OpenAI

Private Safety Processing Overview

Technical White Paper

Published September 22, 2026

Table of Contents

Table of Contents

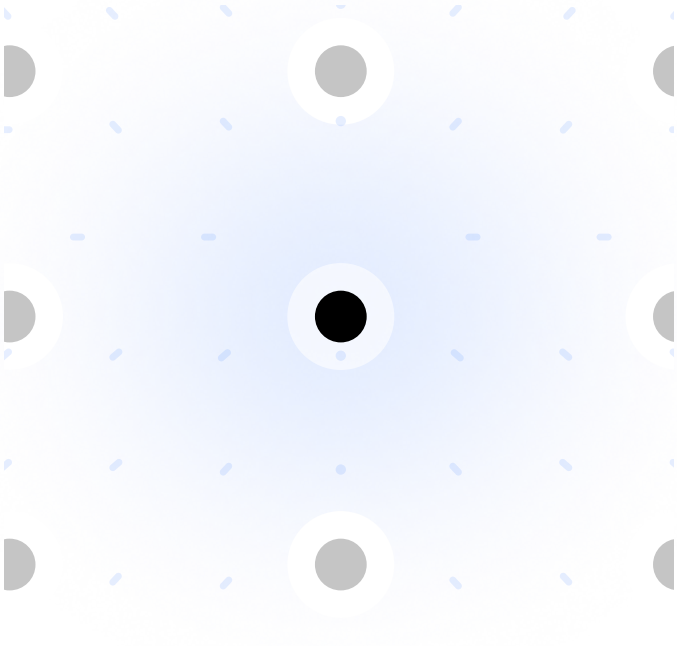
Introduction

OpenAI's mission is to ensure that artificial general intelligence (AGI) benefits all of humanity. Supporting that adoption requires OpenAI and enterprises bringing AI to the world to work together on safety while protecting the privacy of customer data.

Customers need safeguards against harmful activity such as agents that depart from their intended tasks. They also need their work, including cybersecurity and biological research, to proceed without unnecessary disruption.

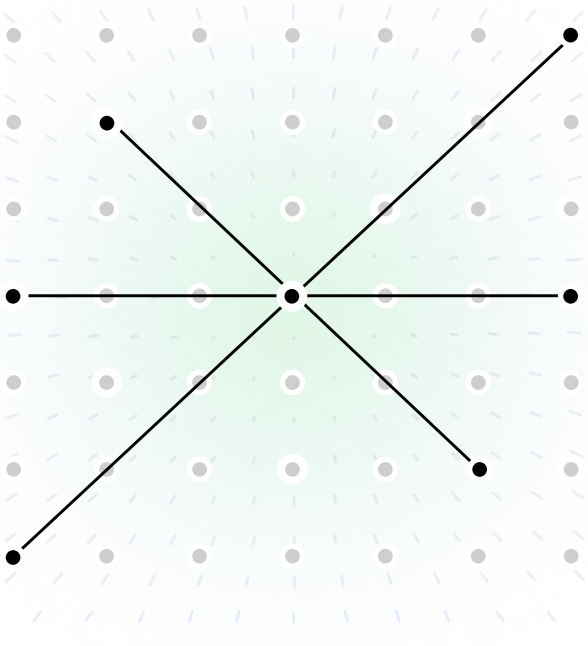
Reviewing related interactions gives safety systems more information to make accurate decisions. A request that appears concerning in isolation may be part of an authorized workflow. A sequence of otherwise ordinary requests may reveal a harmful pattern.

Private Safety Processing (PSP) is designed to retain and review content through automated safety systems while respecting user privacy. It is an early step towards a world where OpenAI and its customers collaborate to identify and prevent misuse. PSP was shaped by customers across industries, regions, and sizes, and their feedback continues to guide this work.



ISOLATED INTERACTION

An isolated observation does not establish a trend



MANY INTERACTIONS

Related observations reveal intent and patterns

Section 1

Zero Data Retention with Private Safety Processing

For many customers, Zero Data Retention (ZDR) is a critical privacy protection. Improvements to safety posture should not compromise customer privacy. Zero Data Retention with Private Safety Processing (ZDR with PSP) supports offline, automated safety review without OpenAI retaining customer prompts or responses.[1]

Under ZDR with PSP, referred prompts and responses are encrypted and retained in customer-controlled storage. OpenAI retains operational metadata and storage references, not plaintext or encrypted copies of customer content. With PSP, the retained content is decrypted only for approved automated safety review, and the decryption keys are inaccessible to OpenAI employees.

This paper focuses on ZDR with PSP. Other configurations of PSP are covered in Appendix A.

[1]Like other frontier model providers, OpenAI is required by law to report apparent child sexual abuse material (CSAM). Images flagged for potential CSAM will continue to be retained for manual review and reporting purposes, even in Zero Data Retention deployments, as they are today.

Section 1

Our Principles

ZDR with PSP is built with three principles:

01 Customers control their content

Customer content is stored in customer-controlled storage. Customers control the permissions and customer-managed Enterprise Key Management (EKM) authorization required to retrieve and decrypt protected safety records.

---

02 No human review

Safety review must not create a new way for OpenAI personnel to read protected customer content. Encrypted customer content is decrypted in an approved, hardware-attested safety runtime that disables human access. Only bounded safety signals and operational metadata leave the PSP protected review in plaintext.

---

03 Content retention for safety only

Content stored in customer-controlled storage serves only approved safety purposes. Customer content cannot be used to train models or be made available to other groups within OpenAI or its partners.

