrock does not share prompts and completions with model providers.

Permissions

No new permission, scope, or consent. Snapshots hold only data the Anthropic (Claude) Compliance integration's existing permissions already let Above read.

Required Scopes

The Compliance Access Key must carry both of these scopes, and Above uses no others:

- `read:compliance_user_data` : read chats, session transcripts, and file metadata.
- `read:compliance_activities` : read the account activity feed.

Above checks both against Anthropic while you are connecting. A key missing either one is **refused** and not stored, and the error names both required scopes. There is no partial mode: the integration either has both or is not connected. No other scope is required, and for least privilege the key should be minted with only these two.

No delete scope, ever

`delete:compliance_user_data` is deliberately never requested, and no Compliance API DELETE endpoint is ever called. Above's value here is read-only evidence. Holding a delete scope would make a compromise of Above capable of destroying your own compliance record, which is a far worse outcome than anything the scope would enable. If your key was minted with the delete scope attached, Above still never uses it, but minting a key without it is the stronger control.

Prerequisites

- An Anthropic organization on a plan that provides the Compliance API.
- The primary owner or an organization owner role in claude.ai, which is what can create a Compliance Access Key and choose its scopes. An organization admin in the Claude Console cannot: that role creates Admin API keys, which reach the activity feed only and are refused here.
- An Above account with the Admin or System Admin role. An IT-role account can see the card and its connection state, but every control is inert for it and the server refuses the change regardless.
- The **Shadow AI** module as well, if you want to read collected conversations in the portal. The Conversations tab requires Shadow AI plus the Claude activity module; collection itself does not. Shadow AI together with the Claude activity module also adds a Claude row with the Integration source badge to the Applications tab, counting the ingested conversations separately from any claude.ai browser usage the extension captured.
- The Claude activity module enabled for your Above tenant, **before** you connect. While it is off, the Claude Enterprise card is not shown at all if nothing has ever been connected, and a connection attempt is refused. Contact Above if the module is not enabled for you.

- Above identities whose email addresses match the Claude accounts you expect to see, so activity can be attributed. Matching is case-insensitive.
-

Connecting

1 Create a Compliance Access Key

In `claude.ai`, go to **Organization settings > API** and create a Compliance Access Key for your organization with the scopes `read:compliance_user_data` and `read:compliance_activities`. Do not add `delete:compliance_user_data`. Anthropic documents the steps under [Set up the Compliance API](#). Confirm the value you copy begins with `sk-ant-api01-`; a value beginning `sk-ant-admin01-` is an Admin API key and will be refused.

2 Connect it in Above

In Above, open **Integrations**, and under **AI Agents** find the **Claude Enterprise** card. Select **Connect**. Paste the key into the dialog and select **Connect** again. Above checks the key against Anthropic before storing it, and a key that fails that check is never stored.

3 Review what the card reports

Once connected, the card shows when it was connected and what it knows about two session families: **Claude Code & Cowork sessions**, run on a person's own machine, and **Cloud Cowork sessions**, hosted remotely. Each reads **not checked yet**, **available**, or **unavailable**. A family your organization does not expose is reported as unavailable rather than as an error, and an organization that later turns it on starts being collected without anyone reconnecting.

4 Run the first collection

The first collection reaches back 30 days for chats and seven days for sessions, so it takes longer than later ones. Select **Collect now** on the card to start it rather than waiting for Above to collect on its own. After that, Above collects new and updated Claude activity **hourly**. **Collect now** starts a run immediately instead of waiting for the next hourly tick; if a run is already queued, the card says so rather than starting a second. Each **Collect now** is recorded in your audit log as a **Claude Re-verify** entry.

5 Rotate or remove the key

To rotate a working key, select **Disconnect**, then **Connect** with the new value. If Anthropic stops accepting the stored key, the card says so and offers **Reconnect**, which opens the same dialog for the replacement key. Either way, storing a replacement key stops any collection still running under the previous key, in the same step that installs the new one, so activity from two Anthropic organizations is never blended. Connecting, reconnecting, and disconnecting are all recorded in the Above audit log.

