

VeriMAS: A Verification-Gated Multi-Agent Framework for Trustworthy Large Language Model Orchestration on the Shop Floor

Tinku Talreja

Department of Computer Science &
Engineering
IET, SAGE University
Indore, India
Talrejatinku123@gmail.com

Prof Sapna Patel

Department of Computer Science &
Engineering
IET, SAGE University
Indore, India
sapna.patel@sageuniversity.in

Abstract—Large language model (LLM)-based multi-agent systems have demonstrated, in simulation, that they can negotiate production allocation, reason about disruptions, and coordinate shop-floor functions. What none of them yet offer is a principled answer to the question a plant manager will ask first: what stops an agent from doing something wrong to a physical machine? This paper proposes VeriMAS, a multi-agent framework whose defining structural commitment is that no agent proposal reaches equipment without passing an independent verification gate. The architecture comprises four role-specialized LLM agents grounded through retrieval-augmented generation over plant documentation; a verification layer combining digital twin simulation, a deterministic rule-based safety envelope, and calibrated risk scoring; a graduated escalation policy that routes uncertain proposals to a human oversight console with the agent's reasoning and the twin's counterfactual trace; and an execution layer confined to auditable connectors. We specify the decision workflow and escalation algorithm, discuss implementation with locally deployed open-weight models against latency and confidentiality constraints, and define a proposed evaluation methodology—simulation testbed, baselines, ablations, and metrics including unsafe-proposal interception rate—explicitly identified as future experimental work rather than reported results. The framework converts trust in shop-floor agents from an assumption into an architectural property, and each gate component yields a testable hypothesis about the cost and value of verification.

Keywords—Large Language Models, Multi-Agent Systems, Verification, Digital Twin, Manufacturing, Safety, Human Oversight

I. INTRODUCTION

Research on LLM-based agents in manufacturing has advanced along a consistent trajectory: from language models that answer questions about production to agent collectives that make production decisions. Role-specialized multi-agent designs now negotiate machine allocation on simulated shopfloors [5], internalize carbon objectives in their deliberations [6], and restructure their own coordination topologies in response to equipment health signals [7]. The architectural creativity of this literature is not in doubt. Its deployability is.

The obstacle is structural rather than incremental. LLMs are probabilistic instruments prone to fabrication in jargon-dense industrial contexts [10], [12], and current orchestration systems largely execute whatever their decision agents conclude. In simulation this is a benign shortcut; connected to a press line or a paint oven, it is a hazard that no accuracy percentage can excuse. Surveys of foundation-model agents in industrial automation consistently identify safety assurance as the field's binding constraint [8], and our own reading of the literature concludes that verification before actuation is the most valuable open design problem in the area.

This paper answers that problem with an architecture. VeriMAS (Verification-gated Multi-Agent System) accepts the premise of the orchestration literature—that role-specialized LLM agents are a productive way to organize shop-floor intelligence—and rejects its implicit trust model. In VeriMAS, agent output is a proposal, never a command. Every proposal must pass an independent verification gate comprising digital twin simulation, a deterministic safety envelope, and calibrated risk scoring; proposals that fail or score as uncertain are escalated to a human console with the evidence needed to decide quickly. The contribution is threefold: (1) a reference architecture and decision workflow for verification-gated agent orchestration; (2) an escalation algorithm that makes the autonomy-oversight trade-off an

explicit, tunable policy rather than an emergent accident; and (3) a proposed evaluation methodology, with baselines and metrics, designed so that the framework's central claims are falsifiable. Consistent with responsible reporting, no experimental results are claimed; Section VII is a plan, not a report.

