



## THE AUTOMATIC AND UNIFIED REFERENCE APPLICATION

NAME OF PROJECT:

TAURA

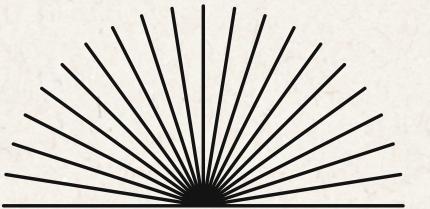
PRESENTED BY:

ADITYA SINGH 23BCI0035  
RISHIT SHIVAM 23BKT0020  
HARSHIT NARANG 23BKT0022  
ISHAAN S 23BME0453  
ADHEESH GARG 23BCT

VIT VELLORE

THEME ID:

4. AI for Core Applications



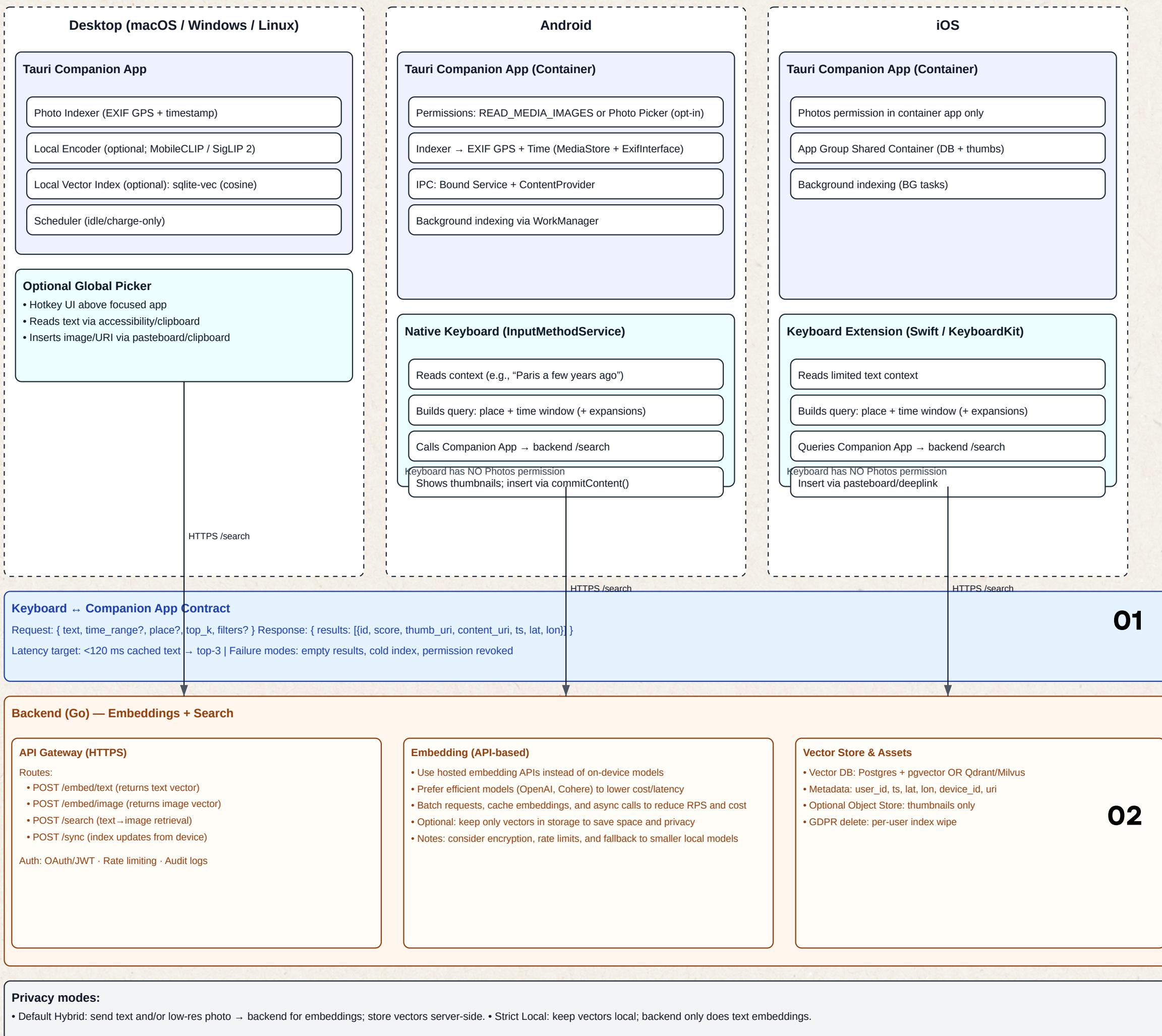
# PROBLEM STATEMENT

People scatter their personal knowledge across screenshots, photos, lecture PDFs, chat logs, memes, bookmarks, and voice memos. When they need to recall one specific fact or item, searching becomes frustrating—slow, error-prone, and often unsuccessful.

Current LLM-based tools either hallucinate or cannot reliably ground answers in a user's private files. This undermines trust and limits their usefulness. Meanwhile, system-level search remains rigid, keyword-only, and blind to context.

Important personal knowledge becomes siloed, scattered, and practically unusable at the moment it's most valuable. When they actually need to retrieve something “Which PDF had that formula?”, “Where did I save that screenshot?”, “Didn't I note this in a chat somewhere?” the process breaks down.

Making context sensitive OS capabilities to allow for integrated daily functioning, optimizing workflows, improving efficiency and making organised collections

