Claude Desktop on third-party platforms

Security Overview

Document Information

Distribution: This document is provided under the terms of your existing NDA with Anthropic and is intended solely for your organization's internal security review. Please do not share or distribute externally.

Version: 3.0

Last Updated: June 2026

Changelog

- Version 3.0: Expanded the MDM key table (tab and permission controls, OTLP content capture, plugin provisioning, credential helper); documented the bootstrap server; corrected the OTLP key descriptions.
- Version 2.0: Added additional MDM keys; improved architecture diagram

This document reflects the architecture as of the date above. As the software continues to evolve, implementation details may change in future releases.

For questions about this document or to request additional technical details, please contact your Anthropic account representative.

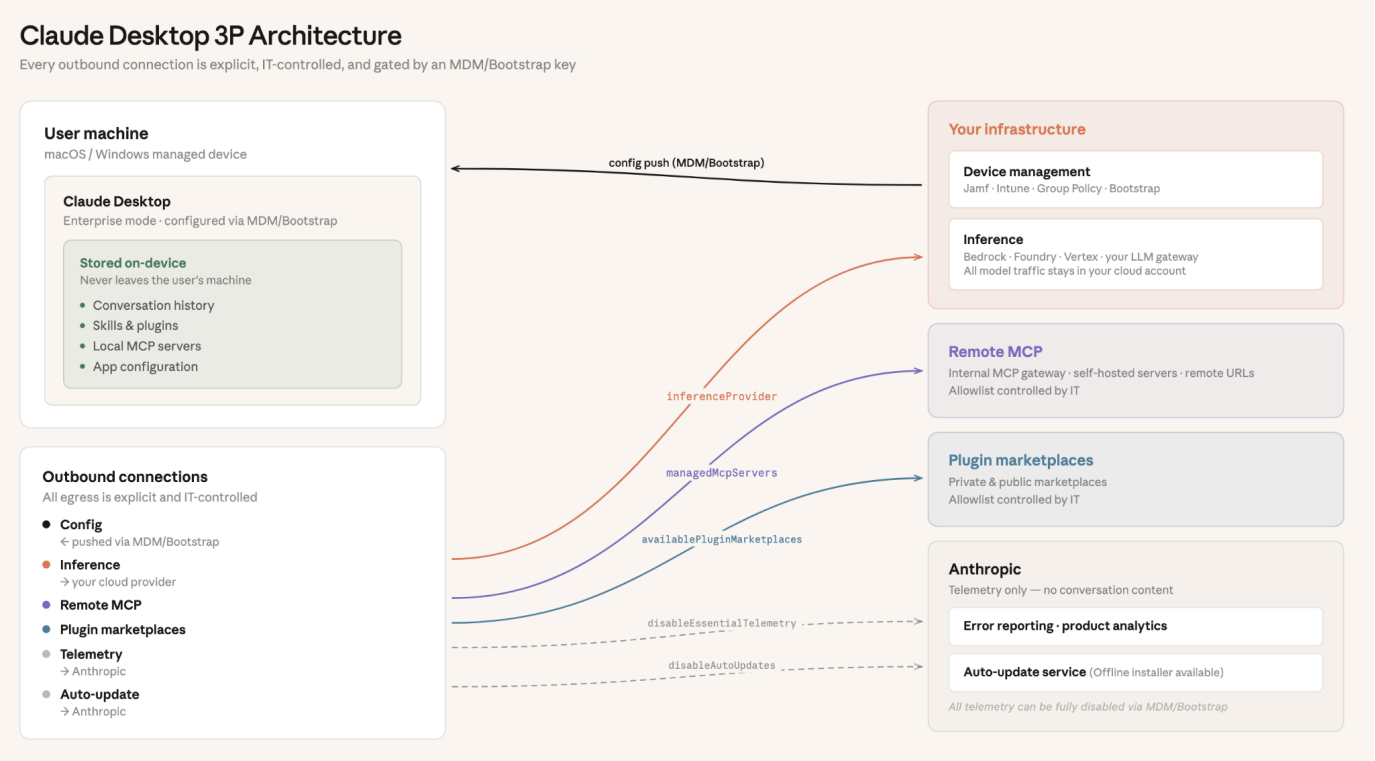
Overview

Claude Desktop on third-party platforms (3P) is the same commercial Claude Desktop build (with some features unavailable in this configuration), configured by your organization's device-management system to route model inference to a cloud provider (AWS Bedrock, Google Cloud Vertex AI, or Microsoft Azure AI Foundry) rather than to Anthropic's hosted API. The third-party inference behavior is activated through managed-preferences keys distributed by your existing MDM tooling.