Architecture

Overview

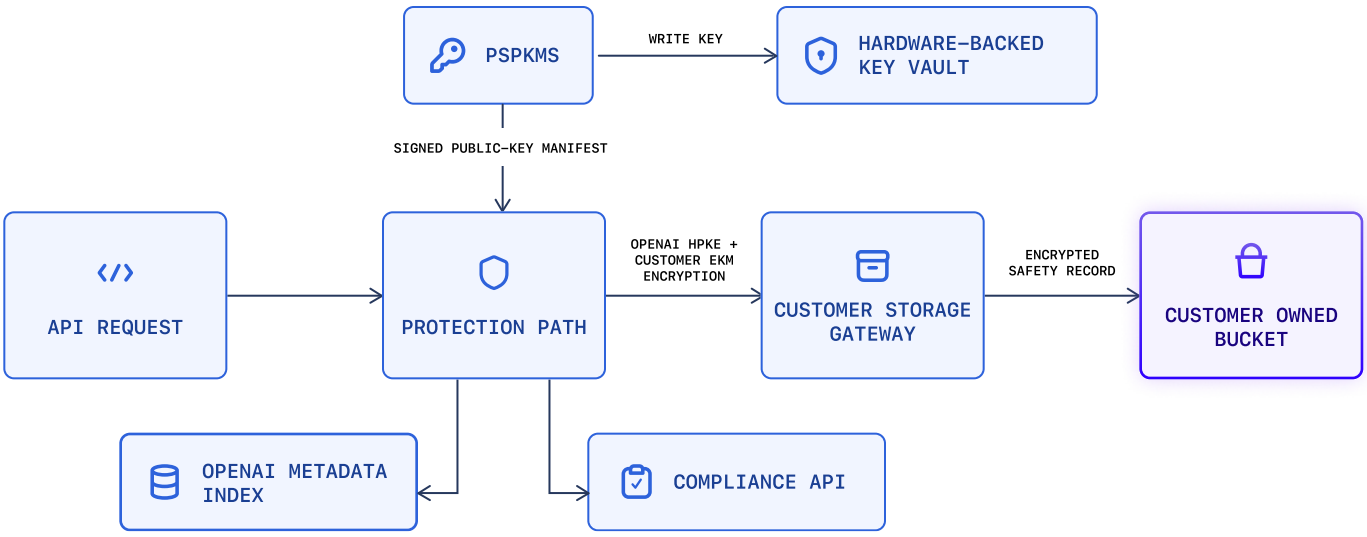
The architecture consists of two flows:

1. The API Request and Retention flow protects and retains eligible API content in a customer-controlled storage container.
2. The Asynchronous Safety Pipeline retrieves records only for approved automated safety review and releases only bounded safety decisions.

API Request and Retention

Interactions (prompts to, and responses from, the LLM), referred to in this paper as "customer content," are referred for review when they trigger a relevant safety-classifier signal or are selected under an approved sampling policy. Reviewing both classifier-triggered and sampled interactions helps identify emerging misuse and legitimate activity incorrectly classified by safety systems.

API Request and Retention



After an interaction is referred, OpenAI applies two encryption layers. The first layer uses OpenAI-managed encryption keys. The second layer is customer-managed via Enterprise Key Management (EKM). The OpenAI Customer Storage Gateway (CSG) then writes the doubly encrypted record to the customer's regional storage container.

The inner decryption keys are available only to approved confidential workloads that disable human access. OpenAI also creates a corresponding index entry containing only operational metadata and a storage reference, which is deleted after 30 days.

If encryption, storage authorization, or regional checks fail, the write is blocked. There is no fallback mechanism to store customer content within OpenAI in case of a failure. This PSP storage flow runs as an ephemeral, asynchronous task that does not block the inference path.

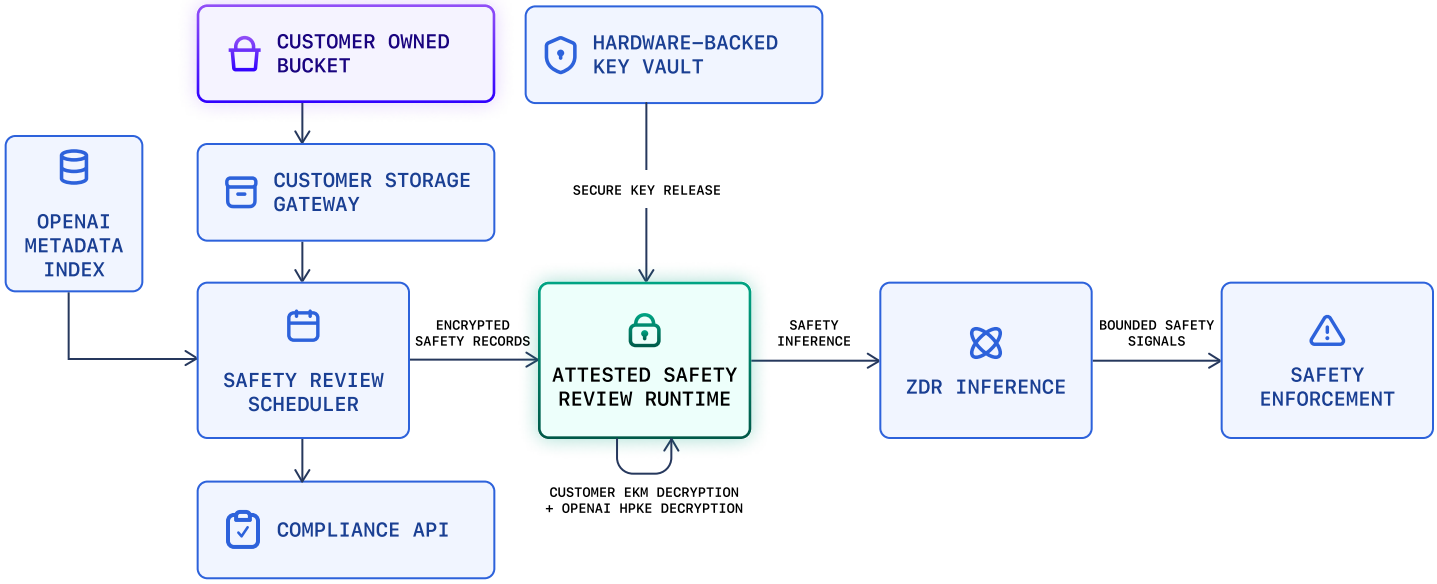
Section 2

Asynchronous Safety Pipeline

Automated safety review occurs asynchronously after records are retained. The Safety Review Scheduler (SRS) identifies the records using OpenAI's operational index, and the CSG retrieves them from customer-controlled storage. Review proceeds only after the records' identity, integrity, expiration, region, and customer-managed EKM authorization are verified.

The Safety Review Runtime (SRR) requests the OpenAI-managed HPKE decryption key through Secure Key Release (SKR). The key is released only after the runtime's identity, hardware attestation, software measurements, and key-release policy are verified. Neither SRS nor CSG receives the decryption key or customer plaintext, and no human access group can release the key. SRR and PSPKMS measurements are recorded in a tamper-evident transparency log.

