coThink

TRUST CENTER / TECHNICAL PAPER

Security Whitepaper

How coThink protects collaboration data across identity, authorization, encryption, AI-provider boundaries, software delivery, monitoring, and recovery.

A public technical overview of identity, authorization, encryption, AI-provider boundaries, software delivery, monitoring, and recovery in the hosted coThink production service.

VERSION 1.0

PUBLISHED August 2026

CLASSIFICATION Public

OWNER coThink Security and Engineering

Contents

This paper is organized for vendor review: scope first, control design second, and current limitations before customer hardening guidance.

Executive summary

coThink is a multi-user AI collaboration workspace. It keeps conversations, files, decisions, tasks, guided sessions, and AI-assisted work inside persistent organizational workspaces. That makes security more than a login screen: the platform must protect tenant boundaries, apply permissions consistently, secure customer-supplied credentials, and make the path from a user request to an AI provider understandable.

coThink's security model is built around five principles:

1. **Verify the user.** coThink supports passkeys, multi-factor authentication, federated sign-in, and enterprise identity controls.
2. **Authorize every action.** Organization, workspace, room, feature, worker, realtime, and search paths are subject to server-side policy enforcement.
3. **Encrypt deliberately.** Traffic is encrypted in transit. Designated secrets and encrypted content receive application-layer encryption. Optional encrypted-room modes let customers choose between server-processable organization-managed encryption and client-held end-to-end encryption.
4. **Keep provider boundaries visible.** Customers choose which AI providers and integrations to use. Data sent to a connected provider is governed by that provider's terms and controls.
5. **Describe maturity honestly.** Implemented, partial, and planned controls are kept separate. coThink has not represented itself as SOC 2 or ISO 27 001 certified, and this whitepaper is not an audit report or certification.

This document describes the hosted coThink production service as of August 2026. Enterprise on-premises or specially contracted deployments may use different infrastructure and control allocations.

1. Scope and security model

The security boundary covered here includes the coThink web application, API, realtime service, background workers, relational data store, queue and cache, object storage, deployment system, monitoring, and the Cloudflare edge path used by the public service.

The platform is designed for organizational separation. Users belong to organizations and workspaces; rooms add a narrower membership and content boundary. Authentication establishes identity, while authorization determines whether that identity may perform a specific action on a specific resource.

coThink uses a shared-responsibility model:

Party	Primary responsibilities
coThink	Operate and patch the service; enforce tenant and role boundaries; protect platform secrets; provide security controls; monitor service health; maintain backup and incident procedures.
Customer organization	Manage users, roles, invitations, MFA and SSO settings; classify data; choose encryption modes and connected providers; safeguard customer-held keys; review exports, retention, and offboarding.
Connected provider	Protect data it receives under its own contract, privacy policy, retention model, and security program. This includes customer-selected AI, identity, calendar, mail, storage, and other integration providers.

No security control removes the need for appropriate customer configuration. A workspace administrator can weaken practical security by over-assigning roles, inviting the wrong person, selecting an unsuitable AI provider, or losing a customer-controlled encryption key.

2. Production architecture

The production service separates public ingress from application and data services.

- **Edge and ingress.** Cloudflare provides the public edge path, TLS termination, CDN functions, and edge security. Production application endpoints enforce HTTPS and HTTP Strict Transport Security (HSTS). Browser and API traffic requires TLS 1.2 or later, with TLS 1.3 preferred where supported.
- **Application services.** Web, API, realtime, and background-worker functions run as separate containerized services. This limits direct exposure and lets operational health be checked independently.
- **Data services.** PostgreSQL is the system of record for structured application data. Redis supports queues, coordination, and selected cache functions. S3-compatible object storage holds customer files and other object payloads.
- **Private service path.** Internal services communicate on private service networks. Application services are reached through the ingress tier rather than being exposed directly to the public internet.
- **Environment separation.** Production, staging, and development use separate configuration and credentials. Development defaults are not accepted as production secrets.
- **Observability.** Service health checks, structured logs, error monitoring, queue and worker health, and public status probes support detection and recovery.

The public status service reports the health of the web, API, database, Redis queue, realtime, and worker components. A green status result is an operational signal, not proof that every security control has been tested at that moment.

