

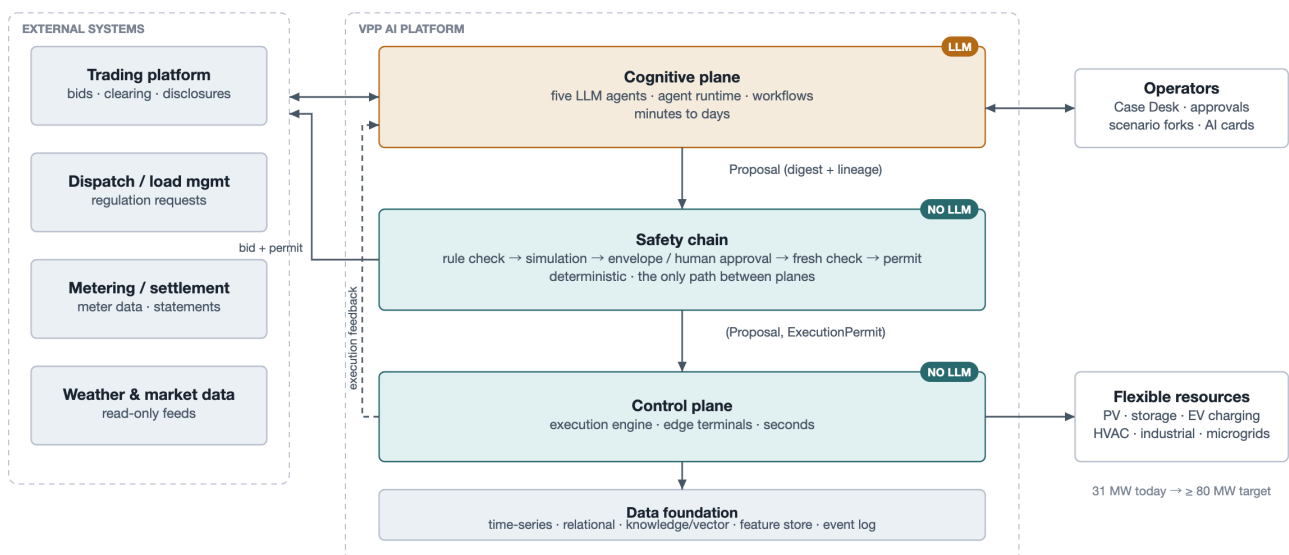
VPP AI Platform — Architecture

Virtual Power Plant Multi-Timescale Intelligent Operations Platform, built on the GuangMing Power LLM

The platform is an AI-assisted decision and controlled-execution system for a virtual power plant in Hubei. Five LLM agents propose market bids, dispatch plans and load-control plans. A deterministic safety chain governs everything before any external effect. Three commitments define the design:

- **Two planes, one channel.** A slow cognitive plane (LLM agents, minutes to days) is separated from a fast control plane (deterministic execution, seconds). The safety chain is the only path between them.
- **The LLM never computes a number.** Prices, volumes and setpoints come only from versioned forecasting, optimisation and simulation tools. Every numeric field in a proposal is a reference to a recorded tool call.
- **Autonomy through envelopes.** Humans approve envelopes (bounded authority with a validity window), not every action. In-envelope actions auto-pass after rule check and simulation. Everything else escalates to a human. Envelopes start empty and widen only on evidence.

