Case Study:

FAQ Surge Assistant for E-commerce Sellers (11.11 Scenario)

Prepared by **Rex Chang**

GenAI & Cloud Solution Architect / Program Lead





Results and Business Relevance

Problem

Shopping-festival support faces repeated questions in several languages.

Measured Impact

Repeated answers were 100×+ faster than generating fresh answers in the best-performing lab setup, after warm-up (20 requests per setup).

Repeat handling

Saved answers reduce repeated generation work.

Reviewability

Source links and route details support answer review.

Faster repeat answers with an inspectable answer path.



Why This System Exists

The Pain

1. Customers keep asking the same questions in different languages.
2. Generating a fresh answer each time slows replies and creates a queue.
3. Support staff still have to step in when replies are too slow or hard to trust.

Why Naive Fixes Break

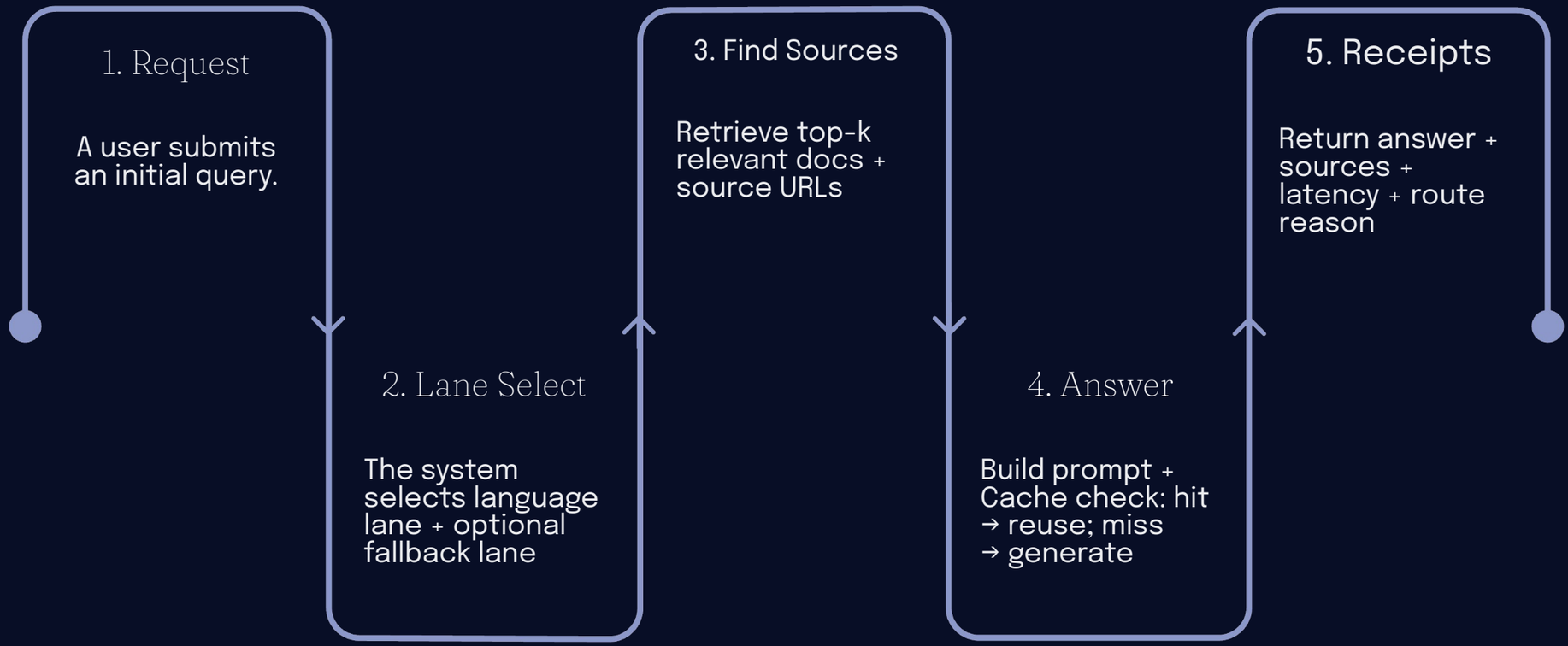
- **Live RAG every request:**

LLM calls still fire; cost scales linearly – repeats never get cheaper

- **Answers without sources:**

Disputes can't be resolved; manual review load spikes at the worst moment

End-to-End Decision Flow



Cost control
Reuse prior answer on repeats (via cache layer)

Risk control
Fallback lane + receipts required

Measurability
p95 / QPS / cache hits recorded

Scalability
Separate lanes per language

Deliberate Choices at Decision Points

Each row maps to a control point from the flow diagram: *Lane Select → Find Sources → Answer → Receipts*.

Flow Node	Deliberate Choice	Alternative	Potential business relevance	Trade-off
Lane Select	Separate lane per language	One shared lane	Language-specific retrieval	More collections / config overhead
Find Sources	Always return source with retrieved docs	Retrieve text only	Source links support review	Slightly more payload and UI surface
Answer	Reuse prior answer on cache hit	Always regenerate	Less repeat generation	Must prevent wrong reuse on similar-but-different queries
Receipts	Show route reason + index badge	No routing visibility	Route visibility supports diagnosis	Adds UI surface area

Warmed-Repeat Benchmark

Sequential Warmed-Repeat Benchmark

What we tested

- Sequential EN/ZH/TH test: 20 requests per setup, after warm-up.

Benchmark acceptance criteria

- **Speed bar**: tail latency improves ≥25% OR throughput improves ≥2×
- **Trust bar:** every answer returns sources + route reason

Results

Measured repeat performance:

- 100×+ faster median replies in the best-performing setup. Throughput improved 62–86× on the primary path and 3.7–4.2× on the backup path.