Asynchronous Safety Pipeline



SRR performs automated review using an approved reviewer prompt and an output schema that satisfies SRR's fixed rules for plaintext output. Changes to reviewer prompts undergo two-party approval, and approved versions are recorded in a tamper-evident transparency log. SRR separately verifies that the supplied prompt appears in the log.

Detailed reviewer results are encrypted before leaving SRR and stored with the original safety record in customer-controlled storage. Only approved, bounded safety decisions and operational identifiers are returned in plaintext. Free-form explanations and reviewer notes remain encrypted for future investigations.

Section 2

Boundaries

The standard Private Safety Processing runtime is protected from human access, but its storage-disabled inference path does not provide an inference-wide Zero Operator Access (ZOA) guarantee. Private Inference is an optional protection that extends operator access protections and attestation to eligible safety-inference workloads. It does not automatically cover the initial API request, which keeps its original ZDR protections, or every OpenAI system.

This white paper covers first-party OpenAI traffic only. Amazon Bedrock and other third-party distribution arrangements are outside its scope.

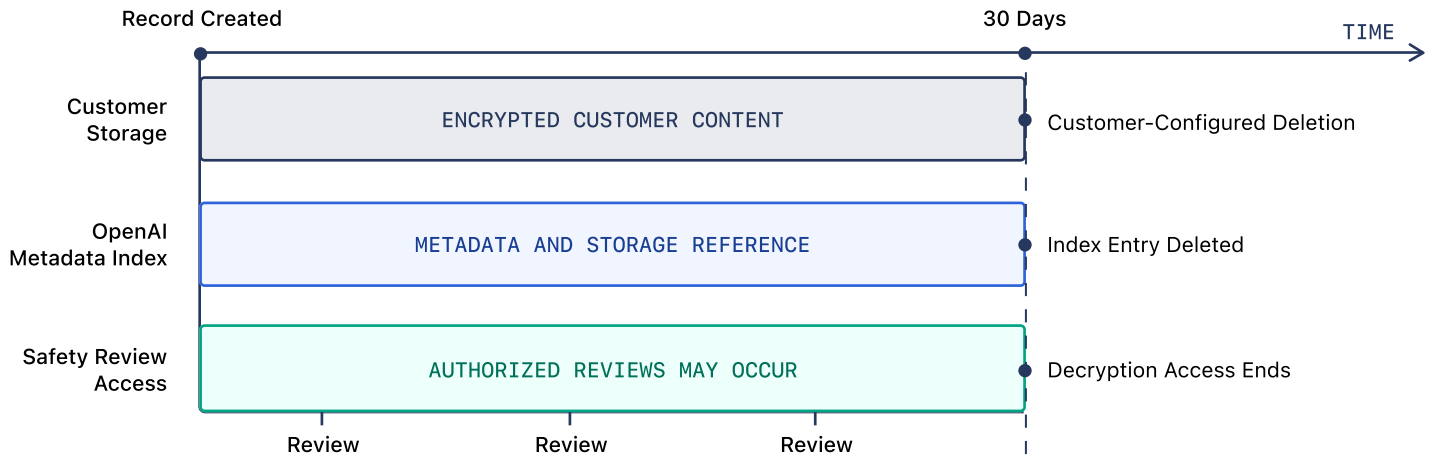
An OpenAI-hosted variation, Private Retention with PSP, is described in Appendix A. It uses substantially the same protected-review controls but retains encrypted safety records in OpenAI-managed infrastructure and is not compatible with ZDR.

Customer Content Lifecycle

Customer Content Retention

Customer content is encrypted before retention in a customer-controlled storage environment for asynchronous safety review. Customers configure their storage environment to retain the encrypted data for at least 30 days.

Customer Content Lifecycle



Section 3

Harm Areas

Customer content is referred for two reasons:

01 A safety classifier triggers during the API request. OpenAI uses automated classifiers to detect requests involving high-risk cybersecurity or biological research. It also scans for additional issues such as potential API key leaks or distillation attacks.

02 A sampling rule selects an interaction during the API request. In order to improve the detection and accuracy of dual-use areas such as cybersecurity, OpenAI can sample up to 100% of traffic for asynchronous review. The exact maximum sampling rate varies by product offering and can be significantly lower. Sampling provides additional context for distinguishing legitimate activity from misuse and reducing unnecessary disruption for customers.

A safety referral does not establish a policy violation.

Data Residency and BAA Coverage

Private Safety Processing supports both Data Residency and Regional Processing as described in the OpenAI documentation. If a customer has opted into Data Residency, they must provide a supported Cloud Service Provider (CSP) storage container for each region where their records are stored.

If a customer has not configured the appropriate regional storage container for a request, or if there is an error writing to the appropriate regional storage container, data is not persisted. Traffic without a geographic constraint is routed through the default processing path.

Private Safety Processing is designed to support use of eligible OpenAI API services under an applicable Business Associate Agreement (BAA) to help support HIPAA compliance. Customers remain responsible for configuring their storage and access controls and meeting their own HIPAA obligations.

Customer Metadata

OpenAI maintains a separate metadata index to locate customer-controlled records and coordinate the automated safety review. The index contains operational metadata and storage references, and does not contain customer content in any form.

Index fields can include organization or project scope, provider, processing region, customer storage location or object reference, content identifier, timestamp, expiration, optional safety identifier, and approved routing or policy metadata.

Offline Access

Asynchronous safety reviews can process retained records during their 30-day decryption window. The Safety Review Scheduler (SRS) periodically scans the metadata index for risk signals such as an anomalous uptick in cybersecurity-classified content. If such a signal is identified, the Asynchronous Safety Pipeline is triggered.

During setup, customers authorize the OpenAI Customer Storage Gateway (CSG) to retrieve encrypted safety records. Subsequent automated reviews rely on these permissions while those permissions remain valid.

Customer content that is fetched and decrypted in this flow is treated as highly sensitive and processed in memory for the explicit purpose of the safety review. The content is decrypted in a hardened, attested Zero Operator Access (ZOA) environment (Section 6).

Deletion

OpenAI’s ability to decrypt customer content automatically expires after 30 days, whether or not the records are deleted from the customer-controlled storage container. OpenAI's workload-scoped HPKE keys follow the existing daily rotation and 30-day destruction schedule. SRR enforces the original key-expiration time even if a key was released into memory before the corresponding vault key was destroyed. The entry in the metadata index is also deleted after 30 days.

Customers are responsible for configuring their storage container to delete data after no less than 30 days. As part of the customer storage registration process, OpenAI verifies that the storage container is configured with at least a 30-day retention period.

