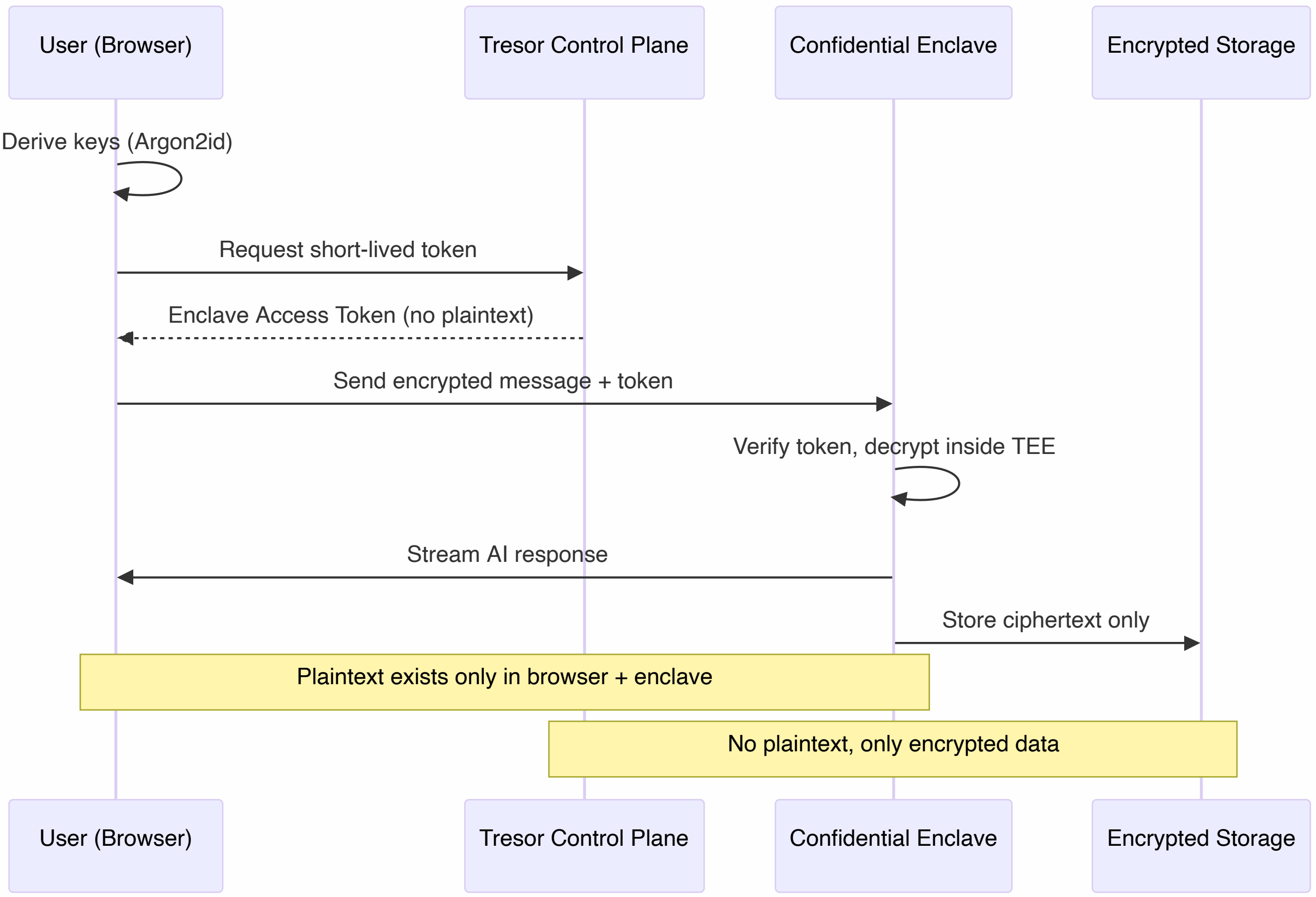
Zero-Access AI Conversations: How Tresor Protects Your Privacy

Executive Summary

Tresor is built on a simple promise: **your conversations belong to you, not us.** Every message you type is protected by end-to-end encryption and processed only inside secure computing environments that even Tresor cannot inspect. Teams can now collaborate inside shared workspaces without ever handing Tresor access to their plaintext. This whitepaper explains the principles and safeguards behind Tresor's zero-access design, showing how we deliver practical confidentiality without trade-offs in usability.