3. Request and data flow

A typical authenticated AI-assisted request follows this sequence:

1. The user authenticates through a supported local, passkey, OAuth, or enterprise identity path.
2. coThink establishes a secure session and evaluates the user's organization, workspace, room, role, feature entitlement, and requested action.
3. The API reads only the data required for the authorized operation. Encrypted content is decrypted only when the selected encryption mode permits server-side processing.
4. If the user invokes an AI function, coThink sends the necessary prompt and permitted context to the selected provider. Customer-supplied provider credentials are decrypted server-side only for the authorized request.
5. The provider returns an output. coThink may store that output, usage metadata, task state, and audit-oriented records according to the workspace configuration and applicable product plan.
6. Realtime updates and background work are re-checked against their own authorization and provenance controls; an earlier browser decision is not treated as blanket authority for every later process.

Operational metadata must remain processable for the service to function. Depending on the feature, this can include tenant and room identifiers, membership, timestamps, routing data, key version identifiers, usage

metrics, and security events. Encrypted-room modes protect message content as documented below; they should not be interpreted as encrypting every item of workspace metadata or every file.

4. Identity and authentication

coThink supports several authentication methods so organizations can select controls appropriate to their risk and plan:

- passkeys using WebAuthn and FIDO2;
- multi-factor authentication using TOTP, email one-time codes, WebAuthn as a second factor, and recovery codes;
- federated sign-in through supported OAuth identity providers;
- SAML single sign-on for entitled enterprise organizations;
- SCIM 2.0 provisioning for entitled enterprise organizations; and
- local email and password authentication.

Local password protection

Local passwords follow coThink's implementation of NIST SP 800-63B memorized-secret guidance:

- passwords may be 8 to 256 characters;
- composition rules do not require arbitrary mixtures of uppercase letters, numbers, or symbols;
- Unicode input is normalized using NFC;
- proposed passwords are checked against a local common-password blocklist and the Have I Been Pwned k-anonymity service;
- password changes fail closed if breach screening is unavailable; and
- passwords are not stored in plaintext.

For storage, coThink hashes a SHA-256 digest of the normalized password with bcrypt at cost factor 10. The pre-hash prevents bcrypt's 72-byte input limit from silently truncating long passphrases. Bcrypt supplies a unique salt for each stored hash. Existing local accounts are upgraded to the digest form on a subsequent successful login when needed.

Sessions and recovery

Production sessions use cookies marked `HttpOnly` and `Secure`, with `SameSite=Lax`. Sessions are invalidated on sign-out and password change. Recovery and verification endpoints are rate-limited. Authentication factors, recovery codes, passkeys, and recovery access remain sensitive credentials; customers should protect them as carefully as passwords.

5. Authorization and privileged access

coThink applies role-based access control at the organization, workspace, and room levels. Owner, Administrator, and Member roles provide the basic organizational model, while resource membership and feature entitlements further narrow access.

Production authorization controls run in enforcement mode across the API, background jobs, realtime operations, and search filtering. Policy and feature-matrix drift checks are included in continuous integration

and production assessment probes. Denied decisions block the operation rather than merely generating a warning.

Privileged platform administration is separate from ordinary workspace administration. Elevated platform operations require passkey step-up authentication. Operator access follows least privilege, uses restricted administrative paths, and is recorded for review.

Security-relevant operations include authentication events, role and membership changes, administrative changes, encryption-policy updates, and key-management actions. Availability of customer-facing audit views varies by plan and feature.

6. Encryption and key management

Data in transit

Browser, API, realtime, and integration traffic uses TLS. Production endpoints redirect or reject insecure access and publish HSTS. Outbound calls to AI and integration providers are made over TLS; customers remain responsible for evaluating each provider endpoint and contract.

Application-layer encryption

Designated secrets and encrypted content use AES-256-GCM before persistence. AES-GCM provides confidentiality and an authentication tag that detects ciphertext modification. Each encryption operation uses a unique nonce or initialization vector appropriate to the scheme.

Application-layer encryption covers, among other designated fields:

- customer-supplied AI provider API keys;
- OAuth and integration tokens handled by encrypted credential stores;
- organization chat-key material;
- encrypted-room message ciphertext and key envelopes; and
- designated integration cache payloads, including advisor calendar-event cache data.

Platform encryption keys are supplied through protected environment configuration and kept separate from stored ciphertext. Production secrets and private keys are not committed to source control. Key versions and fingerprints support identification and controlled rotation without returning secret values through normal API responses.

Infrastructure-layer encryption

Persistent storage also relies on infrastructure controls, but coThink currently classifies infrastructure-layer encryption across all persistent storage as **partial**. The stronger public claim is therefore limited to application-layer encryption for designated secrets and encrypted content. Customers should not assume that every database field, log entry, object, backup, or item of metadata receives the same application-layer encryption treatment.