Customer Content Encryption

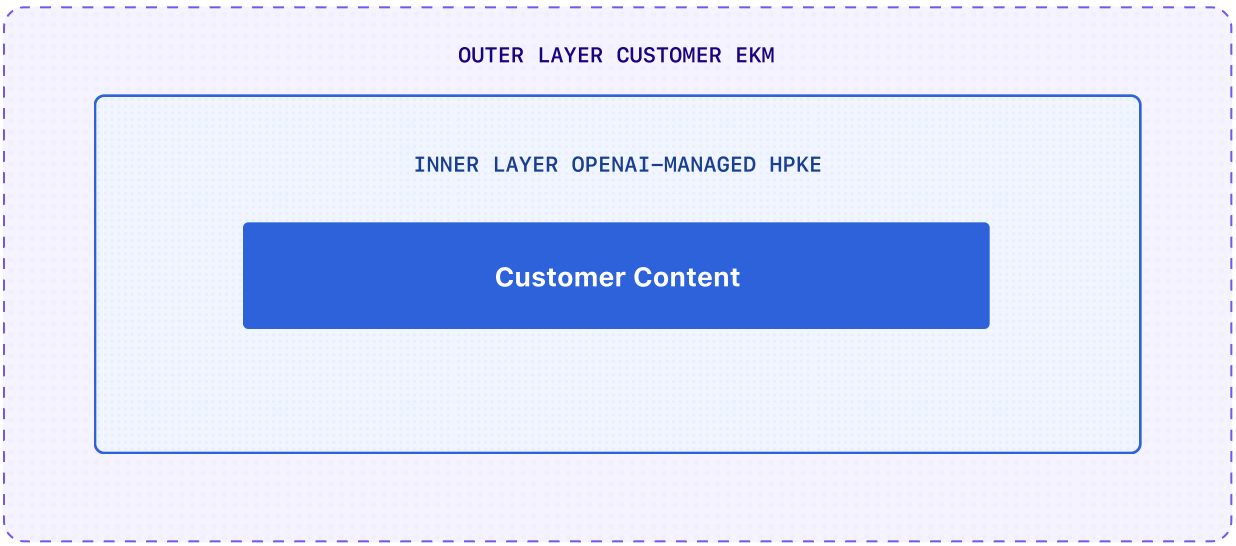
The protection path doubly encrypts a transcript referred for asynchronous safety review before CSG writes it to the customer's approved regional CSP storage container. The two encryption layers are:

01 An inner layer of encryption with OpenAI-managed keys

02 An outer layer of encryption protected through customer-managed EKM

If encryption fails, the system blocks the write to customer storage. It does not create a plaintext record, store an OpenAI-managed payload copy, or use an unauthorized storage fallback. For the asynchronous safety pipeline to run, both layers of encryption must be decrypted. Decryption to plaintext is only possible in the Safety Review Runtime which disables human access (see Section 6 for more details).

Two Encryption Layers



Section 4

Inner HPKE Encryption Envelope

The inner layer protects each safety record using Hybrid Public Key Encryption (HPKE). P-256 establishes a shared secret, HKDF-SHA256 derives encryption keys, and AES-256-GCM encrypts the content and detects tampering.

The versioned envelope authenticates its version, key-selection scope, algorithm, public-key thumbprint, and expiration. Changing any of these fields causes decryption to fail.

Inner safety record envelope

```
{
  "version": "v3",
  "scope": "global",
  "key_thumbprint": "<openai-public-key-thumbprint>",
  "algorithm": "HPKE-v1-DHKEM_P256_HKDF_SHA256-AES256_GCM",
  "ciphertext": "<base64url-encoded-encrypted-customer-content>",
  "encrypted_dek": "<base64url-encoded-HPKE-encapsulated-key>",
  "content_id": "<unique-content-id>",
  "expires_at": "2026-09-18T12:00:00Z"
}
```

In this envelope, encrypted_dek contains the HPKE encapsulation value. The key-selection scope identifies the shared key domain, but it does not identify an individual customer.

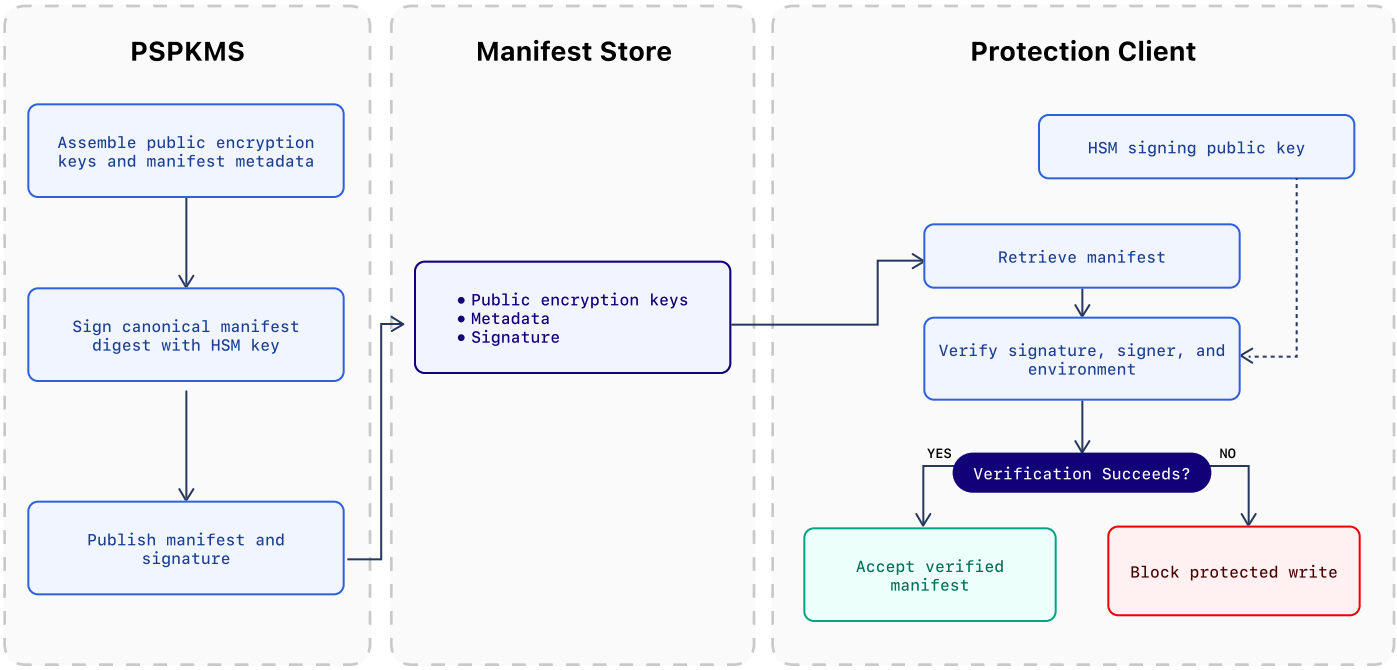
Key Distribution and Manifest Verification

The Private Safety Processing Key Manager Service (PSPKMS) generates workload-scoped HPKE key material, distributes it to the configured cluster vaults, and publishes the corresponding public keys in a signed manifest. The same HPKE key can serve multiple customers; customer-specific encryption is handled by the outer EKM layer.