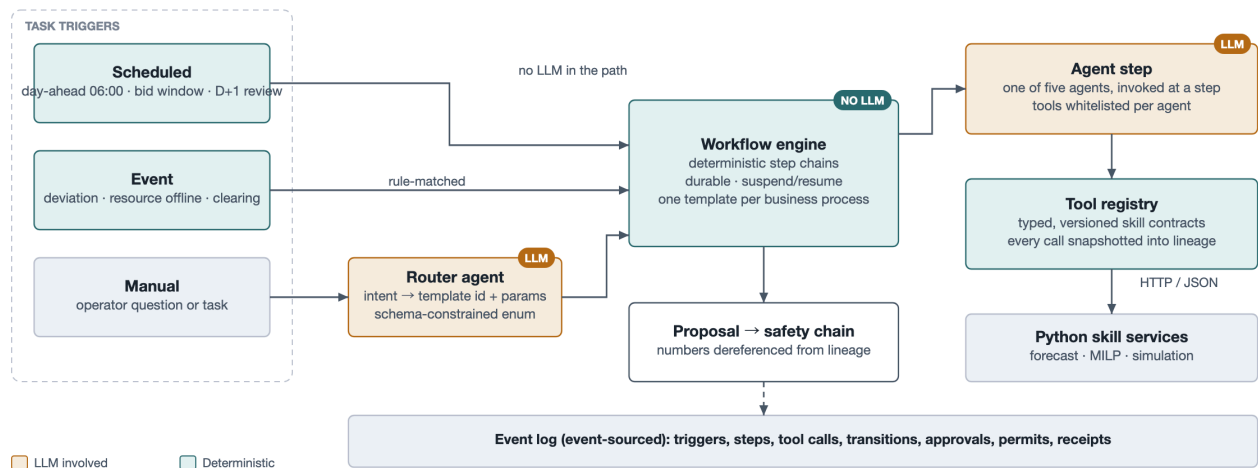
1. System context



The platform sits between the provincial trading, dispatch and metering systems above and the aggregated flexible resources below (31 MW today, at least 80 MW targeted). Operators work through a Case Desk, where every AI output is attached to an accountable case with an objective, an owner and a deadline. The platform overlays the existing 3060 platform: it reuses master data and the

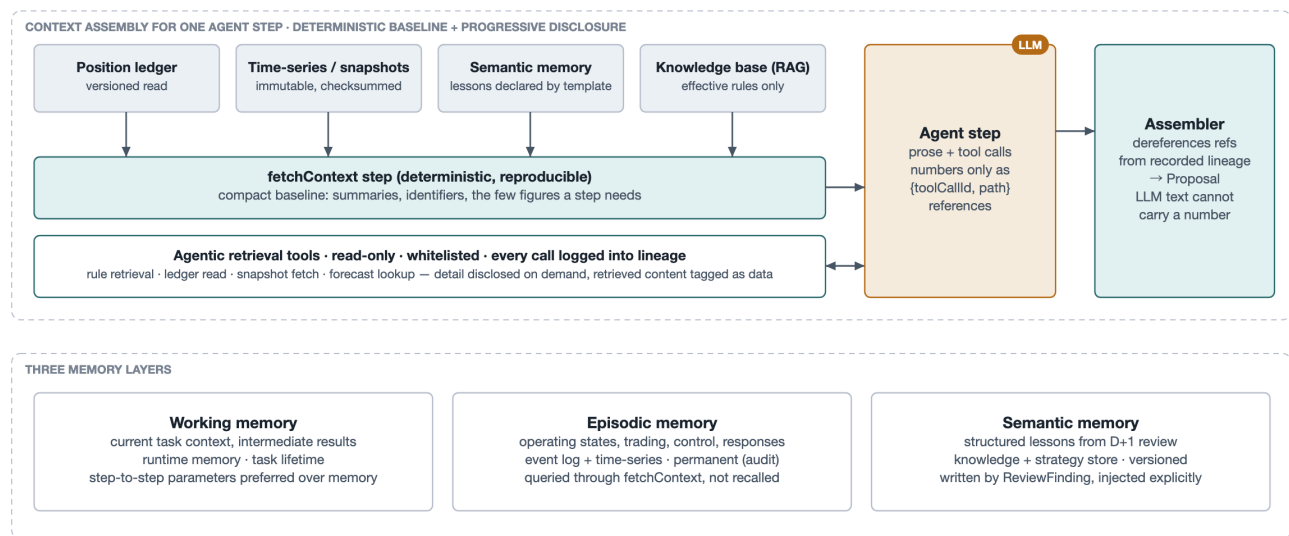
permission system, and it converges every bid and control action into one audited channel so nothing can bypass the safety chain.

2. Agent runtime



Workflows are the entry point, not agents. Every business process (day-ahead bidding, review, intraday correction) is a deterministic, durable chain of steps. Agents are invoked at specific steps with a whitelisted tool set: computation skills that produce numbers, and read-only retrieval tools that fetch detail on demand. Scheduled and event triggers instantiate workflow templates without touching the LLM. Only manual requests pass through the router agent, whose output is constrained by schema to a registered template ID plus parameters. The LLM decides *what* to do; the engine decides *how*. Every trigger, step, tool call, transition, approval and permit is written to an event-sourced log.