II. RELATED WORK

Classical agent-based manufacturing control established decentralized negotiation as a viable organizing principle decades before LLMs, along with the insight that industrial adoption stalls when agent behavior cannot be certified [1]. LLM-based revivals of this program assign models to negotiation and decision roles [5], extend objectives toward sustainability [6], and self-organize agent networks [7]. Grounding techniques—retrieval-augmented generation [3] and its industrial, jargon-aware refinements [11]—address what agents know; reasoning-and-acting patterns [2] address how they act. The missing layer, addressed here, is whether they should: prior couplings of agents to digital twins gesture at pre-execution simulation [9] but do not specify gate semantics, escalation policy, or evaluation. VeriMAS also draws on digital twin state-synchronization research [13] for its simulation gate, and on the Industry 5.0 literature for the principle that consequential authority remains human [4], [14], with explainability treated as an adoption precondition rather than a refinement [15].

III. PROBLEM STATEMENT AND OBJECTIVES

We consider a discrete-manufacturing environment in which production events—machine faults, order changes, quality alarms—require coordinated responses spanning scheduling, maintenance, and quality functions. An LLM-based multi-agent layer is desired for its adaptability and explainability, but the plant imposes four hard constraints: (C1) no action may violate machine, process, or personnel safety limits; (C2) actions must be feasible under current plant state, not the state the model imagines; (C3) response latency must respect operational deadlines, which differ by decision class; and (C4) all decisions must be auditable after the fact. The design objectives follow: O1, preserve the flexibility and natural-language transparency of LLM agents; O2, guarantee C1 deterministically, independent of model behavior; O3, enforce C2 probabilistically through simulation with quantified confidence; O4, make the residual risk of any autonomous action explicit and bounded by policy; and O5, degrade gracefully—when verification cannot certify a proposal, the system must fail toward human judgment, never toward silent execution.

IV. THE VERIMAS ARCHITECTURE

Fig. 1 presents the architecture. Four layers separate concerns that current systems conflate: proposing (agents), knowing (retrieval), checking (verification), and doing (execution), with human oversight positioned as a first-class layer rather than an exception handler.

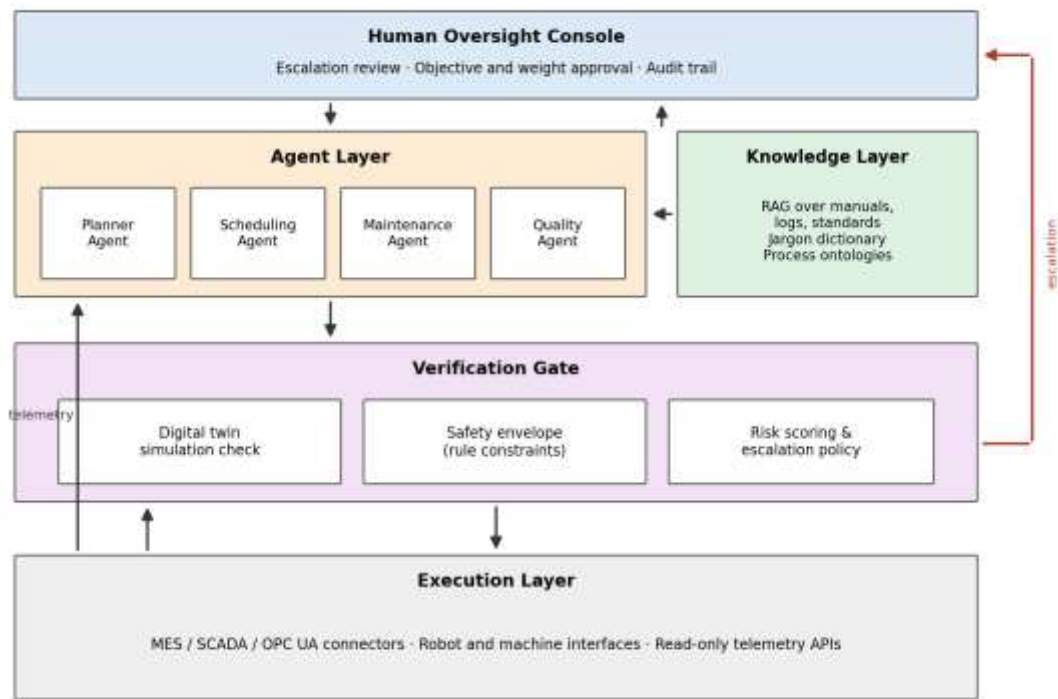


Fig. 1. The VeriMAS architecture. Agent proposals flow downward through the verification gate before reaching execution; telemetry flows upward; uncertain or failed proposals escalate to the human oversight console along the highlighted path.