A separate, non-exportable signing key held in a hardware security module (HSM) authenticates the manifest. Its signature binds the manifest metadata, ordered key entries, intended deployment environment, and signer identity.

The approved PSPKMS workload can publish and sign manifests; human access groups do not write or sign them. The protection path verifies the manifest signature, trusted signer, and intended environment before accepting new keys. An invalid manifest is rejected, and previously trusted keys remain usable only when they are valid.

Manifest Publication and Verification



Section 4

Customer-Managed Outer Encryption (EKM)

The outer encryption layer gives the customer control over an additional decryption authorization via OpenAI EKM. An OpenAI-generated data encryption key (DEK) encrypts the HPKE envelope. The customer's key-management service (KMS) wraps the DEK using a customer-controlled key encryption key (KEK). The encrypted record is stored with the wrapped DEK, while the KEK remains outside OpenAI's systems.

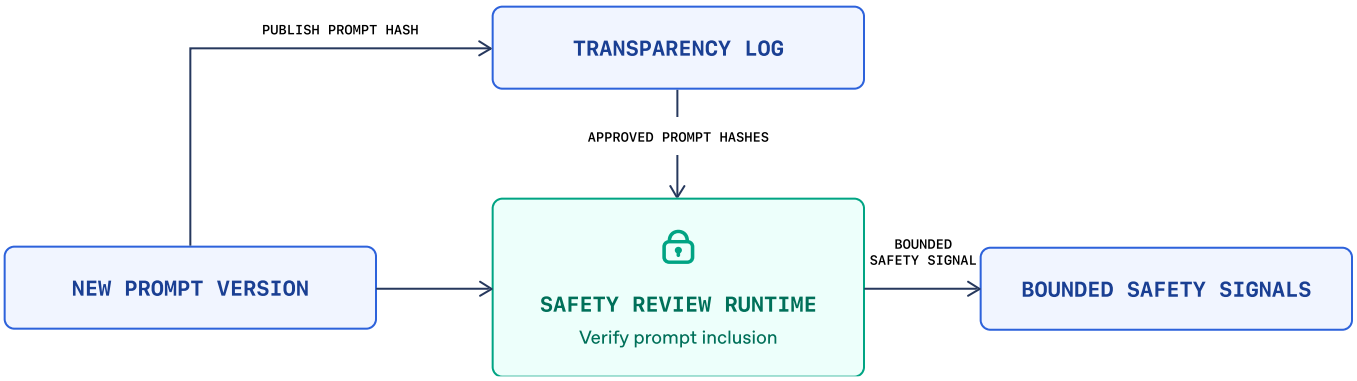
To decrypt the outer layer, the asynchronous safety pipeline uses customer-authorized access to the KMS to recover the DEK. The result is still an HPKE encrypted record; access to plaintext also depends on release of the OpenAI-managed inner key to the approved runtime.

Revoking OpenAI's authorization to use the customer-managed key disables OpenAI's ability to decrypt retained customer content once the revocation takes effect. Revocation does not delete encrypted records or undo processing that has already occurred. For more information, refer to the EKM Technical FAQ.

Bounded Safety Signals

Automated safety review uses a reviewer prompt (OpenAI's instructions for assessing customer content) to define the safety review and an output schema to define the result. The schema identifies the fields that the Safety Review Runtime (SRR) may return in plaintext. SRR independently validates those fields against its runtime rules. The separation between the reviewer prompt and the output schema enables safety analysis to evolve while preserving the disclosure limits of the output schema.

Prompt Iteration and Bounded Safety Output



Prompt Approvals and Transparency

Reviewer prompts are versioned and managed through an internal dashboard. All prompt changes require approval by another authorized reviewer before publication, and prompt history records the prompt, its author and approver, and the approval time.

After approval, the prompt publication process records a cryptographic hash of the reviewer prompt in a tamper-evident transparency log. Before using a prompt for safety analysis, the SRR hashes it and verifies that the hash exists on the transparency log. A missing hash or a tampered transparency log prevents the safety review from proceeding.

The transparency log ensures that only approved prompts are used by the Safety Review Runtime. This control makes prompt changes auditable by authorized personnel.

Section 5

Output Schema

There are three types of data that can leave the Safety Review Runtime:

01 Plaintext signals

Before inference, SRR uses the input schema annotations to identify the fields proposed for plaintext output. It validates those fields against an independently enforced set of runtime rules. These rules restrict the number of fields and permit only explicitly bounded, discrete, fixed-shape fields to leave in plaintext. Free-form strings, transcript excerpts, model explanations, and unbounded values are rejected.

02 Full reviewer result and summaries

The full reviewer result and any associated summaries are encrypted with the same encryption scheme and time to live (TTL) as the original customer content. This data is used as context for future asynchronous safety pipelines. Since this data is derived from customer content, it is stored in customer-controlled storage and not persisted within OpenAI infrastructure.

03 Standard request metadata

Contains additional fields such as request ID, timestamp, time to first token (TTFT), or any other operational fields that are explicitly approved and do not contain any customer content.

