

Automating Systems Simulation at Scale: An LLM Pipeline for Multi-Framework Modeling

RESEARCH QUESTIONS — How can we make this process faster, and trust what comes out?

RQ1 Architecture

What design decisions turn a generic multi-agent LLM framework into a paradigm-aware, multi-paradigm code-synthesis workflow?

RQ2 Speed and trust

How does the pipeline behave compare to expert-authored, and what is its end-to-end latency relative to manual authoring?

RQ3 Practicality

What recurring failure modes survive even after the full guardrail stack is in place, and what disciplines mitigate them?

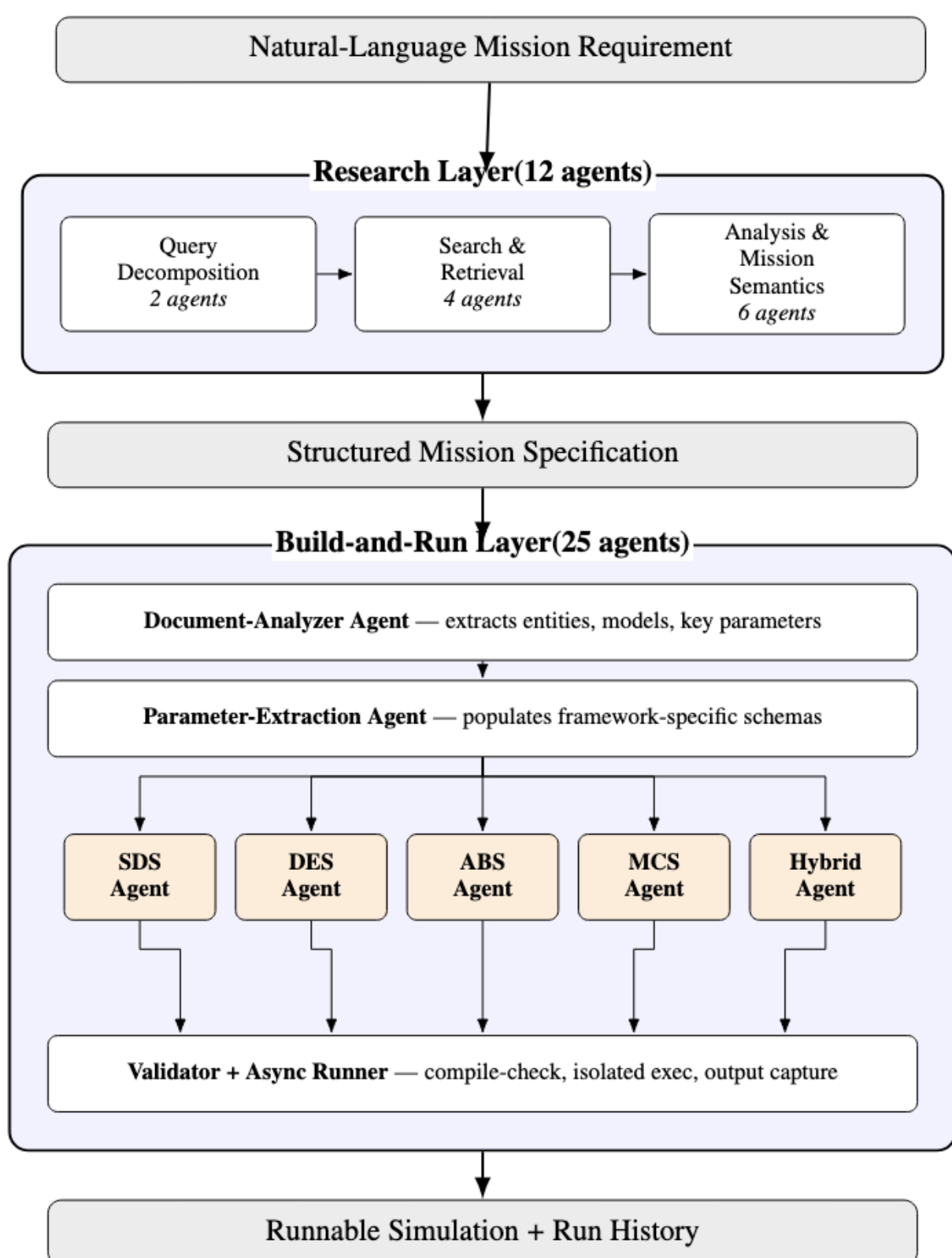
1 The problem

Weeks to months

per scenario, per paradigm - the binding constraint is specialist availability, not compute.

