great. here's the **10-PR roadmap**—tight, vertical, and enforceable. no prompts yet; this is scope + objectives + merge gates only.

---

# global constraints (apply to every pr)

- **diff hygiene:** ≤600 added loc, ≤12 files touched, no todos, no dead stubs.
- **tooling:** ruff + black + mypy --strict + pytest green on ci.
- **contracts only across boundaries:** pydantic models; no untyped blobs.
- **determinism:** seed captured per run; adapters/selector respect seed.
- **metrics by default:** every new component emits latency/retries/cache_hit/tokens/cost where relevant.
- **perf gates in ci:** ttfe < **800 ms**; e2e p50 ≤ **6 s** / p95 ≤ **10 s** on fixtures (start enforcing by pr9).
- **security:** every read/write scoped by org_id; 429s include retry-after; breaker returns **503** + retry-after.

---

## PR1 — scaffolding, contracts, settings, eval skeleton

**purpose:** pin interfaces day 1 and attach tests to them.

**scope:** repo layout, pydantic-settings config, .env.example, pre-commit, base ci; contracts: IntentV1, PlanV1, Choice.V1 + ChoiceFeatures, Attraction.V1 (tri-state, opening_hours map), WeatherDay, FlightOption, Lodging, Money/When/Window/Geo/Provenance; eval/runner.py + 2 dummy scenarios.

**"good" means:** imports are cycle-free; mypy strict passes; eval runner executes and asserts two trivial must_satisfy.

**merge gates:** added loc ≤400; ci green; contracts ≤40 lines/type; constants (buffers, fx policy) defined once.

---

## PR2 — db + alembic + tenancy + idempotency + rate limits

**purpose:** persistence + safety rails before behavior.

**scope:** sqlalchemy models + migrations: org, user, refresh_token, destination, knowledge_item, embedding, agent_run, itinerary, idempotency; redis token bucket for per-user quotas (agent 5/min, crud 60/min).

**“good” means:** migrations up/down clean; composite unique keys include org_id; 429 behavior with retry-after is deterministic.

**merge gates:** tests: cross-org read returns 0; rate-limit unit tests; seed fixtures script.

---

## PR3 — tool executor + cancellation + /healthz + metrics stubs

**purpose:** deterministic edge: timeouts, retries, breaker, cache; cooperative cancel.

**scope:** executor: **2s soft / 4s hard**, 1 retry (200–500 ms jitter), breaker 5/60s → **503 + retry-after**; dedup key sha256(sorted_json(input)); per-tool ttls; cancel token plumbed; /healthz (db + outbound headcheck); metrics registry.

**“good” means:** breaker opens properly; cancel flips runs to cancelled and stops scheduled work; metrics counters/histograms wired.

**merge gates:** unit tests for breaker header, retry jitter bounds, cancel propagation.

---

## PR4 — orchestrator skeleton + sse + minimal ui vertical

**purpose:** end-to-end vertical early (fake nodes).

**scope:** langgraph nodes (intent→planner→selector→tool_exec→verifier→repair→synth→responder) with checkpoints; sse endpoint (bearer auth, heartbeat 1s, throttle ≤10/s, resume by last_ts); streamlit page that subscribes and renders events.

**“good” means:** ttfe < 800 ms with fake nodes; heartbeat seen; reconnect replays.

**merge gates:** tests: sse requires bearer; subscription to other org’s run_id = 403.

---

## PR5 — adapters (weather real + fixtures) + canonical feature mapper + provenance

**purpose:** typed sources + one place for features.

**scope:** adapters: weather (real, 24h cache), flights/lodging/events/transit/fx (fixtures); feature_mapper.py turns tool objects → ChoiceFeatures; provenance includes ref_id|source_url.

**“good” means:** all adapter returns carry provenance; feature mapper is pure/deterministic; no selector touching raw tool fields.

**merge gates:** tests: missing provenance fails; cache hit toggles metric; forced timeouts trip breaker.

---

## PR6 — planner + selector (feature-based) + bounded fan-out

**purpose:** real branching and ranking.

**scope:** planner builds limited branches; selector uses ChoiceFeatures only; fan-out cap ≤4; freeze z-means/std from fixtures; log score vector for chosen + top 2 discarded.