3. Context management and memory

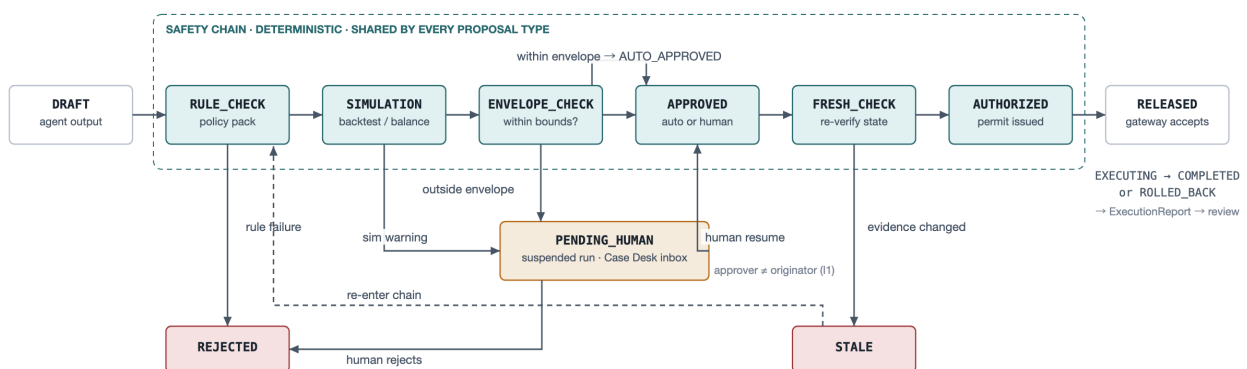


Context for each agent step is assembled by a deterministic upstream step that reads the versioned ledger, immutable snapshots, explicitly declared semantic memory and effective knowledge entries. The agent does not choose its baseline context, so every decision starts from a reproducible footing. Agent output carries numbers only as references into recorded tool-call lineage. The assembler dereferences them, which means LLM text cannot carry a number into a proposal.

Context is built by **progressive disclosure**. The baseline is compact: summaries, identifiers and the few figures a step needs, sized for a conservative context window. Detail is disclosed on demand through **agentic retrieval tools**: read-only, whitelisted registry tools such as rule retrieval, ledger reads, snapshot fetches and forecast lookups that an agent may call when it needs more. Each retrieval is recorded in lineage with its inputs and outputs, so what the agent saw is replayable, and prompt-injection defences treat retrieved content as data, never as instructions. The same principle shapes the operator side: an insight card shows the conclusion first and expands, on request, into the tool outputs and data sources behind it.

Three memory layers serve different lifetimes. Working memory lives for one task. Episodic memory is the event log and time-series store, permanent and queried rather than recalled. Semantic memory holds structured lessons written back by the D+1 review workflow, versioned and injected explicitly by each template. Authoritative operational facts always live in the domain database, never in agent memory.

4. Safety chain and state machine



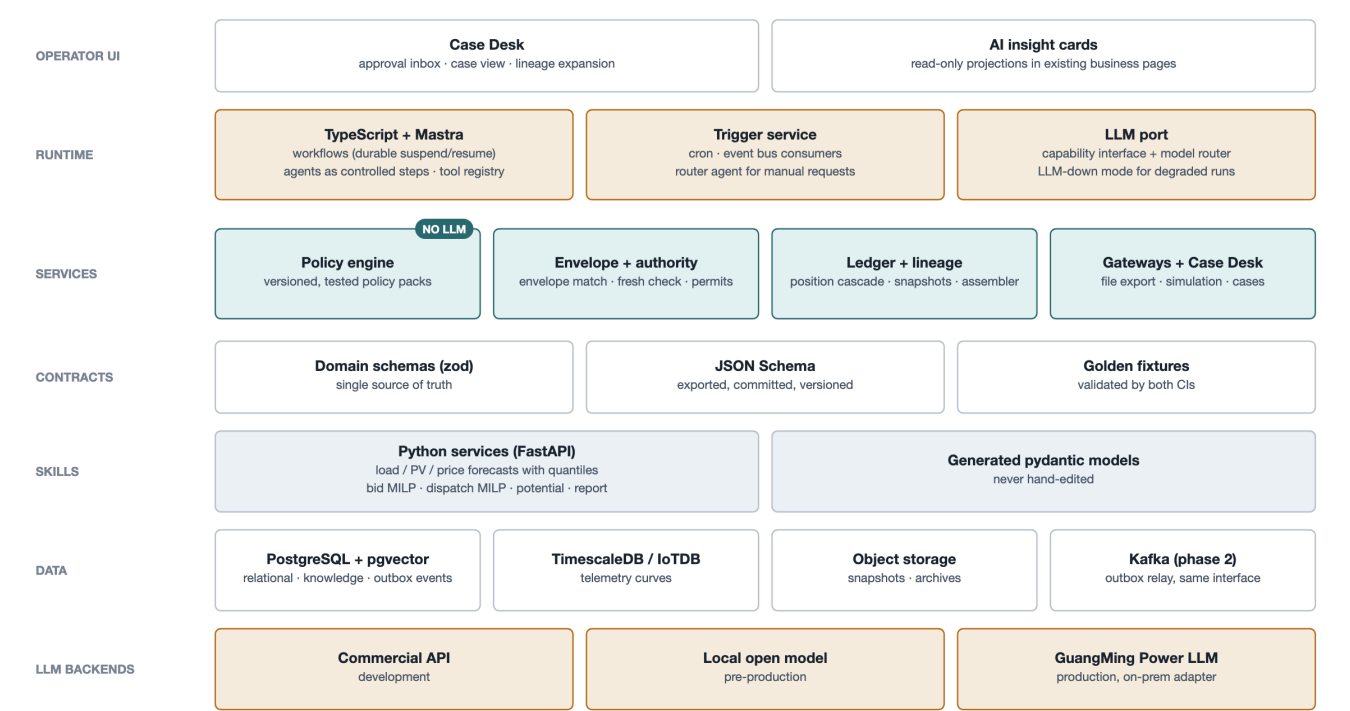
Approval binds to the proposal digest and a validity window (l3). Gateways accept only (Proposal, ExecutionPermit): short-lived, revocable, digest-matched (l4). Simulation warnings always escalate to a human. No code path lets the AI reach APPROVED. Envelopes start empty and widen only on shadow-run evidence.