Example safety review schema

```
{
  "type": "object",
  "properties": {
    "decision": {
      "type": "string",
      "enum": ["allow", "block"],
      "x-srr-public": true
    },
    "signals": {
      "type": "array",
      "items": {
        "type": "string",
        "enum": ["bio", "cyber", "distillation"]
      },
      "maxItems": 4,
      "x-srr-public": true
    },
    "reviewer_notes": {
      "type": "string"
    }
  },
  "required": ["decision", "signals", "reviewer_notes"],
  "additionalProperties": false
}
```

Example safety review result

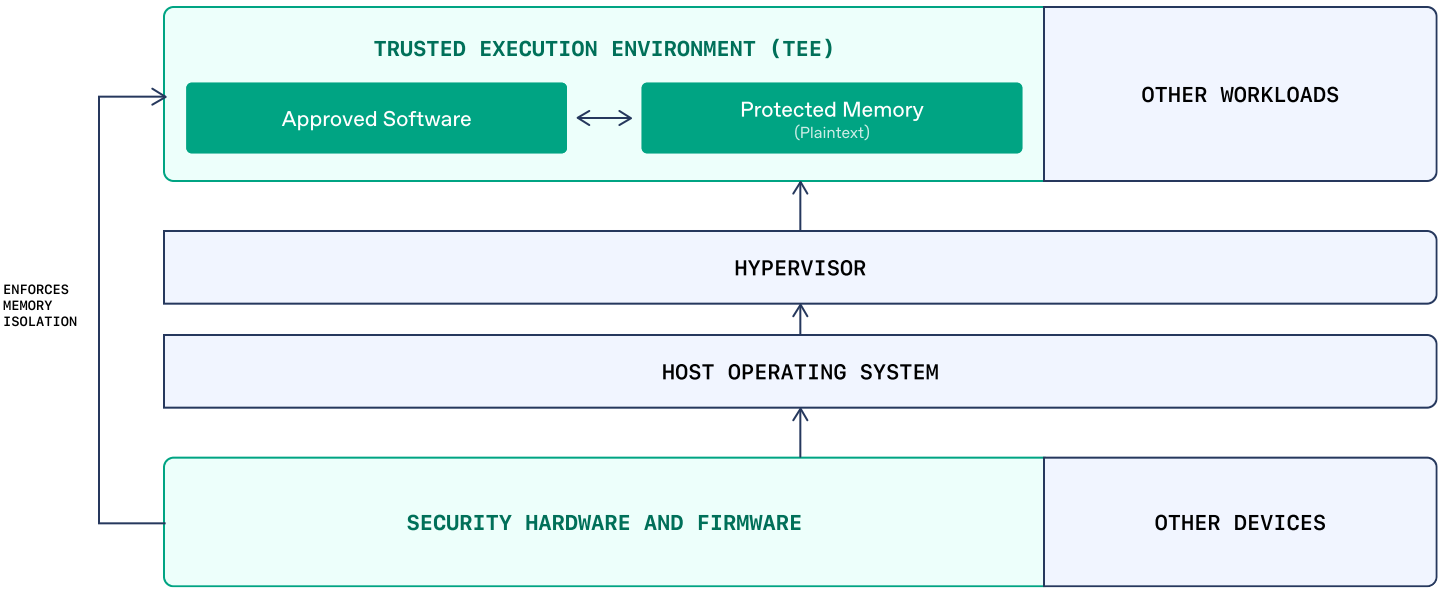
```
{
  "plaintext_public_structured_output_json": {
    "decision": "block",
    "signals": ["bio"]
  },
  "encrypted_full_structured_output": "<encrypted-complete-reviewer-response>",
  "public_metadata": {
    "responses_api_response_id": "resp_01JEXAMPLE"
  }
}

```

Confidential Computing

Confidential computing protects data during processing by running software inside a hardware-backed trusted execution environment (TEE). The TEE isolates protected memory from infrastructure outside that environment, including human access. Approved software inside the TEE can process plaintext.

Private Safety Processing combines this isolation with controlled key release to create an environment in which specific confidential workloads can decrypt customer content and process it for safety review.

Confidential Computing



Attestations

Attestation provides cryptographic evidence of a workload's execution environment and security configuration. In PSP, the attestation evidence identifies the confidential computing enforcement (CCE) policy applied to the workload.

The CCE policy specifies the permitted container contents and execution configuration. Its cryptographic hash is included in the hardware-backed attestation evidence. To receive decryption keys, SRR provides this evidence to the key vault through a process called Secure Key Release.

Section 6

Secure Key Release

OpenAI's inner decryption keys are stored as hardware-backed keys in a key vault. Secure Key Release (SKR) permits their release into an authorized confidential environment when two independent conditions are satisfied:

01 **Workload authorization.** The requesting identity has permission to release the key.

02 **Environment verification.** Attestation evidence from a trusted authority satisfies the key's release policy, including an authorized CCE policy hash.

After successful verification, SKR transfers the key material into the attested environment for use by the approved workload. SRR caches released keys in protected memory. SRR separately checks key validity and expiration when using cached keys to respect the 30-day expiration window. Decryption using customer-managed EKM is a separate process and is not related to SKR.

Key Vault Configuration

SRR runs in dedicated private clusters, each of which has a cluster-local key vault. The vault configuration disables public network access, denies unapproved network access, enables purge protection, and records audit events. Key-management permissions remain separate:

- SRR can read authorized key metadata and request release.
- PSPKMS can provision keys and manage release policies.
- Approved deletion operators can delete or purge keys, but are not given the ability to release or cryptographically use them.
- No human access groups have the key-release role.

Deployments and Change Control

Release preparation records the source revision, resolves application and supporting images to cryptographic hashes, and generates the corresponding CCE policy.

For SRR, the deployment workflow works as follows:

01 Records the authorized CCE hash in a tamper-evident transparency log.

02 Directs PSPKMS to update the applicable key-release policies.

a. PSPKMS verifies that the input CCE hash is on the transparency log.

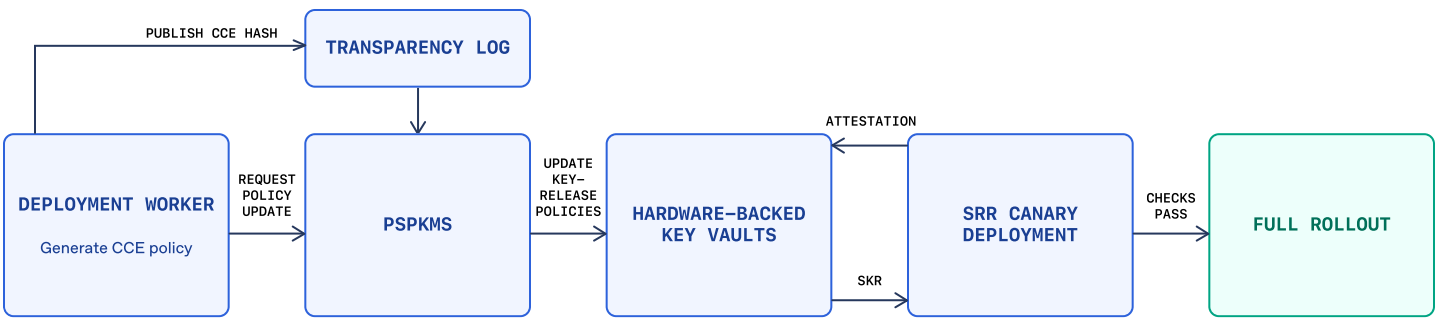
b. PSPKMS updates the key-release policies of existing keys so that the new service is authorized to release decryption keys via SKR.