Answer review:

- Source links and route details support inspection.

Primary-Path Performance

Mode	P95 latency	QPS
Fresh	2.46–5.43 s	0.43–0.56
Saved answer	12–14 ms	34–43

- Saved answers reduce repeat generation.
- Source links support answer review.

Safeguards for Controlled Rollout

Ops & Governance

Enforcement Rules

If performance bar not met → fail;
if receipts missing → fail

Audit Trail

Every request logs route decision, sources, and latency for post-incident review.

Failure Containment

Fallback lane is explicit;
long inputs trigger visible routing notes.

Deterministic Evidence

Logs + receipts make incidents reproducible and debuggable.

Performance thresholds, fallback paths and receipts support evaluation and diagnosis.

Appendix

Technical Deep Dive



Deep Mechanism

Invariants, Freeze Semantics & Replay Harness

Core Invariants

Routing invariant:

- every request resolves to exactly one lane (language × embedder); the lane is always visible via index badge. Fallback route available.

Evidence invariant:

- every response must return sources or an explicit policy reason.

Cache invariant:

- reuse allowed only when cache identity matches (language + lane + normalized prompt + versions).

Decision Protection (Freeze + Override)

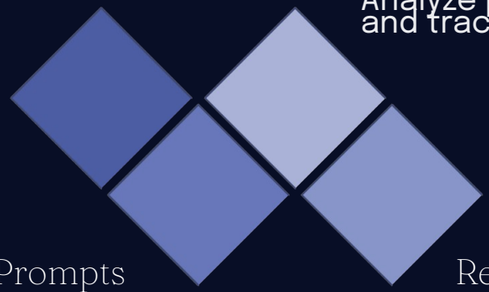
1. If repeat query hits cache → reuse answer + sources (frozen for that identity).
2. Future runs cannot overwrite unless lane or versions change.
3. Override allowed only via explicit backup override or policy-triggered reroute.

Replay Runner

Execute and test system behavior using recorded scenarios.

Metrics & Traces

Analyze performance data and track execution paths.



Hot-set Prompts

Define and manage dynamic prompts for system interactions.

Release Gate

Establish quality checks before deploying new updates.

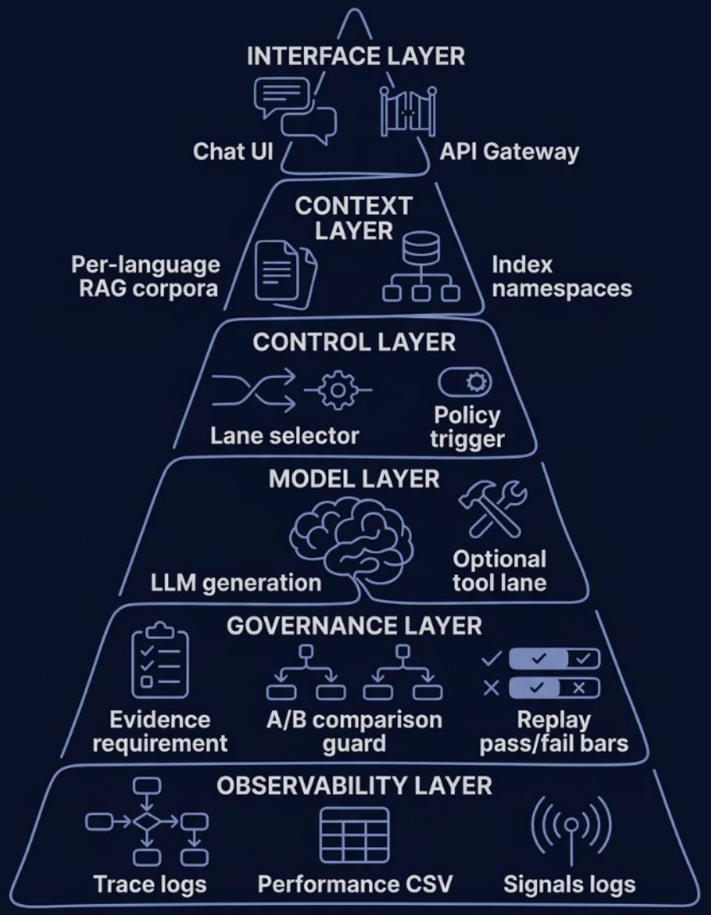
Replay & Evaluation Harness

- **Replay**: committed hot-set prompts reproduce identical traces and metrics.
- **A/B**: run Primary vs Backup on the same query and compare sources + latency.
- **Gate**: ship only if p95/QPS performance bar and evidence bar both pass.

Extended Architecture: Layered System + Data & Policy Flow

This appendix card provides a detailed technical overview of the system's architecture, demonstrating its robustness and thoughtful design.

Layer	What it is in this system	Business purpose
1. Inputs & Context	Query + language + allowed FAQ context	prevents “random context” drift
2. Control Plane	Lane selection + fallback trigger	reduces wrong-lane risk
3. Generation	Produce structured JSON answer	consistent output for ops
4. State & Learning	Reuse cache for repeats	reduces repeat generation
5. Measurement & Evidence	p95/QPS/hit-rate + A/B diff logs	records test comparisons
6. Ops & Governance	receipts required + policy reason surfaced	supports release decisions



Data Flow

- User Query → Retriever (lane) → Router/Policy → LLM → Policy validation → Response + Trace

Thank You!

Rex Chang

GenAI & Cloud Solution Architect / Program Lead

Portfolio: rexchang.dev | LinkedIn: linkedin.com/in/rex-chang-tw | Email: rex@rexchang.dev