Organization-managed encrypted rooms (ct-org-v1)

Organization-managed rooms encrypt message content with AES-256-GCM using a 256-bit organization chat key. Per-message nonces and authenticated associated data bind ciphertext to the organization, room,

message, and key version. Organization key material is itself encrypted at rest and can be generated, imported, and rotated through authorized administrative flows.

Because the server can access the active organization key for an authorized request, this mode supports server-side functions such as AI inference, search, and export on decrypted content. It protects stored message content but is not end-to-end encryption from the service operator.

End-to-end encrypted rooms (ct-e2ee-v1)

End-to-end encrypted rooms generate a unique AES-256 content key for each message. That content key is wrapped separately for each recipient using RSA-OAEP-256 with 4096-bit user keys. The server stores ciphertext and recipient envelopes; recipient private keys remain client-side and are held in session memory during use.

coThink cannot decrypt E2EE message content without a recipient private key. This also means server-side features that require plaintext - including many AI, search, moderation, and export functions - may be unavailable or reduced in an E2EE room. If customer-held private keys are lost, coThink cannot recover the encrypted messages.

Encrypted-room controls currently describe message content. Unless another feature is explicitly documented as encrypted, customers should not assume the same E2EE protection applies to room names, membership, timestamps, audit events, files, whiteboards, tasks, or other workspace artifacts.

7. AI governance and third-party providers

coThink supports customer choice among AI providers and models. An organization may connect its own provider credentials or use an included AI option where available. Provider and model availability depends on configuration, plan, and product status.

The AI boundary works as follows:

- coThink sends prompts, permitted context, and necessary metadata only when a user or enabled workflow invokes an AI-assisted feature;
- customer-supplied provider keys are encrypted at rest and decrypted only server-side for authorized requests;
- coThink does not mark up customer-supplied provider usage, and the customer maintains the commercial relationship with that provider;
- prompts and outputs sent to a provider are subject to that provider's retention, training, privacy, security, and regional-processing terms; and
- customers are responsible for selecting a provider appropriate for the sensitivity and regulatory status of their data.

coThink does not use Customer Content through its platform-improvement processes to train third-party foundation models. coThink may process deidentified or pseudonymized workspace signals to improve its own routing, facilitation, memory, recommendation, workflow, and collaboration systems as described in the Terms of Service and Privacy Policy. That distinction matters: it is not a promise that no product-improvement processing occurs, and it does not override a connected provider's own policies.

Current subprocessors and customer-optional providers are listed in the Trust Center. Material categories include edge infrastructure, billing, transactional email, observability, AI providers, and customer-enabled productivity integrations.

8. Application and software-delivery security

Application defenses include:

- server-side input validation and structured API contracts;
- rate limiting on authentication, recovery, and other abuse-sensitive routes;
- CSRF origin validation and tokens on state-changing browser requests;
- a Content Security Policy and related security headers;
- safe rendering defaults in the React application;
- server-side authorization rather than reliance on hidden interface elements;
- structured logging with sensitive-field redaction; and
- separation of credentials from source code and build artifacts.

Changes are reviewed in version control. Pull-request and release workflows include linting, type checks, policy-drift checks, dependency audits, and other automated tests appropriate to the affected repository. Container image and dependency scanning are integrated into release workflows, with additional software-bill-of-materials, scanning, signing, and provenance gates available as release policy controls.

Production releases identify component source revisions and images, classify database migrations, and run service-specific smoke checks. Database-impacting changes require backup evidence according to migration risk. Application rollback uses a previously known-good release manifest; database restoration is treated as a separate, explicitly approved operation.

A formal recurring third-party penetration-testing program and publication of penetration-test summaries are planned, not represented as completed. Security researchers may report vulnerabilities through coThink's responsible-disclosure process.

9. Logging, monitoring, and privacy controls

coThink records security and operational telemetry needed to run the service, investigate failures, and review important changes. Logging pipelines redact common patterns for passwords, authorization headers, API keys, tokens, encryption keys, and other sensitive fields before emission. Redaction reduces risk but is not a substitute for avoiding unnecessary sensitive data in logs.

Sentry is used for error and performance monitoring. Client-side session replay, when sampled, is configured to mask text and input values and to block media. Diagnostic events may still contain technical metadata such as URLs, browser details, timing, stack traces, and error context. Sentry is listed as a subprocessor for this reason.

Monitoring covers application health, failed authentication patterns, background workers, queues, realtime functions, integration health, and service degradation. Public status reporting is designed to remain available independently of the primary application path when practical.

10. Incident response and vulnerability handling

Security reports may be sent to security@wecothink.com or submitted through the Trust Center. coThink triages reports, investigates scope, preserves relevant evidence, and coordinates remediation according to severity.

