

ZEGX

Egyptian Exchange intelligence, delivered on Telegram

Prepared for prospective operators & partners
Version 1.0

EXECUTIVE SUMMARY

The product in one page

ZEGX is a Telegram-native AI assistant that answers questions about the Egyptian Exchange (EGX) in Egyptian Arabic and English. It combines a reasoning agent, live web search restricted to trusted market sources, and an activation-key system that turns any Telegram chat into a private, auditable trading desk companion.

< 4s	EGX	24/7	AR / EN
Median response	Market coverage	Availability	Languages

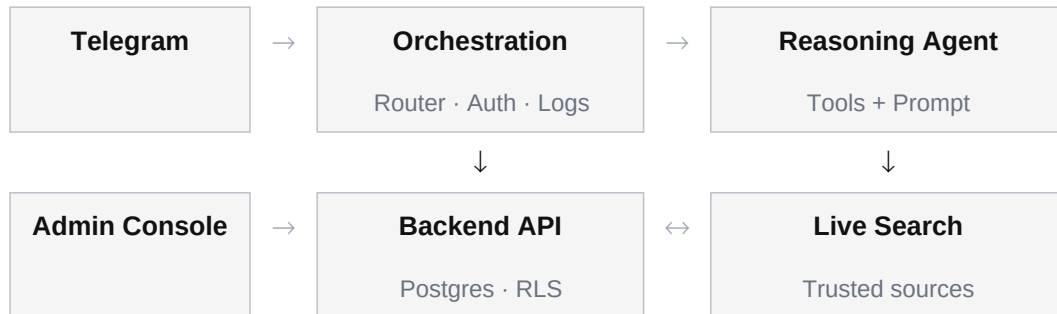
What users get

- **Ticker Q&A** — prices, moves, volume, and context, cited from live sources.
- **Watchlist** — personal ticker list per Telegram user.
- **Morning digest** — scheduled 08:00 Cairo brief of watchlist news.
- **Commands** — /status, /watchlist, /digest, /logout.
- **Activation keys** — issued from the admin console; revocable at any time.

ARCHITECTURE

How the pieces fit

A single Telegram entry point routes every message through an orchestration layer that validates the user, calls the reasoning model, and enforces guardrails before replying. State lives in Postgres; nothing runs on the user's device.



Component roles

- **Telegram** — sole user interface. No app to install, no separate login.
- **Orchestration** — validates the activation key, routes commands, logs each turn.
- **Reasoning Agent** — plans tool use, composes the reply, capped iterations for cost control.
- **Live Search** — narrowed to trusted market sources with a same-day time filter.
- **Backend API** — signed endpoints for key check, quota, watchlist, digest opt-in, chat log, and status.
- **Postgres** — activation keys, redemptions, watchlists, chat logs; row-level security throughout.
- **Admin Console** — issue and revoke keys, inspect redemptions, watch usage.

RELIABILITY & SECURITY

What keeps it stable

Every request is authenticated, every write is logged, and the reasoning budget is bounded. When a tool fails, the assistant degrades gracefully instead of stalling the chat.

Guardrails

- **Signed endpoints** — every backend call carries a shared API key; unsigned traffic is rejected.
- **Row-level security** — Postgres policies scope every read to the caller's identity.
- **Bounded reasoning** — the agent runs at most a few iterations per turn to keep latency and cost predictable.
- **Time-boxed search** — market queries are restricted to the current day and to a curated source list.
- **Empty-reply fallback** — when a tool returns nothing, the bot answers with a clear retry message.
- **Revocation** — /logout and admin revoke both invalidate the redemption instantly.
- **Admin key** — a single unlimited operator key, isolated from user-issued keys.

Failure modes handled

- Stale snippets from generic web results → same-day filter plus source list.
- Runaway agent loops → hard iteration cap and tight system prompt.
- Empty model output → deterministic fallback reply.
- Lost activation → user can re-key without contacting support.

ENGINEERING HIGHLIGHTS

The details worth calling out

Activation & identity

Keys are opaque strings issued from the admin console and redeemed on first message. A redemption row binds the key to a Telegram chat id; only the latest active redemption is trusted, so re-keying after `/logout` just works.

Prompt & tools

The system prompt fixes the persona in Egyptian Arabic, forbids repeated searches, and points the agent at the live-search tool with a same-day constraint. Tools are described minimally so the model does not waste tokens paraphrasing them.

Backend surface

Endpoint	Purpose
POST <code>/api/public/validate-key</code>	Redeem a key on first contact
POST <code>/api/public/check-access</code>	Gate every subsequent message
POST <code>/api/public/consume-quota</code>	Track per-day usage
POST <code>/api/public/watchlist</code>	Read / write user's tickers
POST <code>/api/public/digest-optin</code>	Toggle morning digest
GET <code>/api/public/digest-recipients</code>	List users for the 08:00 job
POST <code>/api/public/log-chat</code>	Append a turn to the audit log
POST <code>/api/public/status</code>	Compose the <code>/status</code> card
POST <code>/api/public/revoke</code>	Handle <code>/logout</code> and admin revoke

Morning digest

A scheduled job runs at 08:00 Africa/Cairo, pulls opted-in users and their watchlists, batches a single search per ticker, summarizes, and posts one message per user. Users can toggle it any time from the chat.

OUTCOMES

What we ship, at a glance

1	9	2	0
Chat interface	Backend endpoints	Scheduled jobs	Client installs

Why this shape

- Telegram removes the install and onboarding tax entirely.
- Activation keys let operators run trials, partnerships, and private cohorts without touching auth code.
- A tight backend surface keeps the reasoning layer swappable — model, provider, or prompt can change without touching the chat.
- Row-level security in the database means the admin console and the bot share the same rules.