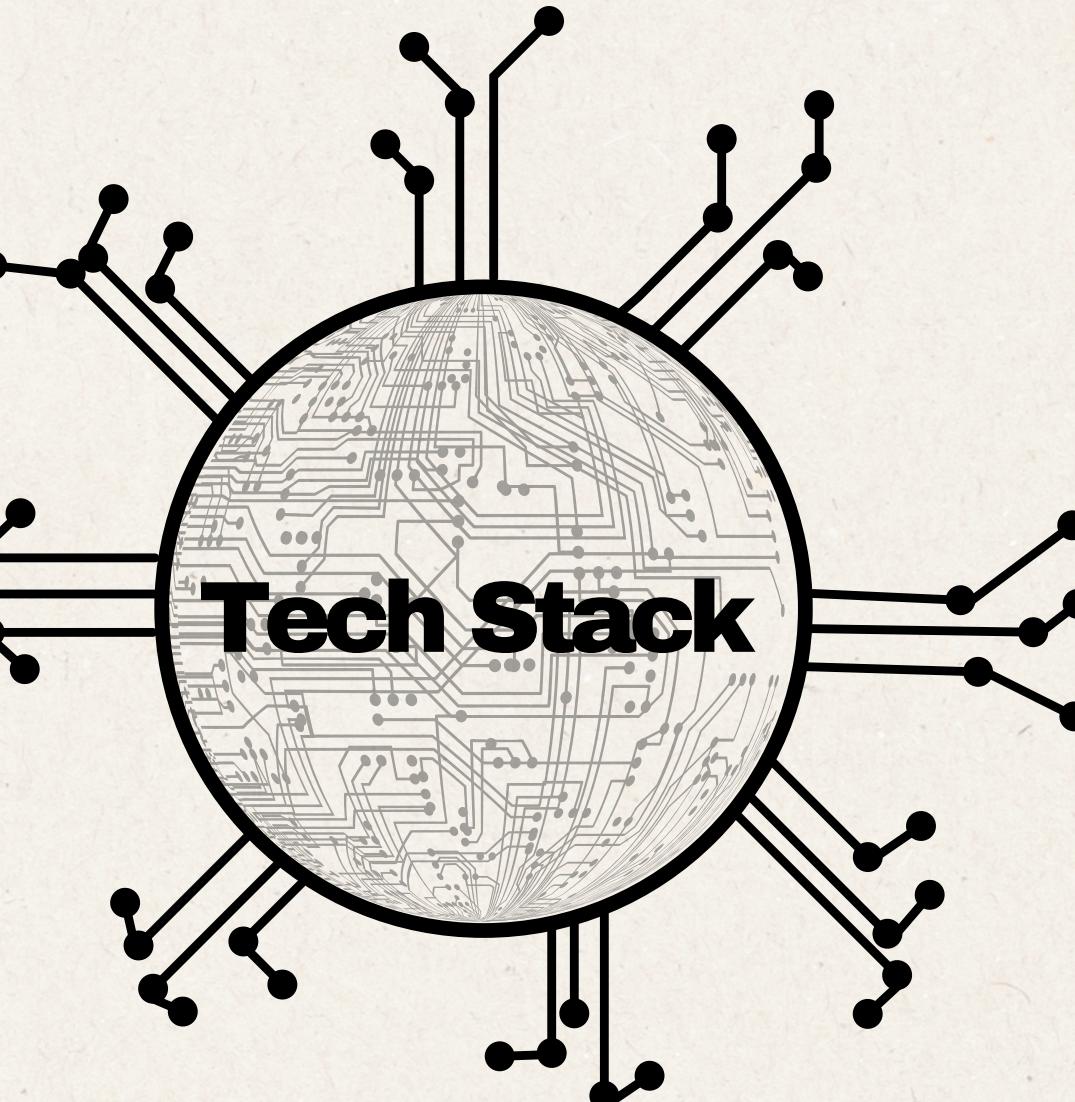


# Proposed Solution

The main application is built with Tauri, making it cross-platform across desktop, Android, and iOS. Tauri acts as the central engine that handles multimodal access to data, embedding generation with a CLIP-like model, and fast retrieval using a local and server-side vector databases. By keeping all the heavy lifting inside Tauri, the system ensures a single codebase, efficient maintenance, and privacy by design, since all computation and storage remain local to the device.

**Light native keyboard hooks – small platform-specific extensions (Android IME, iOS Keyboard Extension) forward typed text to the Tauri app and display the photo suggestions inline.**

Beyond images, Taura indexes PDFs, notes, chats, and more into a local vector database for context-aware search. Natural language queries like “that chart from last week’s lecture” instantly surface the right file. A powerful cross-platform core with lightweight native hooks makes personal knowledge retrieval seamless at the OS level.