03 Runs a limited canary deployment, including key loading and a safety review on encrypted test data.

04 Proceeds with the full rollout.

Any change to the source code, egress policies, or any other configuration of SRR triggers a full deployment flow. PSPKMS uses a separately controlled deployment workflow. Its release artifacts also identify the source revision, image hashes, and CCE policy. As with SRR, the PSPKMS CCE policy hash is recorded in a tamper-evident transparency log.

SRR Deployments and Change Control



Section 6

Workload Transparency

Workload transparency covers the approved SRR and PSPKMS workloads, where each authorized CCE policy is recorded on a transparency log. The corresponding workload artifacts and measurements are recorded in blob storage.

Authorized auditors can inspect which SRR and PSPKMS workload configurations were approved, the conditions under which they were authorized, and changes to the associated software or policies. These records do not contain customer content.

Collaborative Safety Investigations

The organizations we work with handle some of the most sensitive information in their sectors, including financial records, health data, confidential business plans, and proprietary research. Protecting that information is essential to meeting regulatory obligations, maintaining customer trust, and preserving their competitive advantage.

When an asynchronous safety review identifies a concern that warrants further investigation, OpenAI can send a notice to the customer through the appropriate channel. This notice is not an enforcement decision but the start of a collaborative effort in which both parties work together to resolve the potential concern.

Customers may choose to provide context or content to help with the investigation. OpenAI personnel may review that information alongside the permitted safety signals returned by the automated review.

This safety collaboration does not give OpenAI employees access to the original encrypted records or detailed reviewer results. Those records can be reprocessed only through the approved automated reviewer path, with the same data residency, storage, and encryption controls. Reprocessing is possible while the records remain available, customer storage and EKM authorizations remain valid, and the OpenAI decryption key has not expired.

Collaborative Safety Investigation

Guardrails

Private Safety Processing applies independent controls to operational logging, key management, and the safety processing environment. These controls limit what automated reviewers can access and disclose, including when customer content contains malicious instructions or a reviewer behaves unexpectedly.

Observability

SRR uses a structured, allowlisted logging policy. Only approved events and fields may be emitted; unrecognized events are blocked and unapproved fields are removed. This prevents application changes or errors from introducing customer content into operational logs. Logs may include service health, request volumes, attestation and key-release status, approved safety-decision categories, and sanitized failure classifications. These logs exclude customer content, safety reviewer explanations, sensitive identifiers, and personally identifiable information.

Retryable failures involving customer storage, customer-managed keys, OpenAI-managed keys, inference, or infrastructure are recorded using approved error classifications without exposing customer content. Invalid, revoked, expired, or permanently undecryptable inputs are not decrypted.

We also further restrict log access to authorized operators, and filter unstructured output before it leaves the protected workload.

Non-Targetability

OpenAI-managed HPKE keys are shared across customers. Inner-key selection, rotation, and release policies operate at the shared workload level and do not depend on customer identity. The HPKE envelope and its authenticated associated data do not include organization or project identifier fields. The encryption path does not add customer identity markers to ciphertext or record identifiers. This separation limits the use of customer identity within the Safety Review Runtime.

These protections do not prevent an authorized workflow from selecting a customer's records or prevent the content itself from containing identifying information. When customer-managed EKM is applied, a separate outer encryption layer uses a customer-specific EKM scope. That layer does not change the shared HPKE key or its release policy.

Section 8

Safety Processing Environment

The safety processing environment treats customer content as untrusted. Instructions contained in a reviewed record cannot grant additional data access, enable tools, change network destinations, or modify the review policy. Network policy limits communication to approved dependencies. The automated reviewer cannot use its response to execute arbitrary commands or select a destination for customer content.

Section 9

Customer Visibility

Eligible authorized administrators with Compliance API access can inspect record-level safety-retention lifecycle events. These events describe when and why a record is created, accessed, or expires without exposing customer content. Customers must request and be granted Compliance API access.

- **RETENTION_CREATED:** Includes a stable content identifier, an approved high-level retention reason, and a customer-provided safety identifier when supported.
- **CONTENT_ACCESSED:** Uses the same content identifier and includes approved access context for a protected safety review.

Approved high-level reason categories can include CYBER, BIO, SAMPLED, and SUSPECTED_ABUSE. These labels describe broad retention context; they do not disclose the transcript, underlying classifier evidence, or detailed reviewer reasoning.

The same content identifier links subsequent access and expiration events to the original retention event. Customer-side cloud-storage and key-management audit logs provide additional evidence of customer-authorized object access, subject to provider configuration and logging availability.

Example customer-visible retention event

```
{
  "event_id": "<unique-event-id>",
  "type": "PRIVATE_SAFETY_PROCESSING",
  "principal": {"id": "org_01JEXAMPLE", "type": "API_PLATFORM_ORG"},
  "actor": {"type": "SYSTEM", "id": "openai_safety_system"},
  "timestamp": "2026-08-11T12:00:00Z",
  "event_type": "RETENTION_CREATED",
  "event_details": {
    "detail_type": "RETENTION_CREATED",
    "content_id": "<unique-content-id>",
    "reason": "CYBER",
    "safety_identifier": "<customer-provided-safety-id>"
  }
}
```

Compliance

OpenAI services offered to business and developer customers under the OpenAI Services Agreement, including the API platform, are audited internally at least annually and assessed by independent third-party auditors. SOC 2 Type 2 and SOC 3 reports have been issued to OpenAI. OpenAI maintains industry-recognized security certifications and accreditations, including ISO/IEC 27001, 27017, 27018, 27701, and 42001.

