Master Plan v1

Okay so the plan , here's a clean, end-to-end roadmap with tight exit criteria and quick "smoke tests" per phase. start at 0, stop whenever the phase's smoke test passes.

Phase 0 — North-Star & Guardrails

Goal: Define success, scope, and safety.

Deliverables: success metrics, target providers, repo policy (branch, PR, reviews), secrets layout, budget caps, data policy.

Smoke test: a 1-page ADR (/docs/ADR-0001-orchestration-goals.md) plus a CLI dry-run that prints the planned loop without calling any API: npm run plan:dry.

Phase 0.5 — Repo Bootstrap

Goal: Minimal runnable skeleton.

Deliverables: TS monorepo, apps/orchestrator with kernel.ts, providers/*, integrations/github.ts, memory.ts, budget.ts, unit test harness, ESLint/Prettier, CI.

Smoke test: npm test passes; npm run dev starts an event loop using mock agents and exits cleanly.

Phase 1 — Two-Agent Loop (All Mocks)

Goal: Prove message protocol + debate loop locally.

Deliverables: JSON message schema (propose/critique/implement/verify), in-memory bus, loop exit conditions.

Smoke test: npm run poc:mock "add a README section" prints a mock diff and "CONVERGED" within 3 turns.

Phase 2 — Real Provider Adapters (Claude + OpenAI/GPT)

Goal: Swap mocks for real API calls.

Deliverables: providers/anthropic.ts, providers/openai.ts, env keys, retry/backoff.

Smoke test: npm run ping:providers returns both model IDs + token counts; budget tracker records spend.

Phase 3 — Git Integration & PR Automation

Goal: From diff → branch → commit → PR.

Deliverables: Octokit adapter, patch applier, PR body with provenance footer (prompts, hashes).

Smoke test: npm run poc:pr "add LICENSE" opens a PR on a sandbox repo; PR body contains "orchestrator-provenance".

Phase 4 — Budget & Context Manager

Goal: Keep costs predictable.

Deliverables: working vs project memory, token ceilings per step/feature, adaptive truncation, caching.

Smoke test: run the same task 3×; total spend variance ≤10%; run aborts gracefully when MAX_USD reached.

Phase 5 — Review Gates (CodeRabbit + Copilot)

Goal: Automatic review on PRs.

Deliverables: CI workflow to request both reviewers; status checks required to merge.

Smoke test: opening a PR triggers 2 bot comments; intentionally add a simple vuln and see at least one bot flag it.

Phase 6 — Minimal RAG on the Repo

Goal: Make agents repo-aware, cheaply.

Deliverables: file indexer (chunking + embeddings), semantic retrieve-then-prompt.

Smoke test: prompt "update src/auth.ts per policy"; builder cites the correct chunk id/lines in its rationale.

Phase 7 — Debate Protocol Hardening

Goal: Traceable reasoning.

Deliverables: claims[], evidence[], risks[] objects; critique rubric; convergence rules.

Smoke test: inject a bug; Architect must emit risk: high with a concrete fix; loop converges after fix.

Phase 8 — Test-Writer Role (3rd Agent or Builder-Mode)

Goal: Generate & run acceptance tests before code.

Deliverables: test template generator, local test runner hook.

Smoke test: orchestrator creates a failing test, code changes, tests pass, PR includes both.

Phase 9 — Policy/Risk Engine (Onyx)

Goal: Prevent unsafe/illicit output.

Deliverables: secret/PII scanners, license compatibility check, policy DSL, hard stops.

Smoke test: attempt to commit an .env secret or GPL into MIT repo → build fails with actionable error.

Phase 10 — Observability & Telemetry

Goal: See every step & dollar.

Deliverables: event store (SQLite/PG), spans for each turn, spend per feature, error catalog.

Smoke test: dashboard shows a full run timeline, token totals, and top 3 costly prompts.

Phase 11 — Operator UI (Thin)

Goal: Supervise & steer.

Deliverables: small React app: runs list, live log tail, "pause / edit prompt / resume," "rerun step."

Smoke test: pause mid-run, edit an instruction, resume; run completes and artifacts persist.
Phase 12 — Parallelism & Sharding
Goal: Speed via concurrency.
Deliverables: subtask graph execution (tests/docs/refactor in parallel), concurrency limiter, budget partitioning.
Smoke test: two subtasks run in parallel without exceeding budget or tripping rate limits; wall-time < sequential by ≥30%.
Phase 13 — Human-in-the-Loop Governance
Goal: Safe autonomy with approvals.
Deliverables: approval gates (high-risk changes), "escalate" path, rationale bundle for reviewers.
Smoke test: risky migration triggers an approval request with diffs, claims, risks, and rollback plan attached.
Phase 14 — Multi-Agent Expansion (Full Cast)
Goal: Add Security Architect, Doc Writer, Refactorer, Performance-Tuner.
Deliverables: role policies, conflict resolution (vote / tie-breaker), turn-taking strategies.
Smoke test: role conflict intentionally seeded; tie-break resolves; final PR includes code + docs + perf note.
Phase 15 — Reliability & SLOs
Goal: Production hardening.
Deliverables: retries, circuit breakers, idempotent tasks, dead-letter queue, snapshot/restore.
SLOs: p95 feature cycle time, error rate, cost per merged PR.
Smoke test: kill a provider mid-run; fallback path completes; SLO dashboard stays green.
Phase 16 — Extensibility & Ecosystem
Goal: Easy to add tools/providers.
Deliverables: adapter interface, conformance tests, plugin registry, sample "new provider in <200 LOC>."
Smoke test: add a dummy provider; all conformance tests pass; a real run can route 1 step through it.
Phase 17 — Domain Integrations (Optional)
Goal: Bring in your world.
Deliverables: Canvas-MCP telemetry hooks, project templates for SAE assignments, Atlas research assist.
Smoke test: from the UI, select a template ("JWT auth feature" or "Audio tooling micro-service"), the orchestrator loads the right scaffolds and completes a PR with docs.

———

Quick "Are we done?" checklist (per phase)

- **Each phase leaves behind: code, one-page doc, and a runnable npm run smoke:p{N}.**
- **Promotion rule: a phase isn't "done" until its smoke script passes on a clean checkout with only .env filled.**

"