perplexity

Backend Blueprint for EVOLVE SaaS

Key Principle

Build once, scale later: every component below runs locally on laptops, on a $5 USD VPS, or on-prem servers – and upgrades seamlessly to Kubernetes without rewrites.

1. Core Runtimes and Languages

Layer	Language	Why	Free Toolchain
API Gateway & user-facing REST/GraphQL	**TypeScript (Node 18+)**	huge ecosystem, SSR with Next.js front-end	ts-node, Nodemon, ESLint, Prettier
AI / data services	**Python 3.12 (FastAPI)**	native LangChain/LangGraph, tons of ML libs	Uvicorn, Pydantic v2
Edge functions / Bots	**Deno** (optional)	zero-config TS/JS at the edge	Deploy on Supabase Edge Functions
Scripting / infra	**Bash, Dockerfile, Terraform HCL**	ubiquitous dev-ops stack	Docker CE, Terraform OSS

All code lives in a **Turborepo** monorepo – one `pnpm install` spins up every service in dev.

2. Service Decomposition

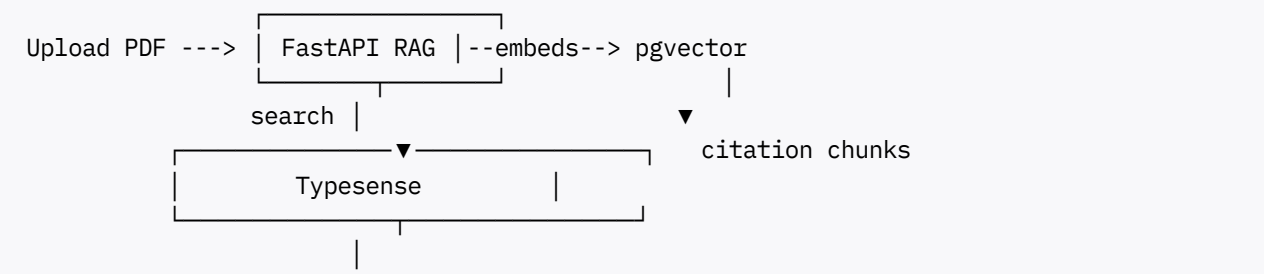
Microservice	Main Route	Libraries & Frameworks	Persistence	Notes
Auth-Service	`/auth/*`	FastAPI + fastapi-auth-jwt[1][2]	PostgreSQL (Supabase)	Password, magic-link, social OAuth; rotates RSA-256 keys nightly
User-Profile	`/users/*`	TypeORM (TS)	PostgreSQL	RLS policies via Supabase
Content RAG	`/rag/*`	LangChain 0.2 + pgvector	PostgreSQL + Redis	Splits uploads → embeddings; sticks sources for citation
LLM Gateway	`/llm/*`	**LocalAI** server[3]	–	Wraps Mixtral 8×7B, Stable-LM-Zephyr, Whisper.cpp
Realtime Hub	`/realtime` (WS)	Supabase Realtime (Postgres-CDC)	–	Live cursor & chat events
Search API	`/search`	**Typesense** self-hosted	Typesense cluster	<50 ms prefix & semantic hybrid search
Media Store	`/media/*`	**MinIO S3**[4][5][6]	MinIO Disk	Secure, versioned, signed-URL uploads

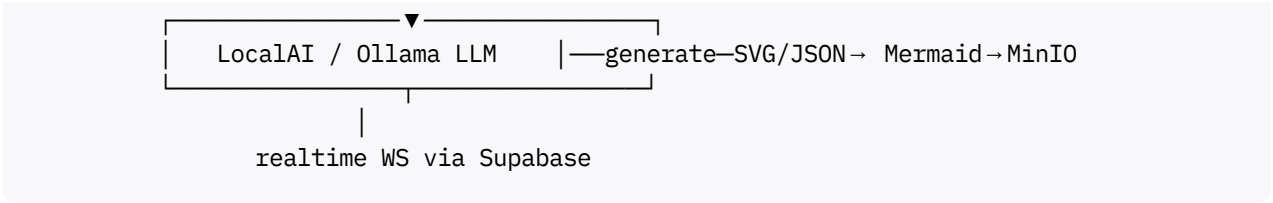
Microservice	Main Route	Libraries & Frameworks	Persistence	Notes
Queue & Tasks	`nats://`	NATS JetStream; Celery (Python)	Redis	Fan-out e-mails, transcript jobs
Observability	`/metrics`	Prometheus + Grafana	Prometheus	Exporters for Node, Python, PostgreSQL