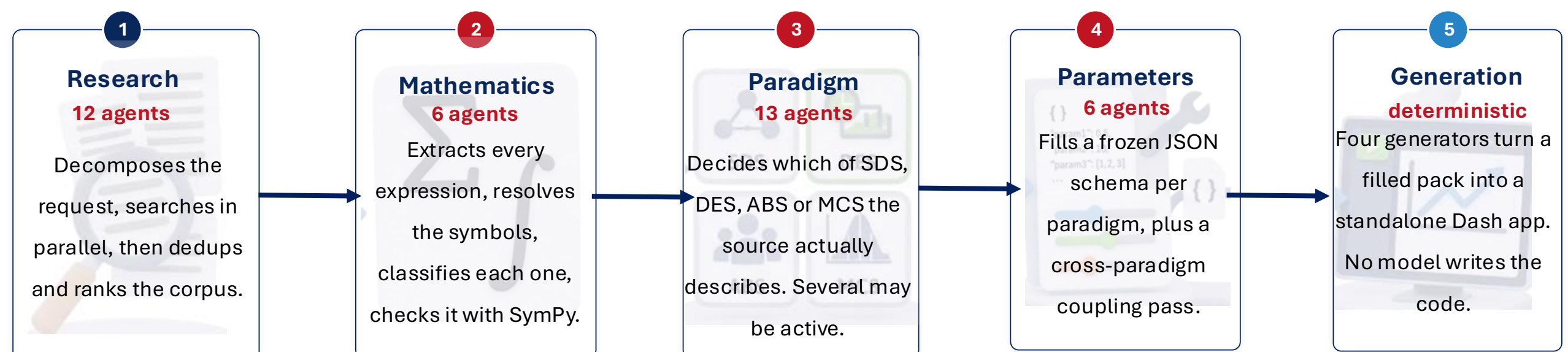
- 1) Choose the paradigm
- 2) Manually create the system
- 4) Decision making and support

2 The architecture



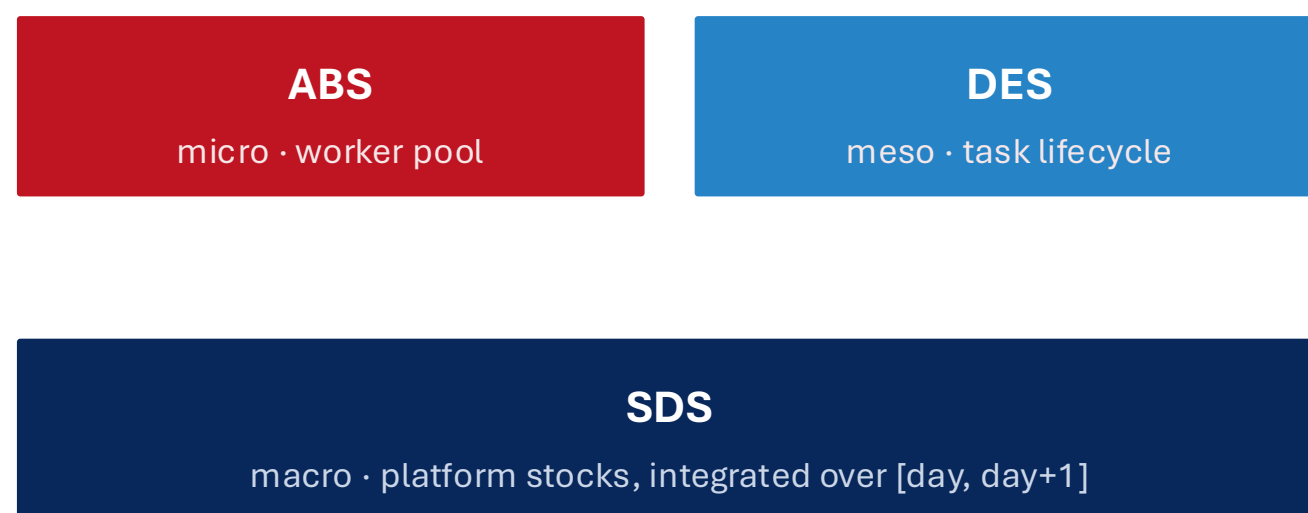
Two cooperating layers, one typed contract between them.
Nothing propagates until it type-checks.