In this configuration, conversation content never reaches Anthropic infrastructure. Prompts, attached files, tool inputs and outputs, and model responses travel only between the end-user's machine and your chosen cloud inference endpoint. Administrators narrow that allowlist further with three policy keys covering error reporting, product analytics, and non-essential supporting services, and may disable auto-update to operate fully air-gapped from Anthropic.

An architecture diagram is provided below. The relevant flags and egress paths are described in more detail throughout this document.

This document only covers 3P-specific architecture changes. For more information about the underlying desktop architecture of Cowork (which is shared between first-party and third-party modes), please reference “Claude Cowork Desktop Security Architecture Overview” in the Anthropic Trust Center.



1. Telemetry Controls

The application separates outbound diagnostic traffic into three categories, each controlled by a single MDM key. All three channels of telemetry are **on (data is sent) by default**, and become **off (data is not sent)** when the disable keys are set to `true`.

No telemetry channel — in any configuration — transmits prompt text, model responses, attached file contents, or inference API credentials. The categories below describe diagnostic and usage metadata only.

1.1 disableEssentialTelemetry — Error reporting and performance timing

Default	`false` (error reports are sent)
When `true`	Crash reports, error stack traces, and page-performance timing are not sent. The application does not initialize its error-reporting or real-user-monitoring clients, and the corresponding network destinations are removed from the CSP.

What is sent (error reports):

- Exception type, error-category code, and error message text with user home-directory paths redacted to <home>/…
- Stack trace with frame filenames normalized to application-relative paths (no on-disk install location)
- Breadcrumb trail of recent application lifecycle events (window opened, view navigated) and recent HTTP requests (method and URL only; query strings and fragments stripped)
- Application name and version; runtime versions (Electron, Chromium, Node)
- OS name and version; CPU model string; total memory; screen resolution; application memory working set
- Plugin, skill, extension, and MCP-server identifiers with user-chosen names replaced by <redacted> (Anthropic-published names preserved)
- A random per-install UUID generated on first launch (not derived from hardware, the OS user account, or any Anthropic account)
- Coarse geolocation (country and region) derived server-side from connection IP
- Event timestamp

What is sent (performance timing):

- Page-load and view-transition timing; resource-fetch latency; long-task duration
- Application version; OS and browser-engine version; device family; viewport dimensions
- An anonymous session UUID (no user identity is attached)
- Client IP address and IP-derived geolocation (country and region)

What is never sent:

- Conversation content — prompts, model responses, tool inputs/outputs, attached file contents
- Account email, display name, or any Anthropic account identifier
- Machine hostname or OS user-account name
- Hardware serial numbers, MAC addresses, or other hardware-derived identifiers
- Local-variable values from stack frames
- Cookies or HTTP request/response headers and bodies
- Session-replay recordings (the replay feature is disabled; sample rate is zero)

Scrubbing applied before transmission:

A scrubbing layer runs on every error event before transmission. It redacts user-directory paths from error messages (macOS and Windows home-directory patterns), strips query strings and fragments from all recorded URLs, and replaces user-chosen plugin, skill, and MCP server names with a `<redacted>` placeholder while preserving Anthropic-published ones. The reporting client is configured not to capture local-variable values from stack frames, not to send the machine hostname, and not to attach IP address, cookies, or request headers. The user identifier is a random UUID generated at install time; it is not derived from hardware, the OS user account, or any Anthropic account. Error and crash reports are retained for 90 days; performance-timing data for 30 days. Application-health and product-analytics events are retained per Anthropic's data-retention policy.