Summarizing a Claude File

File contents are not collected, and file contents are not stored. When an analyst working an incident needs to know what was in a file, the incident evidence panel offers a **Summarize a Claude file** action that lists the Claude files belonging to **that incident's subject**, newest first. Choosing one asks Above to read that file from Anthropic **once**, describe its contents, and keep only the description. The file itself is never written to Above's storage.

A summary can be requested only when all of these hold:

- The incident belongs to your company and is visible to that analyst.
- The incident is neither hidden nor frozen.
- The file's parent conversation is a Claude conversation, is complete, and is still available.
- The file belongs to the incident's subject.
- The Claude module is enabled, the integration is active, and the key carries the user-data scope.

A frozen incident is refused explicitly: Above says the incident is frozen, because that is a state you can act on. Every other authorization failure is refused identically, without indicating which condition did not hold. That uniformity is intentional: a more specific answer would let anyone with an incident URL learn who in your organization uses Claude.

Whether a file *can* be summarized is decided from the metadata Above already holds, before anything is fetched from Anthropic. A file is refused up front, and no request is made to your Anthropic organization, when its type is one Above cannot read, when it is larger than **25 MiB**, or when Anthropic reported no size for it. After a file is read, its bytes still decide: a file whose contents do not match its declared type, or that yields no readable text, is recorded as unreadable.

The work runs in the background, so the panel reports where each file stands: **Summarizing**, **Summarized**, **Could not be read**, **Too large to summarize**, **Unsupported file type**, **No recorded size**, or **Summary failed**. Only the last of those can be asked for again: re-asking about a file that is too large, or whose bytes could not be read, would call Anthropic again to be told the same thing.

Requesting a summary writes **two** audit entries, both in the same transaction as the request itself so neither can record something that did not happen: a **Claude File Summary Requested** entry naming the file, and a **Claude File Summary Requested (Anthropic)** entry naming its kind and the product surface it came from, so you can filter your audit log for exactly the files read out of a provider. Neither entry contains the summary, and neither contains any of the file's contents.

The file is read once and never stored

Above holds the file's bytes only for as long as it takes to describe them. What is kept is the description, stored against the file's record and encrypted at rest, under your own tenant envelope key where Tenant Encryption is enabled for your workspace, and under Above's platform key otherwise. A summary is written by a model from the file's contents; treat it as a description, not as the file.

When Content Is Not Available

Content can stop being readable for several different reasons. Three of them are configuration:

- **HIPAA.** A HIPAA-configured organization does not have its local Claude Code sessions captured, so the session lists contain none. Above records empty coverage rather than an error.
- **Zero data retention.** A ZDR session is omitted from the lists and cannot be retrieved. Above marks that one session unavailable and carries on with the others. ZDR is a policy, not an erasure.
- **The session family being switched off.** Anthropic can report that Claude Code and Cowork sessions are not available to your organization at all. Above records that and keeps collecting chats and remote sessions.

Individual messages can also come back without content, because they were never captured, because they aged past your organization's retention period, because the message was too large, or because a customer-managed encryption key covering them has been revoked. Above records that the turn existed and why its content is absent.

Unavailable content is not evidence

Content Above cannot read is never reconstructed, inferred, or summarized. A conversation with no available text is recorded as having no content, is not sent to a model, and cannot be revealed to an analyst. This holds for content under a revoked customer-managed key specifically: the record that the conversation existed survives, and its content does not become evidence.

When Anthropic reports a chat as deleted, or a transcript as no longer available, Above records that fact and stops showing the text, in the portal and in investigations alike. An investigation that already cited a conversation stays coherent, because the finding and the timeline entry are Above's own records of what an analyst saw, not a live re-read of provider content.

Disconnecting is not deletion. Disconnecting removes the stored Compliance Access Key and stops further collection. It does not delete the Claude activity Above already collected. To remove that history, use Above's data-deletion process.