For customer-impacting incidents, operators use an acknowledge-update-resolve communication pattern. Legal or contractual notification duties are handled according to the applicable agreement and law. Unless a separate written agreement says otherwise, the public service does not promise a specific security-response time, uptime percentage, recovery objective, or service credit.

11. Availability, backup, and recovery

Production services expose health checks and are monitored independently. Database, queue, realtime, worker, API, and web health contribute to operational status.

The current database-protection baseline uses scheduled PostgreSQL logical backups, with a default 14-day retention period and documented restore procedures. Restore readiness is verified through archive checks and non-production or approved break-glass drills. Recovery from these backups is a manual operation.

This baseline should not be mistaken for continuous point-in-time recovery. Additional off-host backup resilience and write-ahead-log archiving are roadmap improvements. Backup statements in this section refer to the relational database; customers evaluating file or object recovery should request the current data-class-specific recovery scope.

Recovery-point and recovery-time values are operational targets, not public contractual commitments. Customers requiring committed RPO, RTO, availability, support response, or data-residency terms should address them in an enterprise agreement.

12. Data ownership, retention, export, and deletion

Customers retain ownership of their prompts, messages, files, notes, decisions, workspace content, and session artifacts. coThink does not sell Customer Content.

Export capabilities are available according to plan and configuration. Account deletion is processed as a reviewed request rather than immediate erasure. Deleted information may remain in backups for a limited period, and production retention-purge automation is currently classified as partial while purge jobs remain in dry-run. Customers with fixed retention or deletion obligations should confirm the applicable workflow before placing regulated data in the service.

The Privacy Policy and Data Processing Addendum describe processing roles, data-subject support, subprocessors, international transfers, and deletion or return obligations. Those documents control where their terms are more specific than this technical overview.

13. Compliance and assurance status

coThink labels framework status to avoid implying an independent assurance result that does not exist.

Area	Status as of August 2026	Meaning
GDPR	Policy aligned	Commitments are documented in the Privacy Policy and DPA; this is not a certification.
CCPA / CPRA	Policy aligned	California disclosures and rights are documented in the Privacy Policy.

Area	Status as of August 2026	Meaning
SOC 2 Type I	Roadmap	No completed SOC 2 Type I report is represented.
SOC 2 Type II	Roadmap	No completed SOC 2 Type II report is represented.
ISO 27 001	Roadmap	No ISO 27 001 certification is represented.
Formal penetration-testing program	Planned	Third-party assessment summaries will be published or shared when available.
Infrastructure encryption at rest	Partial	Application-layer encryption protects designated secrets and encrypted content; coverage is not universal.
Retention and deletion automation	Partial	Policies and reviewed requests exist; production purge automation remains in dry-run.
Business continuity documentation	Planned	Backup, restore, rollback, and incident procedures exist; a complete published continuity program remains on the roadmap.

This whitepaper is self-described product documentation. It is not an independent audit, legal opinion, penetration-test result, guarantee of security, or replacement for a customer's own risk assessment.

14. Customer hardening checklist

Organizations can materially improve their coThink security posture by taking the following steps:

1. Require passkeys or MFA for every user, and use SAML SSO and SCIM where plan and identity architecture support them.
2. Assign the fewest administrative roles necessary. Review owners, administrators, room memberships, and pending invitations regularly.
3. Choose room encryption based on the actual workflow. Use organization-managed encryption when authorized server-side AI and search are required; use E2EE only when client-held-key confidentiality outweighs those functions.
4. Establish a recovery process for passkeys, MFA, and customer-held encryption keys before relying on them for critical work.
5. Evaluate each AI and integration provider's retention, training, regional-processing, and security terms. Do not assume coThink's controls replace the provider's.
6. Classify regulated or highly sensitive data before upload. Confirm contractual requirements, retention, export, recovery, and deletion behavior for that data class.
7. Export critical records according to the organization's continuity and records-management policy.
8. Remove departing users promptly and rotate shared or potentially exposed credentials.
9. Report suspected compromise quickly to security@wecothink.com.

Related documentation

- Trust Center
- Security Overview
- Encryption and Key Management
- Identity and Access Management

- Application Security
- Availability and Resilience
- Controls inventory
- Subprocessors
- Privacy Policy
- Data Processing Addendum
- Responsible Disclosure
- Public Status

Document control

Version: 1.0 **Published:** August 2026 **Classification:** Public **Owner:** coThink Security and Engineering **Contact:** security@wecothink.com

This document will be revised when material architecture, control, subprocessor, or assurance status changes. The Trust Center is the current public source for updates between whitepaper revisions.