DIRAYA // PRODUCTION AI ENGINEERING

YOUR AI FEATURE: A REFERENCE ARCHITECTURE REVIEW

The build we start from, the three risks that kill most AI features, and a realistic 8-week timeline.

A note from Mohammed, who runs engineering at Diraya. You asked for a review. Below is the reference architecture we start every build from, the three risks we watch kill most AI features before they ship, and a realistic path to production. This is the general version. For the one tailored to **your** exact stack and data, reply with two lines on what you are building and I send it back inside 48 hours, or grab 15 minutes on the call link below.

1. The reference architecture

1. **Eval harness, built first.** A written test suite that defines good enough to ship. Everything else is measured against it. This single inversion, evals before agent, is what separates a demo from a product.
2. **Data and retrieval layer.** Clean ingestion, sane chunking, and a retrieval step you can score on its own. Most failures start here, not in the model.
3. **Model and agent layer.** The smallest model and the fewest steps that pass the evals. You scale up only when the evals demand it, never the other way around.
4. **Guardrails.** Input validation, output schema enforcement, injection defense on all retrieved content, and an explicit refusal path for when the agent should not answer.
5. **Observability.** Per-step tracing with a trace id from day one, so any bad answer a user reports is reproducible instead of a mystery.
6. **Handover.** The repo, the eval suite, and the docs in your team's hands. You own it and keep shipping on top of it. No lock-in.

2. The three risks that kill most builds

- **A** **No eval suite.** Without it you cannot tell a regression from noise, and the gap between a demo and production stays invisible until a customer finds it for you.
- **B** **Retrieval and data quality.** The model is rarely the problem. Stale indexes, bad chunk boundaries, and empty-retrieval hallucinations are. Fix the data path before you touch the prompt.
- **C** **No observability or guardrails.** Something will break in production. When it does you need a trace to fix it fast and a guardrail to contain the blast radius. The full 40 failure modes are in our Agent Ghost-Cases List.

3. A realistic timeline to production

Week 1	Write the eval suite together. Agree what good enough to ship means, in numbers.
Week 2	First working milestone ships against the evals.
Weeks 3 to 6	Build out the agent and harden it against the evals until the failure rate is low enough.
Weeks 7 to 8	Production hardening, observability, guardrails, and full handover.
Day 56	You own the repo, the evals, and the docs.

The price is fixed before we start. The first milestone lands on **day 14 or you owe nothing** and walk with the code we have built so far.

This is the reference. The version tailored to your stack, your data, and your timeline is a 15-minute call: **calendly.com/amoura-ma-diraya/30min**. Or reply with two lines on what you are building and I send the tailored review back inside 48 hours.