## Client (Companion App & Keyboards)

Tauri v2 (React + TypeScript + Tailwind) UI; Rust side with tokio

- EXIF/GPS: kamadak-exif; thumbnails via image.
- Optional local vectors: sqlite-vec (cosine); DB via sqlx.
- Android IME: Kotlin + Jetpack Compose, InputMethodService, commitContent(), IPC via Bound Service/ContentProvider.
- On-device (optional) encoders: TFLite + NNAPI (Android), Core ML (iOS) for MobileCLIP/SigLIP.

## Backend (Go)

- Go 1.22 + Fiber APIs: /embed/\*, /search, /sync.
- Embeddings: ONNX Runtime Go (SigLIP-2/OpenCLIP) or gRPC → Python/PyTorch worker.
- Vector search: Postgres 17 + pgvector (or Qdrant/Milvus).
- Cache: Redis; thumbnails: S3/R2/MinIO.
- Auth: OAuth2 → JWT/PASETO; rate-limit middleware.

## Infra/DevOps

- Docker, Caddy/NGINX (TLS/HTTP3), GitHub Actions CI/CD.
- Observability: OpenTelemetry → Prometheus/Grafana, logs in Loki.
- Deploy: Fly.io or K8s (GPU pool for embedder).

## Privacy modes

- Hybrid (text/low-res to server, vectors server-side)  
or Strict Local (only text server-side).

