

upswing • SECOND MULTI-AGENT SYSTEM

AUTONOMOUS SOLUTION ARCHITECT

Solution Factory

INTAKE • DESIGN • SIEGE • CONSENSUS • SPEC

One plain-English sentence in, a build-ready specification out. A LangGraph pipeline that interviews you, designs a grounded solution, then turns a hostile panel loose on its own design until no blocking objection survives.

BUILT & VERIFIED ON REAL RUNS / TWO MANDATORY HUMAN GATES

 SOLUTION-FACTORY-LG



LANGGRAPH • LANGCHAIN • POSTGRES

THE PREMISE

A design is guilty until a hostile panel fails to convict it.

Turning a vague "we should automate X" into something a team can build takes a senior architect days of interrogating the request, researching prior art, drafting, then surviving review.

Most AI architect tools collapse that into one model call that grades its own homework: confident, ungrounded, unchallenged. Solution Factory refuses the shortcut. It separates the agent that designs from the panel that attacks, and treats consensus as a measurable state, not a vibe.

01 · ADVERSARIAL SEPARATION

The architect is structurally forbidden from grading its own work. Critics see only the design and their lens, never the architect's reasoning.

02 · GROUND CLAIMS, FLAG GAPS

Every design claim is cited to a curated corpus, and what it cannot ground it openly tags as a gap. Provenance is recorded and audited.

03 · CONSENSUS IS A NUMBER

Consensus has exactly one definition: zero remaining open **BLOCKER** findings. A deterministic count, never "the panel feels good."

One sentence becomes a build spec.

Everything between the two human gates runs autonomously. The gates are cheap-first: the first sits before any expensive design, the last before anything is called "build this."



Design principle: durable orchestration as the spine, an adversarial panel as the conscience, the right model on the right call, and a human at each of the two highest-stakes moments.

It interviews you before it spends a dollar.



INTAKE

A discussion panel reads your one sentence and asks the questions a good consultant would ask before quoting work.

01 • THE PANEL FORMS

The system reads your sentence and chooses which experts belong in the room: a security reviewer, a performance reviewer, a systems architect, picked to fit the problem. It opens a ledger row to track the run.

02 • SOCRATIC Q&A

It puts roughly a dozen questions to you, what volume, what defines the right outcome, how fast must it run, then pauses. You answer in free text. From your answers it writes **PROPOSAL.md**, a sharpened statement of the actual problem.

■ GATE 1 YOUR FIRST APPROVAL

Approve it, reject it (the run ends), or send it back for another round. Nothing expensive has happened yet. This gate is cheap insurance that the machine understood you before it spends real design effort.

A single architect, grounded in a curated corpus.

01 • QUERY THE CORPUS

One architect agent queries an Obsidian vault of roughly 2,950 curated atoms across 18 domains, pulling the patterns, failure modes, and reference designs relevant to your problem. The corpus is a resource it cites, never a cage it obeys.

02 • DRAFT THE SCHEMATIC

It writes an end-to-end design: components, data flow, infrastructure, cost notes, alternatives. Every claim it can ground it cites as `[[kind:slug]]`; where the corpus is silent it says so and tags a web fallback.

03 • IT NEVER GRADES ITS OWN WORK

Citations that resolve to real atoms are recorded as provenance. The architect only reads the corpus; it never sees critic state, and critics never see its tool transcript.

14/14

citations resolvable
zero hallucinated



• GROUNDED BY_DOMAIN

• GAP-FLAGGED

• NO STORED VECTORS

A hostile panel attacks, in parallel and in isolation.

Up to thirteen critic personas fan out, each reviewing the schematic seeing only the design and its own lens. A security persona flags an unauthenticated webhook as a **BLOCKER**; a performance persona warns the classifier will miss its latency. Their findings are merged.

THE REVISION-ROUTER ARITHMETIC

After every round, a router decides with pure Python. This is the whole governance model, in five lines: deterministic, auditable, and the reason the system cannot silently spin or fake a win.

```
# after each critique round
if len(this_round) == 0:      # panel said nothing escalate
if len(open_blockers) == 0:  # design survived consensus
if iteration ≥ 3:           # cap backstop escalate
if open_blockers ≥ prev_count: # not shrinking escalate
else:                       # progress, revise revise
```

✓ CONSENSUS

Zero open blockers. The design survived. Move on.