**“good” means:** happy-path scenario runs e2e with real adapters/fixtures; score logs appear; branches obey cap.

**merge gates:** eval: happy path passes; unit: selector never references nonexistent fields.

---

## PR7 — verifiers: budget, feasibility (hours/buffers/tz/dst/last train), weather (tri-state), prefs

**purpose:** correctness wall, pure functions.

**scope:** budget (selected only via deref; fx T-1; +10% slippage), feasibility (any window covers slot; airport 120m, in-city 15m, museums 20m; tz-aware; dst jump tests; last train cutoff), weather (blocking/advisory), prefs.

**“good” means:** 4 negative scenarios flip to violations pre-repair; properties guard time math.

**merge gates:** tests: split-hours (13:00 fail, 15:00 pass), rainy unknown advisory vs outdoor blocking, overnight flight, dst forward/back; metrics: budget_delta_usd_cents.

---

## PR8 — repair loop + partial recompute + decision diffs

**purpose:** bounded, explainable fixes.

**scope:** moves: airport → hotel tier → reorder → replace; ≤2 moves/cycle; ≤3 cycles; partial recompute reuse; diff {usd_delta_cents, minutes_delta, reason, provenance}; stream decisions.

**“good” means:** first-repair success ≥70%; median repairs/success ≤1.0; reuse ≥60%.

**merge gates:** eval cases enriched to include repair success assertions; metrics emitted for reuse + decisions.

---

## PR9 — synthesizer + “no evidence, no claim” + ui right-rail + perf gates

**purpose:** render trusted output; wire perf/citation gates.

**scope:** synthesizer from structured state only; citations per field from provenance; ui right-rail shows tools, timings, checks, decisions, citations; ci perf tests (ttfe / p50 / p95 on fixtures).

**“good” means:** provenance_coverage ≥ .95 on golden; no hallucinated fields when data missing; ci enforces perf slo.

**merge gates:** tests: coverage check; ci job fails if perf exceeds thresholds.

---

## PR10 — auth hardening + sse tenancy test + chaos toggles + full eval + readme demo + ablations

**purpose:** production basics + proof.

**scope:** jwt rs256 (access 15m / refresh 7d) with rotation; argon2id; lockout after 5 fails/5-min backoff; sse tenancy test; chaos env flags (FORCE_TOOL_TIMEOUT, EMPTY_RAG, DROP_SSE); eval suite 10–12 scenarios (budget pinch, closed venue, split hours, rainy unknown vs outdoor, overnight, dst, last train, fx outage, partial-day arrival/departure, check-in/out windows, locked slot); README with demo script + ablation results (no cache/parallel/repair).

**“good” means:** rotation works, lockout works, sse cross-org is blocked; chaos degrades gracefully (banner, omissions, no crashes); eval pass-rate ≥90%.

**merge gates:** tests for rotation/lockout/tenancy; chaos demos recorded; ablation numbers present and sane.

---

# parallelization + dependencies

- pr1→pr4 strictly serial.
- pr5 can start once pr4 exists; pr6 after pr5.
- pr7 can start once contracts are set (pr1) and planner skeleton exists (pr4), but merges after pr6.
- pr8 after pr7.
- pr9 after pr5–8.
- pr10 last.

---

# blind-spot sentry (reviewers must check each pr)

- any untyped json crossing a boundary? **reject**.

- any boolean that could be unknown but isn't tri-state? **fix**.
- any reference to schema fields that don't exist? **fix**.
- breaker returning cached error body instead of **503 + retry-after**? **fix**.
- sse without bearer or without org scoping? **fix**.
- provenance missing on any rendered claim? **fix**.
- branches > cap or selector reading raw adapter fields? **fix**.
- ci perf gates missing (from pr9 onward)? **fix**.

---

# week-level outcome (what this roadmap guarantees if followed)

- **day 1–2:** contracts stable, eval/ci in place, vertical sse path visible.
- **day 3–4:** real tools + feature-based selector + verifiers passing core negatives.
- **day 5:** bounded repair + trusted synthesis + perf/citation gates + auth/ops basics + eval ≥90% + chaos degradation demos.

if you want, next step i'll turn **pr1** into a crisp issue with acceptance checklist and a *short* context brief for claude—but per your instruction, no prompts yet.