3 Five synthesis stages



The agents produce the specification. The generator produces the code. That split is what makes the output auditable.

4 Coupling the paradigms



Exogenous terms enter derivatives(), so they are integrated rather than added at display time — mass balance holds by construction.
Solo simulations are imported unmodified and keep running standalone.

5 Trustworthiness

Trustworthiness is architectural, not a property of the model you picked.

Architecture checks the ID. Behaviour checks the alibi.

The "fake mustache" problem: structurally valid, behaviorally suspicious.

Validation here is entirely static - a compile check, a trap scan, and an agent reading the emitted code - so presence is not effect, and a curve can still be wrong.

6 The validation bug zoo

Failures that survived the current guardrails and became the next validation agenda.

Topology bug
Diagrams give structure, not coefficients.

Crossing-link bug
Curved links that cross are still lost.

Backbone-drift bug
Model upgrades can re-open old failures.

Semantic-validation bug
Runs pass presence checks yet drift outside tolerance.

Hybrid-gap bug
Only two coupling channels are wired end to end.

Dangling-dependency bug
A missing Stage 5 repair module silently no-ops.

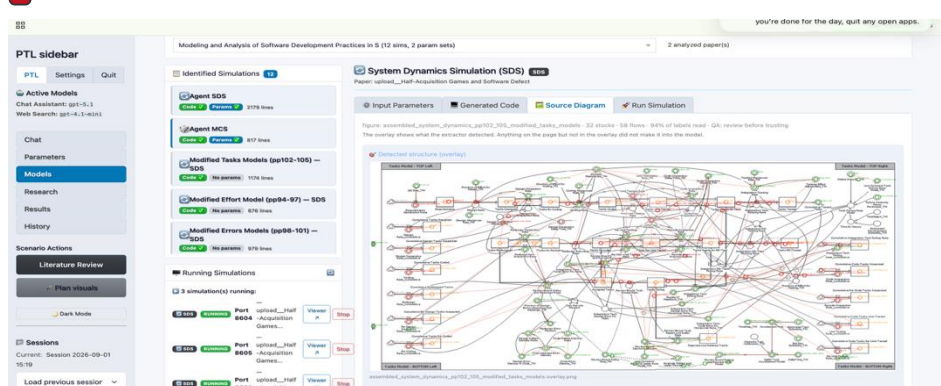
7 Speed

Sequential agent stages = wall clock time

End-to-end : [10, 26] minutes

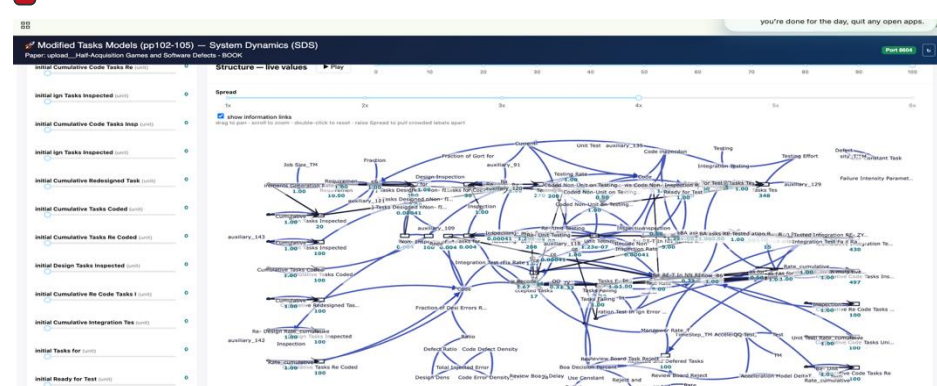
8 Examples Generated Simulations

SDS · model discovery



12 simulations found in the one paper: Agent SDS at 2,179 lines, three Modified-Model SDS variants at 876-1,174 lines, plus an MCS. Overlay reports 32 stocks, 58 flows, 94% of labels read.

SDS · running app



Modified Tasks Models (pp102-105) live on port 8604. Every initial stock is a slider; the structure view redraws against live values.

DES · running app



266 entities arrived, 167 completed, 17 failed, 16.7 per day across 239.9 simulated hours. The seven-stage lifecycle came from the paper's text.

ABS · running app



Day 38: 337 tasks, 213 failed, 124 open, zero completed. Reproduced headlessly over 120 days at pass rate 0.000.