Performance timing data (page load, asset fetch latency, long-task duration) is collected with input masking enabled, user-interaction tracking disabled, and session replay disabled. No user identity is attached.

Why an organization would leave this enabled

Error reports are the primary mechanism by which Anthropic detects, reproduces, and fixes faults specific to enterprise deployments. Disabling this key has no effect on application functionality, but issues encountered on your fleet will be invisible to Anthropic unless reported manually.

1.2 disableNonessentialTelemetry — Product analytics

Default	`false` (analytics are sent)
When `true`	Product-usage analytics and application-health events are not sent. The analytics client is not initialized, the event-logging queue drains without transmitting, and the corresponding network destinations are removed from the CSP.

What is sent (application-health events):

- Event name (e.g. desktop_session_initialized, desktop_update_downloaded, vm_startup_completed, notification_shown)
- Event-specific counters and durations — e.g. step duration in milliseconds, exit code, count of installed extensions, character-length of an input. **Counts and durations only; never the content itself.**
- The random per-install UUID described above
- organization_id — the value of the deploymentOrganizationUuid MDM key if you set one, otherwise a fixed placeholder UUID
- deployment_mode — a fixed string indicating the application is running in enterprise configuration

- Application version and commit hash; OS platform, architecture, version, release, and build; CPU model string; total and free memory
- Event timestamp

What is sent (product analytics — renderer):

- Events that fire without an account carry only: an anonymous client-generated ID, the page path, locale, timezone, user-agent string, and event-specific properties (enums, booleans, counts, durations).

What is never sent:

- Conversation content — prompts, model responses, tool inputs/outputs, attached file contents
- Input text (only the integer length is recorded, never the text)
- Account email, display name, or any Anthropic account identifier (structurally absent in enterprise mode)
- OS user-account name
- IP address (redacted to a literal "REDACTED" string before write)

Scrubbing applied before transmission:

Because Claude Desktop 3P operates without an Anthropic-hosted account, no email address, account identifier, display name, or marketing-attribution cookie is ever populated. Events carry: a per-install random UUID, the `organization_id` supplied via the `deploymentOrganizationUuid` MDM key (or a fixed placeholder UUID if unset), application version and build hash, OS platform/version, CPU model string, total/free memory, and event-specific counters. Where an event references something user-named (a workspace path, a plugin identifier, a command's stderr tail), the same path-redaction and name-redaction rules described above are applied before queueing. Events record **counts and durations, not content**. The length of a quick-entry submission is logged as an integer, while the text itself is not.

Why an organization would leave this enabled

Aggregate usage data informs which enterprise features Anthropic invests in. Disabling this key has no functional effect on the application.

1.3 disableNonessentialServices — Supporting services

Default	`false` (services are reachable)
When `true`	The application does not fetch connector favicons from a third-party icon service, and does not load the sandboxed iframe used to render interactive Artifacts.

This key governs outbound requests that are not telemetry but are also not required for core operation:

Service	What it does	Effect when disabled
Connector favicons	Fetches a 16×16 icon for each configured MCP server from a public favicon service. **This request discloses the hostname of each configured MCP server to that third-party service.**	Connectors display a generic letter-tile icon instead.
Artifact sandbox	Loads an isolated-origin iframe used to render interactive HTML/React Artifacts produced by the model.	Artifacts containing executable web content do not render. Text, Markdown, and CSV artifacts are unaffected.

Why an organization would set this to true

Organizations that operate internal MCP servers on private hostnames and do not want those hostnames disclosed to a third-party favicon resolver should set this key. Organizations with no need for interactive Artifacts can also set it to minimize the egress surface.

2. Network Egress Allowlist

The table below lists every external destination the application may contact in Claude Desktop 3P, grouped by function and by the MDM key that removes it. Destinations are enforced by Content Security Policy; a destination removed by policy is rejected by the browser engine, not by application logic.