Validation Checklist

After connecting, confirm:

- The key was accepted, which means its prefix was correct and both scopes were confirmed against Anthropic.
- The card shows the integration as connected, with a connection date.
- An integration-connected entry is present in the Above audit log.
- After the first collection, open **Shadow AI > Conversations** and confirm Claude rows appear for people whose Claude email matches an Above identity. Conversations is the tab that carries provider-collected transcripts; the Prompts tab shows browser-observed activity only, so an empty Prompts tab is not a sign that collection failed.

- If you expected Claude Code sessions and see none, check these three causes: Bedrock, Vertex, or Console API key authentication; a HIPAA-configured organization; or the session family being unavailable to your organization, which the card reports directly.
- On an incident whose subject uses Claude, the evidence panel offers **Summarize a Claude file** and lists that subject's files.

After disconnecting, confirm:

- The card no longer shows the integration as connected.
- An integration-disconnected entry is present in the Above audit log.
- The key is deleted in claude.ai organization settings, if you are retiring it there as well.

Security and Least Privilege

Recommended controls, ahead of and after enablement:

- Mint the Compliance Access Key with only the two read scopes. Do not attach `delete:compliance_user_data`, even though Above would never use it.
- Keep the key out of shared password stores and chat. Above stores it encrypted at rest and never returns it to a browser. Once the connection succeeds, delete any local plaintext copy you made along the way. Above cannot see where else a pasted key may have landed, so treat clipboard history, password managers, backups and chat logs as part of that cleanup, and rotate the key if you are unsure.
- Rotate the key on your normal credential-rotation schedule: mint a new one, then use **Disconnect** followed by **Connect** in Above. Above replaces the stored value rather than accumulating keys.
- Restrict Above portal access to authorized administrators and investigators, and use the per-user identity visibility settings so each analyst sees only the people they should.
- Disconnect the integration in Above and delete the key in claude.ai organization settings when it is no longer needed.

Disconnecting

Open **Integrations**, find the **Claude Enterprise** card, select **Disconnect**, and confirm. **Disconnect** stays available to an administrator even if the Claude activity module is later switched off, so a stored key can always be removed.

What disconnecting does, and what it does not do

Disconnecting deletes the stored Compliance Access Key from Above and records an audit-log entry. Collection stops, the investigator activity lookup becomes unavailable, and no new file summary can be requested. A collection already under way is stopped in the same step, rather than allowed to finish. Claude activity already collected is **kept, not deleted**, so investigations that cited it stay complete. To remove that stored history, use Above's data-deletion process. To retire the credential on Anthropic's side as well, delete the key in claude.ai under Organization settings > API.

Out of Scope

So that the boundary is explicit rather than inferred, this integration does **not** include any of the following:

- **OpenTelemetry telemetry from Claude Code.** Above offers a separate Claude Code OpenTelemetry integration for that; this one reads the Compliance API and nothing else.
 - **Refusal or moderation classification.** Above does not label a prompt as refused, moderated, or policy-violating from this source.
 - **The Anthropic organization user directory.** Above does not read your Claude user roster; attribution is by email address against identities Above already holds.
 - **Any write or delete against Anthropic.** No Compliance API write or DELETE endpoint is ever called.
-

Official references

- Compliance API access and scopes: <https://platform.claude.com/docs/en/manage-claude/compliance-api-access>
 - Compliance content and data: <https://platform.claude.com/docs/en/manage-claude/compliance-content-data>
 - Compliance sessions (Claude Code and Cowork): <https://platform.claude.com/docs/en/manage-claude/compliance-sessions>
 - Compliance activity feed: <https://platform.claude.com/docs/en/manage-claude/compliance-activity-feed>
 - Compliance API errors: <https://platform.claude.com/docs/en/manage-claude/compliance-errors>
-

AI agent identities created from this source

A Claude Code or Claude Cowork session (`product_surface` `claude_code` or `cowork`) whose account is a person in your directory joins that person's agent identity for the product, the same identity their endpoint sessions land on. claude.ai chat conversations and transcripts owned by a Claude agent create none. See AI Agents as Identities.