Every intended action is a `Proposal` with a typed payload, full lineage and an immutable digest. One lifecycle workflow implements the chain for all proposal types:

STAGE	WHAT HAPPENS
Rule check	Deterministic policy pack: compliance, constraints, ledger consistency, lineage completeness, separation of duties
Simulation	Revenue backtest for bids, power balance for control plans. A warning always escalates to a human
Envelope check	Within an approved envelope: auto-approved with a full record. Otherwise the run suspends into the Case Desk inbox
Human approval	A resume by an approver whose identity is separated from the origination chain. Binds to the digest and a validity window
Fresh-state check	Immediately before release: resources online, constraints satisfied, evidence unchanged. Otherwise <code>STALE</code> and re-enter
Permit and release	An <code>ExecutionPermit</code> is issued: short-lived, revocable, digest-matched. Gateways accept only the <code>(Proposal, ExecutionPermit)</code> pair

Suspended approvals survive process restarts. Permit revocation is the fastest rollback: the gateway refuses further commands and edge terminals fall back to the last approved plan or a safe curve.

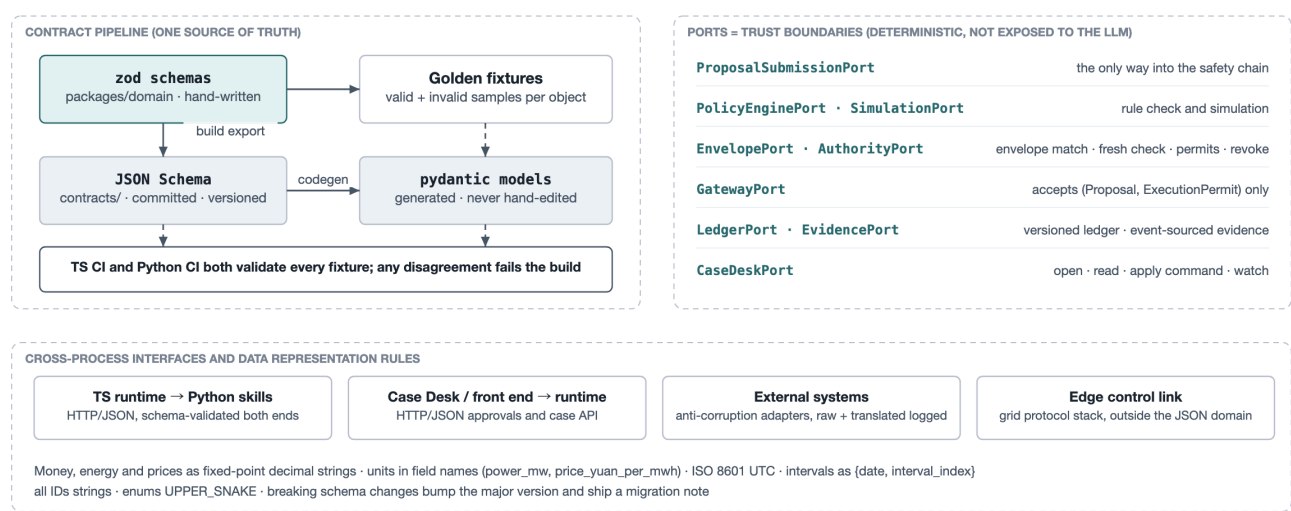
5. Technology stack



Backend swaps are gated by the L1 evaluation set and an L4 shadow comparison, not judgment. About a third of the platform is framework; the differentiating parts are the deterministic services.