Function	Domain(s)	Controlled by
Model inference	Configured Bedrock / Vertex AI / Azure AI Foundry endpoint, or self-hosted gateway URL	`inferenceProvider` + provider-specific keys (always allowed)
Auto-update — version check	`api.anthropic.com` (path `/api/desktop/*/update` only)	`disableAutoUpdates`

Auto-update — binary download	`downloads.claude.ai` (path `/releases/*`)	`disableAutoUpdates`
Sandbox VM image download and Claude Code CLI download (for Code tab)	`downloads.claude.ai` (path `/vms/*`)	Always allowed (one-time, content-addressed by SHA)
Error reporting	`sentry.io`, `*.sentry.io`	`disableEssentialTelemetry`
Performance timing	`browser-intake-datadoghq.com`, `browser-intake-us3-datadoghq.com`, `browser-intake-us5-datadoghq.com`, `browser-intake-ap1-datadoghq.com`, `browser-intake-ap2-datadoghq.com`, `browser-intake-datadoghq.eu`, `browser-intake-ddog-gov.com`	`disableEssentialTelemetry`
Product analytics	`a-cdn.anthropic.com`, `a-api.anthropic.com`	`disableNonessentialTelemetry`
Application-health events	`claude.ai` (path `/api/event_logging/*` only)	`disableNonessentialTelemetry`
Connector favicons	`www.google.com` (path `/s2/favicons` only), `*.gstatic.com` (path `/faviconV2` only)	`disableNonessentialServices`
Artifact sandbox iframe	`www.claudeusercontent.com`	`disableNonessentialServices`

With `disableEssentialTelemetry`, `disableNonessentialTelemetry`, `disableNonessentialServices`, and `disableAutoUpdates` all set to `true`, the only remaining

egress is to your inference endpoint and the one-time VM image fetch from `downloads.claude.ai/vms/*`.

3. Administrator Controls (MDM)

All keys are delivered via standard managed-preferences channels: a Configuration Profile targeting domain `com.anthropic.claudefordesktop` on macOS, or the equivalent registry policy path on Windows. Boolean keys are absent-means-default; setting a key explicitly to its default value is equivalent to leaving it unset.

Key	Type	Default	Controls	Locked-down recommendation
`allowedWorkspaceFolders`	JSON array of strings	unset (unrestricted)	Absolute paths the user may attach as workspace folders for the agent to read and edit. A leading `~` expands to the per-user home directory.	Set to a fixed list (e.g. `["~/Projects"]`) to prevent users from mounting sensitive locations such as `~/Library`, network shares, or the filesystem root.
`coworkEgressAllowedHosts`	JSON array of strings	unset (only the inference endpoint is reachable)	Allowlist of domains that can be egressed to by the Cowork main process and VM.	Set to a fixed list (e.g. `["domain.com"]`) to prevent Cowork from egressing to non-allowed domains.
`disabledBuiltinTools`	JSON array of strings	unset (all tools available)	Removes named tools from the agent's tool list (e.g. `["WebSearch", "WebFetch"]`).	Set to disable any built-in tool whose network or filesystem reach exceeds your policy.
`builtinToolPolicy`	JSON object	unset (allow)	Per-tool approval policy. Setting a tool	Set tools whose actions you want

			to ask requires user approval before each call; allow is the default behavior. Use disabledBuiltinTools to remove a tool entirely.	human-gated to ask, without removing them from the tool list.
`autoModeEnabled`	boolean	false	Whether users may select Auto mode (Act without asking) in the Cowork and Code permission selectors. In Auto mode a safety classifier decides which actions need approval instead of the static per-tool policy.	Leave false where every consequential action must be approved by the user.
`isLocalDevMcpEnabled`	boolean	`true`	Whether end users may add their own stdio MCP servers in Settings. When `false`, only servers from `managedMcpServers` are available.	`false` for fleets where connector approval is centralized.
`isClaudeCodeForDesktopEnabled`	boolean	`true`	Whether the Code tab is exposed in the Claude Desktop 3P application.	`false` if you do not want to provide access to the Code tab, which runs the Claude Code CLI.
`coworkTabEnabled`	boolean	`true`	Whether the Cowork tab is exposed. At least one of the Cowork, Code, or Chat tabs must remain enabled; if all are disabled the application re-enables	false if your deployment is Code-only or Chat-only.