3. Cross-Cutting Building Blocks

Need	OSS/Free Option	Why it fits
SQL + vector search	**Supabase Postgres** (500 MB free)[7] + `pgvector`	one DB handles auth, data, embeddings
Object storage	**MinIO** standalone, 50 MB binary, S3-API[4][8]	unlimited, on any disk/NAS
Full-text & typo-tolerant search	**Typesense** (Apache 2.0)	deploys in one Docker; 128 MB RAM ok
AI orchestration	**LangChain & LangGraph**	multi-step agents, retry, tracing
LLM inference	**LocalAI**[3] or **Ollama**	drop-in OpenAI-compatible, no token cost
Speech-to-text	**Whisper.cpp** or **Vosk** (offline)[9][10]	runs on CPU; batch lectures overnight
TTS	**Coqui-TTS** (MIT)	generate voice output offline
Diagram render	**Mermaid CLI**	convert LLM-generated code → SVG
Queue / cache	**Redis Stack** (Docker)	pub-sub, rate-limit, task results
Event bus	**NATS**	zero-config, <1 MB binary
API docs	FastAPI + OpenAPI	auto-docs every microservice
Secrets & config	**Doppler** free tier or `.env` + **direnv**	central secret rotation
Monitoring	**Prometheus**, **Grafana**, **Loki**	all OSS, 15-min setup
CI/CD	**GitHub Actions** free runners	build, test, push images
CDN	**Cloudflare Free**	TLS, caching, WAF

4. Data Flow Snapshot





5. Authentication & Security

1. Short-lived **JWT** (15 min) issued by Auth-Service with `RS256` keys.
2. **Refresh tokens** stored HttpOnly, rotated every 24 h.
3. **Fine-grained RLS** in Supabase: `user_id = auth.uid()`.
4. Media links signed with MinIO **presigned URL** expiring in 10 min.
5. Rate-limit every endpoint with **Redis Token Bucket** middleware.
6. All inter-service traffic over **mTLS** with Caddy certificates.

6. Dev-Ops & Deployment

Environment	Stack	Cost
Local dev	`docker compose up` (13 containers)	free
Single-node prod	**CapRover** on 1 vCPU / 2 GB VPS	≈ ₹400 / mo
Scale-out	**k3s** + Longhorn; MinIO distributed (4 × disk)	runs on Raspberry Pi cluster
Edge functions	Supabase Edge (Deno)	50 ms × 500k invocations / mo free
Backups	`wal-g` push to MinIO nightly	local disks only

CI/CD pipeline:

```
name: build

on:  push:
      branches: [main]

jobs:
  backend:
    runs-on: ubuntu-latest
    services:
      postgres:  image: supabase/postgres:15
    steps:
      - uses: actions/checkout@v4
      - run: docker compose -f docker-compose.test.yml up --abort-on-container-exit
      - uses: crazy-max/ghaction-docker-buildx@v5
        with:
          push: true
          tags: ghcr.io/evolve/api:sha-${{ github.sha }}
```

7. Approximate Monthly Cost (5 000 MAU)

Component	Option	Free Tier	Expected Usage	Cost
Supabase DB	Free Plan	500 MB	400 MB	₹0
Supabase Edge / Realtime	Free	2 GB egress	1 GB	₹0
MinIO on 100 GB VPS	Hetzner CX21	–	100 GB	₹350
LocalAI on same VPS (GPU optional)	–	–	CPU	₹0
Typesense, Redis, NATS	on same VPS	–	–	₹0
GitHub Actions	OSS minutes	2 000 / mo	800	₹0
Monitoring (Prom + Grafana Cloud free)	10k series	5k	₹0	
Backups to another VPS	rsync weekly	–	100 GB	₹200
TOTAL				**≈ ₹550 / month**

All optional cloud APIs (OpenAI, Deepgram, etc.) can be toggled off to remain 100% cost-free.

8. Drop-In Feature Modules (plug & play)

Feature	OSS Project	Integration Time
Code execution sandbox	**WASM-running "Paiza-io" fork**	1 day
Infinite canvas	**tldraw + y-js**	0.5 day
Leaderboard & badges	**Orbit-DB** or Supabase functions	0.5 day
Forum & comments	**Fider** (Go) embedded iframe	1 hour
Math rendering	**KaTeX** serverless	instant
Analytics	**Plausible** self-host	1 hour

9. Getting Started in < 10 Minutes

```
git clone https://github.com/evolve/evolve-stack
cd evolve-stack
cp .env.example .env          # set JWT_SECRET etc.
docker compose up -d          # boots Postgres, MinIO, Redis, LocalAI, Typesense
pnpm -r dev                   # gateway + FastAPI hot-reload
```

Visit:

- http://localhost:54323 for Supabase Studio

- http://localhost:9001 for MinIO Console (`minioadmin:minioadmin`)
- http://localhost:3000 for Next.js + API playground

Take-away

With nothing more than free open-source binaries and generous starter tiers, EVOLVE's backend can deliver real-time, AI-powered, multimodal learning to thousands of students at **under ₹600 per month** – and the exact same codebase scales to multi-region Kubernetes when the university grows.