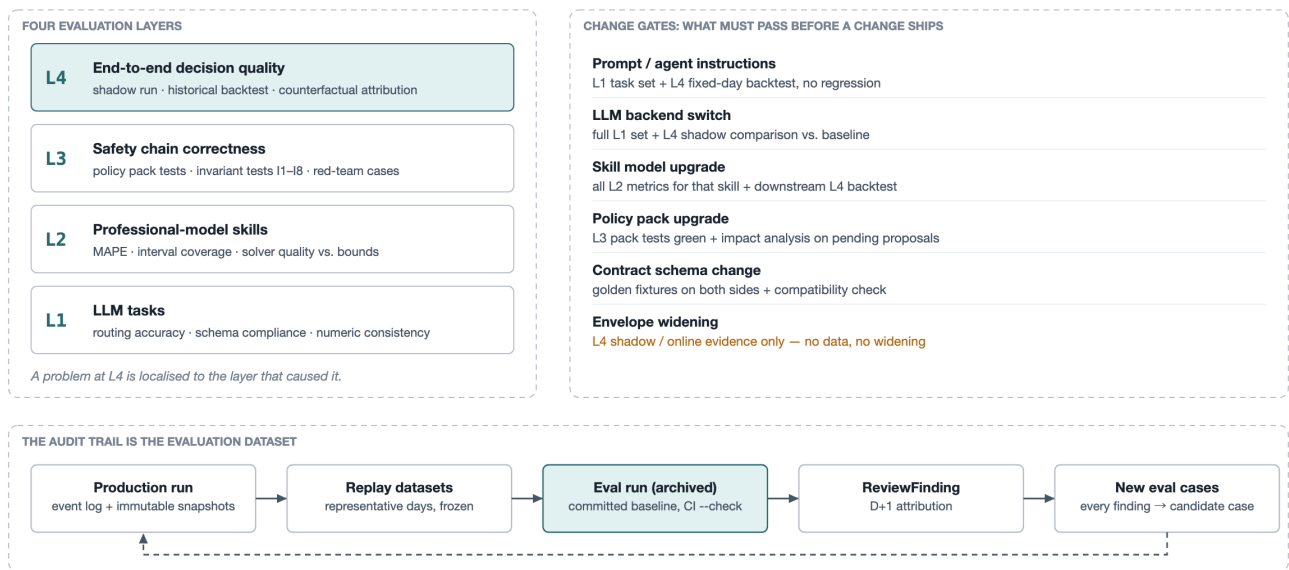
The runtime, agents, workflows, domain schemas and deterministic services are TypeScript on Mastra, chosen for typed tool contracts and durable suspend/resume. The professional-model skills are Python HTTP services. LLM calls go through a capability-oriented port with a model router, so the backend moves from commercial APIs in development to a local open model in pre-production and the GuangMing Power LLM adapter in production with no business-code change. The runtime also has an LLM-down mode: forecasting, optimisation, checking, approval, bidding and control all run without the LLM. Roughly a third of the platform comes from the framework. The differentiating assets are the deterministic services.

6. Interfaces and contracts



Hand-written zod schemas are the single source of truth. The build exports JSON Schema to a committed contracts directory, from which pydantic models are generated and never hand-edited. Golden fixtures, valid and invalid, are validated by both the TypeScript and Python CI, and any disagreement fails the build. Inside the platform, each trust boundary is a port that the LLM cannot call: proposal submission, policy engine, simulation, envelope, authority, gateway, ledger, evidence and Case Desk. Data representation rules are part of the contract: decimal strings for money and energy, units in field names, UTC timestamps, intervals as date plus index, string IDs, upper-case enums.

7. Evaluation-driven development



Because every decision is replayable from retained evidence, the audit trail is also the evaluation dataset. Four layers localise problems: L1 LLM tasks, L2 professional-model skills, L3 safety-chain correctness including one automated test per invariant, and L4 end-to-end decision quality through shadow runs and historical backtests. Evaluation is a change gate. Prompt changes require the L1 set and an L4 backtest. A backend switch requires the full L1 set and an L4 shadow comparison. Skill upgrades must update their committed L2 baseline in the same change. Policy-pack upgrades require all rule tests green. Envelope widening is approved only on L4 shadow or online evidence. Every review finding becomes a candidate evaluation case, so failures feed the next baseline.