			Cowork with a warning.	
`chatTabEnabled`	boolean	`false`	Whether the Chat tab is exposed. Opt-in; the Chat tab is a beta feature and is also hidden when beta features are disabled.	Leave false until your organization has reviewed the Chat surface for its policies.
`chatAdvancedFileAnalysisEnabled`	boolean	`false`	Allows Claude to run code in an isolated sandbox during Chat conversations to analyze attached files it cannot read natively. The sandbox can read only the files attached to the conversation. Has no effect unless chatTabEnabled is true. Beta feature.	Leave false unless Chat users need spreadsheet and document analysis; review sandbox behavior first.
`disableBundledSkills`	boolean	`false`	Disables the bundled skills and workflows that ship with the Code tab engine (deep research and similar).	true where bundled skills cannot function, for example when web access is egress-blocked.
`isDesktopExtensionEnabled`	boolean	`true`	Whether end users may install local desktop extensions (`.mcpb` packages).	`false` if extensions are not part of your deployment, or pair with the signature-required key below.
`isDesktopExtensionSignatureRequired`	boolean	`false`	Rejects any desktop extension whose package signature does not chain to a	`true`. This is the primary control preventing unsigned or tampered

Bootstrap server (per-user configuration). In addition to fixed MDM values, configuration can be delivered per user by a bootstrap server: bootstrapUrl (MDM only) points the application at an HTTPS endpoint your organization operates that returns each user's configuration at sign-in, bootstrapOidc has the user authenticate to that endpoint through your identity provider first, and bootstrapEnabled opts a device out. Values returned by the bootstrap server override local settings, become read-only on the device, and may include per-user inference credentials. The device contacts your endpoint directly; Anthropic infrastructure is not involved. Treat the bootstrap server as a trusted configuration source with the same change controls as your MDM, since it can set most of the keys in the table above.

4. End-User Controls

Within the bounds set by the administrator keys above, individual users can configure the following in **Settings**:

Surface	What the user can do	Administrator override
MCP servers	Add, edit, and remove personal MCP server connections (URL, transport, headers, OAuth).	Hidden entirely when `isLocalDevMcpEnabled` is `false`; user sees only administrator-provisioned servers.
Desktop extensions	Install `.mcpb` extension packages from disk.	Install disabled when `isDesktopExtensionEnabled` is `false`; unsigned packages rejected when `isDesktopExtensionSignatureRequired` is `true`.
Skills and plugins	Install plugins (bundles of skills, agents, hooks, and MCP servers) from a marketplace or local path.	Administrator-provisioned plugins are pushed to a root-owned system directory via MDM and load alongside any user-installed plugins.
Workspace folders	Attach local folders for the agent to read and edit.	Restricted to paths under `allowedWorkspaceFolders` when that key is set.
Built-in tools	Toggle individual built-in tools (web search, file operations, etc.) per conversation.	Tools listed in `disabledBuiltinTools` are removed from the tool list and cannot be re-enabled by the user.

5. Data Residency

Model inference traffic is sent only to the cloud inference endpoint and region your organization configures. For Amazon Bedrock and Google Cloud's Vertex AI, that endpoint resides in the region you select when provisioning the resource; Anthropic has no visibility into, and receives no copy of, that traffic.

Microsoft Foundry is currently in preview. In this preview platform integration, Claude models run on Anthropic's infrastructure. This is a commercial integration for billing and access through Azure. As an independent processor for Microsoft, customers using Claude through Microsoft Foundry are subject to Anthropic's data use terms. Anthropic continues to provide its industry-leading safety and data commitments, including zero data retention availability.

When using Bedrock and Vertex inference, the only data that can leave the device for Anthropic or its contracted third-party telemetry processors is the diagnostic and usage metadata described in **Section 1, Telemetry Controls**, and every such destination can be disabled by MDM policy. With `disableEssentialTelemetry`, `disableNonessentialTelemetry`, `disableNonessentialServices`, and `disableAutoUpdates` all set to `true`, the application makes no outbound connections to Anthropic or its telemetry processors during normal operation, satisfying strict data-residency requirements without network-layer controls.