Customers can review audit reports, security assurance documentation, and CAIQ, SIG, VSA, and HECVAT questionnaires through the OpenAI Trust Portal. Contractual commitments governing the processing and protection of customer data are described in OpenAI's Data Processing Addendum (DPA).

These measures can help support customer compliance with applicable privacy regulations, including the EU General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). Additional audits and penetration testing are in progress.

Government User Data Request Policy

OpenAI follows its Government User Data Request Policy when governments ask for user data. OpenAI discloses data about the users of our service only when there is a valid legal process that is consistent with international human rights laws, principles, and norms, or if OpenAI has a good faith belief that there is an emergency involving a danger of death or serious physical injury to a person and that the user data is necessary to prevent that harm. OpenAI provides only the data specified in the legal process.

OpenAI reserves the right to notify the affected users of any demand we receive (unless we are prohibited by law or exceptional circumstances apply). With PSP, OpenAI personnel can access customer content only as ciphertext.

Planned Enhancements

Future work is planned to extend our core Private Safety Processing protections to include additional services, operational dependencies, and customer-facing controls.

Private Inference

We expect to extend confidential-computing protection beyond the protected safety-inference stack when necessary technology, capacity, and product commitments are met. This will extend operator access protections to the rest of the plaintext-handling components of the Asynchronous Safety Pipeline.

Improved Safety Collaboration

As designed, eligible administrators may review why a safety record was retained through the Compliance API. We expect to expand the scope of these records to include additional information like classifier scores so that trusted customers can ingest these signals into their own safety pipelines.

Limits Across Repeated Reviews

PSP currently does not impose a limit on the total information disclosed across repeated reviews of the same content. Compliance API events and cloud provider audit logs provide visibility into when content is accessed for safety review. We expect to add protections that limit cumulative disclosure.

Glossary

Term	Definition
Private Safety Processing	An abuse-monitoring mode that protects API content referred for asynchronous safety review from human review. Private Safety Processing is a distinct exception path for eligible traffic, not a synonym for ZDR.
Zero Data Retention (ZDR)	A data-control mode intended to avoid normal abuse-monitoring retention.
ZDR with PSP	A customer-controlled safety architecture that requires customer-managed EKM and a regional customer-controlled cloud storage container. OpenAI performs approved automated safety review while retaining approved metadata and storage references, but not customer content or encrypted customer payloads.
Private Retention with PSP	The OpenAI-hosted variation described in Appendix A. It retains encrypted safety records in OpenAI-managed infrastructure and is not ZDR.
Cloud Service Provider (CSP)	A provider of cloud infrastructure, including customer-controlled storage, regional access controls, audit logging, and supported key-management services.
Customer-controlled storage	A cloud storage container owned and administered by the customer, in the applicable processing region, that holds encrypted safety records and remains subject to customer-controlled access and lifecycle policies.
Customer Storage Gateway (CSG)	An OpenAI-operated service that uses customer-authorized cloud identities to read and write encrypted records in customer-controlled storage.

Safety Referral	The process of routing eligible API content to an automated safety review.
Safety Agent	The automated reviewer that evaluates referred content for an approved safety purpose.
Safety Review Runtime (SRR)	The approved confidential environment that decrypts a protected safety record, invokes the Safety Agent, encrypts the detailed result, and returns bounded safety decisions.
Encrypted Safety Record (ESR)	The encrypted customer-controlled record containing content referred for safety review and the approved fields required to process it.
Safety Review Scheduler (SRS)	The workflow that schedules protected safety reviews using approved metadata and passes bounded safety decisions to automated enforcement.
Bounded safety decision	A predefined safety classification that automated enforcement can use without receiving the original customer content or detailed reviewer output.
Private Safety Processing Key Manager Service (PSPKMS)	The service that manages OpenAI's workload-scoped HPKE keys, provisions approved per-cluster vaults, rotates and destroys key material, manages release policies, and publishes signed public-key manifests.
Hybrid Public Key Encryption (HPKE)	The inner encryption envelope, using the IETF P-256 / HKDF-SHA256 / AES-256-GCM suite.
Secure Key Release (SKR)	The cloud mechanism that checks confidential-compute attestation and key-release policy before returning private-key material to an approved workload.

Enterprise Key Management (EKM)	A customer-controlled outer encryption envelope. A key-encryption key in the customer's supported key-management service protects the storage data-encryption key.
Customer-controlled storage and customer-managed encryption (EKM)	Separate controls: customer-controlled storage governs where encrypted records reside and whether they can be retrieved; customer-managed EKM independently governs authorization to decrypt the outer encryption envelope.
Compliance API	The customer-facing interface through which eligible, authorized administrators can inspect approved record-level retention and access events when their organization has access.
Zero Operator Access (ZOA)	A protection that prevents infrastructure operators from accessing protected customer content on the covered systems.

Private Retention with PSP

Private Retention with PSP is for eligible customers that do not use a customer-controlled cloud storage container. Encrypted safety records reside in OpenAI-managed infrastructure, so this offering is not ZDR.

How the Storage Boundary Differs

Eligible first-party traffic follows the same approved safety-referral process and uses the same OpenAI-managed HPKE encryption described in the main architecture. Instead of writing encrypted safety records to a customer-controlled regional container, the approved storage path writes them to OpenAI-managed encrypted safety storage that respects data residency requirements.

OpenAI-managed storage retains the encrypted customer payload and the approved metadata required for asynchronous review. Customers do not provide or control the storage container in this architecture. Supported customers can add customer-managed EKM, but that optional configuration does not make the offering ZDR.

Under Private Retention with PSP, the Compliance API emits a lifecycle event with detail type RETENTION_EXPIRED after the configured retention period ends.

Appendix A

Shared Protections

The Private Retention with PSP variation retains substantially the same protection controls:

- OpenAI HPKE encryption before durable storage.
- Signed OpenAI key manifests and workload-scoped HPKE keys.
- Daily inner-key rotation and key destruction after 30 days.
- Hardware-backed attestation and secure key release to an approved SRR workload.
- Encrypted detailed reviewer output and bounded enforcement decisions.
- Compliance API lifecycle events when the customer has access.
- Supported appeals flows that do not provide OpenAI access to customer content.
- Approved operational logging that excludes protected customer content.
- Optional Private Inference only for customers that meet the applicable eligibility and capacity requirements.

The customer still does not supply the OpenAI inner public encryption key. Customer-managed EKM, when supported and enabled, adds an independent outer encryption control but does not change the OpenAI-managed storage boundary.