Architecture at a Glance



This simple flow illustrates Tresor's zero-access model: only your browser and the secure enclave ever handle plaintext, while the control plane and storage remain blind to your content.

Zero-Access Principles

Tresor's architecture reduces the trust you place in us to the absolute minimum:

- **Client-side encryption:** All chat content is encrypted in your browser before it ever reaches our systems.
- **Confidential enclaves:** AI processing runs in attested secure hardware (Trusted Execution Environments), preventing even infrastructure operators from peeking inside.
- **Content-blind orchestration:** Our control plane manages identity, quota, and policies — but never touches your plaintext.
- **Cryptographic accountability:** Every streamed response carries verifiable receipts proving it was generated inside the expected secure environment.

The result: Tresor staff, service providers, and even attackers with database access cannot read your chats.

How Encryption Protects Your Conversations

At the core of Tresor's zero-access model is **key material owned by the user**:

- A strong passphrase in your browser derives a master key using **Argon2id**, a modern memory-hard key derivation function.
- From this, unique per-chat keys are created to encrypt every title and message with **XChaCha20-Poly1305 authenticated encryption**.
- These keys never leave your device unprotected. Tresor's servers only store ciphertexts and encrypted keys, never raw secrets.
- Optional **recovery codes** allow you to regain access if you forget your password, without granting Tresor backdoors into your data.

Plaintext lives only in your active browser memory long enough to stream a conversation, then disappears.

Confidential Compute & Attestation

When you start a conversation, your browser connects directly to Tresor's confidential computing fabric:

- A **short-lived access token** proves your identity and permissions, but does not contain your message.
- The enclave verifies this token and runs your prompt securely inside hardware-enforced isolation.
- **Remote attestation** produces cryptographic evidence of the enclave's integrity, embedded in receipts you can verify.
- Every streamed output is tied to digests of your request and response, ensuring tamper-proof

accountability.

This means your sensitive data is only ever visible inside hardware we can prove is locked down to the right code and policies.

Data Handling & Operational Safeguards

Zero-access security is not just about cryptography — it's also about disciplined operations:

- **Encrypted storage only:** Titles, messages, keys, and recovery bundles are written as ciphertext with associated metadata.
- **Strict validation:** Our APIs reject any attempt to submit plaintext, preventing accidental leaks.
- **Transport security:** Mutual TLS, strict CORS rules, and optional certificate pinning ensure safe browser-to-enclave connections.
- **Signed error envelopes:** Even failure cases are cryptographically signed, so tampering or policy violations can be detected.

Together, these safeguards make it practically impossible for unauthorized parties — including Tresor itself — to read or alter your conversations.

Secure Workspaces & Encrypted Sharing

Zero-access guarantees now extend beyond individuals to entire teams:

- **Workspace-scoped encryption domains:** Each workspace derives its own master key hierarchy, keeping projects cryptographically isolated from one another.
- **Per-member key wrapping:** Invitations deliver encrypted key shards that only the recipient's passphrase-unlocked device can unwrap, so Tresor never sees the underlying workspace keys.
- **Granular roles without plaintext exposure:** Admins can manage permissions, retention, and quotas while the system enforces that only authorized members decrypt shared conversations.
- **Shared history that stays private:** Titles, messages, and attachments remain end-to-end encrypted even when multiple collaborators work in the same project.

The result is collaborative AI assistance that feels natural for teams while preserving the zero-access barrier that defines Tresor.

Why It Matters

For technology leaders, this design means you can adopt AI without introducing hidden data-exposure risks.
For organizations, it aligns with strict compliance requirements by ensuring sensitive inputs never leave your control.
For individuals, it brings the confidence that your private ideas, plans, or research are yours alone.

In short: Tresor lets you benefit from AI while keeping confidentiality at the center.

Looking Ahead

Tresor’s roadmap builds on the same zero-access foundation:

- Automatic key rotation and streamlined recovery options.
- Deeper integration of attestation into collaboration and audit trails.
- Privacy-preserving retrieval and web search: upcoming RAG (retrieval-augmented generation) and confidential web search capabilities will let users enrich conversations with verified external knowledge — all processed inside secure enclaves, so no plaintext ever leaves the trusted boundary.
- Unified context management: projects, ephemeral chats, and web results will share a single verifiable retrieval pipeline, allowing transparent citations without compromising privacy.

Our commitment remains unchanged: **we enable confidential collaboration without ever taking possession of your content.**

Authors

Tresor Security & Infrastructure Engineering Team
October 2025