⌚ REVISE · CAP 3

Blockers shrinking, under the cap. The reviser takes every open finding and rewrites; the panel re-attacks the new draft.

✓ ESCALATE TO A HUMAN

Stall, cap, or an empty panel. A panel returning no findings is an alarm, not an all-clear. The machine refuses to spin.

Three real runs. Every terminal path safe.

The same brief converged organically, hit the iteration cap, and stalled across three runs. The value is not a fixed answer. It is that every ending lands at a human gate with an honest, reconciled finding count, never a false green.

RUN A · ORGANIC

9 → 2 → 0

blocker curve

Reached consensus on its own: zero residual blockers.
Reconcile 41 == 41.

• ORGANIC CONSENSUS

RUN B · BASELINE

11 → 2 → 3

blocker curve

Hit the iteration cap, escalated, human accepted
residual risk. Reconcile 55 == 55.

• CAP → ACCEPT RISK

RUN C · INSTRUMENTED

5 → 5

stalled, did not converge

The panel found five real, specific design contradictions the
reviser could not clear in one pass. It detected the lack of
progress and escalated, handing a human an honest list, not a
checkmark. Reconcile 34 == 34.

• STALL → ACCEPT RISK

CONSENSUS → BUILD SPEC



Consensus writes CONSENSUS.md, then a phased .planning/ roadmap with per-phase verification
criteria, the thing an engineer can actually build from.



The build spec is shown, the human approves, and v1 is
done. The deliverable is the approved spec.

A durable LangGraph state machine.

14

nodes in a single StateGraph over one SpikeState.

4

interrupt points: intake, Gate 1, escalation, Gate 2.

Send fan-out · interrupt() gates

Each critic runs as its own parallel branch via Send; a deferred panel_reduce barriers on all of them. Human gates are durable interrupt(): the run pauses in Postgres and resumes from any process, days later.

Keyed-merge reducers

Findings merge on a dedup key so a finding can flip open to resolved in place, instead of appending a duplicate that strands the blocker count above zero. A crashed critic overwrites its own row.

THREE STORES, NEVER CONFLATED

Artifacts

• SOURCE OF TRUTH

Markdown deliverables on disk, written first, content-hashed.

Checkpoint

• DURABLE

Opaque LangGraph run state in Postgres. Enables crash-safe resume and the interrupts.

Ledger

• PROJECTION

A rebuildable projection of each phase boundary. Reconciliation asserts ledger blockers == consensus.

Spend dollars only where they change the outcome.

🔗 CRITIC PANEL → claude -p • Max, flat-rate

The high-cardinality calls, up to 13 critics across 3 rounds, run headless on the flat-rate subscription. The dominant token cost is effectively zero marginal dollars.

⚙️ ARCHITECT + REVISER → native API • metered

The few high-stakes calls run on the metered key, unlocking streaming, generation params, structured output, and adaptive thinking, where model-layer leverage compounds.

BILLING ISOLATION, A LOAD-BEARING INVARIANT

The metered key is passed explicitly to the client and never written to the global env. That is what lets both transports coexist in one process: the panel stays free on the subscription while only the design calls bill metered.

76%

saved vs all-metered
on the baseline run

\$0.52-\$0.87

real metered, three runs
vs \$2.0-\$3.3 notional

HONEST STATUS

Functional and verified. A local dev tool, by design.

4 / 4

stages functional
and verified

3

end-to-end runs,
all reconcile clean

~24m

autonomous wall-clock
per instrumented run

HONEST LIMITS, HELD IN THE OPEN

Novel briefs cap and escalate often, the human gate is load-bearing. Autonomous implementation against a repo is out of scope for v1. It is not RAG: the architect grounds via `by_domain` over a curated corpus, with no stored vectors. Three runs show the terminal paths hold; they are not a statistical sample.



THE SECOND SYSTEM

A companion to FORGE. Two multi-agent systems, both code-defensible: durable orchestration on Temporal, and an autonomous solution architect on LangGraph.

BUILT & OPERATED BY

SAGE

upswing

SALEM, OREGON