A. Agent Layer

Four role-specialized agents—planner, scheduling, maintenance, and quality—deliberate over production events. Roles follow the division of labor validated in prior multi-agent designs [5] but with a restricted contract: an agent's output is a structured proposal object containing the intended action, its parameters, the agent's reasoning trace, the retrieved evidence it relied on, and a self-assessed confidence. Agents may negotiate among themselves (for instance, maintenance and scheduling agents trading a maintenance window against a due date) before submitting a joint proposal; negotiation transcripts attach to the proposal for auditability.

B. Knowledge Layer

All agent reasoning is grounded through retrieval-augmented generation over plant documentation—equipment manuals, process standards, historical work orders—using jargon-aware retrieval in the style of reflection-based query augmentation [11], a curated jargon dictionary, and process ontologies that normalize entity references across documents. The knowledge layer is read-only and versioned: every proposal records the document versions it retrieved, so that a later audit can reconstruct exactly what the agent knew.

C. Verification Gate

The gate is the framework's core commitment and comprises three independent checks. (1) Digital twin simulation: the proposal is executed against a synchronized twin of the affected cell or line [9], [13]; the twin returns predicted effects on throughput, quality, energy, and constraint violations. (2) Safety envelope: a deterministic rule engine—deliberately non-learned, so its behavior is certifiable—checks the proposal against hard limits (speeds, temperatures, interlocks, lockout states, personnel zones). A safety-envelope failure is non-negotiable and cannot be overridden by any model output, satisfying O2. (3) Risk scoring: a calibrated scorer combines the twin's predicted-effect uncertainty, the agent's self-assessed confidence, the novelty of the proposal relative to historical actions, and the blast radius of the affected equipment into a scalar risk score R .

D. Escalation Policy and Human Oversight

Policy thresholds partition R into three bands: autonomous execution (R below the lower threshold), execute-with-notification (intermediate), and mandatory human review (above the upper threshold). Thresholds are set per decision class—a schedule resequencing tolerates more autonomy than a parameter change on a safety-relevant process—and are owned by plant management, not by the system. The oversight console presents escalated proposals with the agent's reasoning, the retrieved evidence, and the twin's counterfactual trace, so that a human can decide in minutes what the agent deliberated in seconds. Every decision, autonomous or human, enters an append-only audit log, satisfying C4.

The console's design borrows a principle from safety-critical interface research: escalations must be decidable, not merely visible. Each escalation therefore leads with the twin's predicted deltas against current state (what changes if approved), followed by the safety-envelope verdict, the agent's reasoning, and one-click access to the retrieved sources. Reviewers can approve, reject, or modify-and-resimulate, and every modification re-enters the gate rather than bypassing it. Response-time expectations are part of the policy: decision classes carry maximum review latencies, and an unreviewed escalation times out to the safe default for its class—never to execution.

E. Execution Layer

Verified proposals execute exclusively through typed connectors to the MES, SCADA, and OPC UA interfaces, each connector exposing the minimal action vocabulary its decision class requires. Agents hold no credentials and cannot reach equipment except through the gate; telemetry APIs are read-only. This confinement means that even a fully compromised or hallucinating agent can, at worst, generate proposals that the gate must reject—shrinking the trusted computing base from the model to the gate.

V. DECISION WORKFLOW AND ESCALATION ALGORITHM

Fig. 2 traces the life of a decision. A production event or operator request activates the relevant agents; a grounded proposal is produced, optionally after inter-agent negotiation; the gate simulates and checks it; the risk score routes it to execution or escalation; and the realized outcome feeds back into agent context and twin calibration, closing the learning loop without ever letting learning bypass verification.

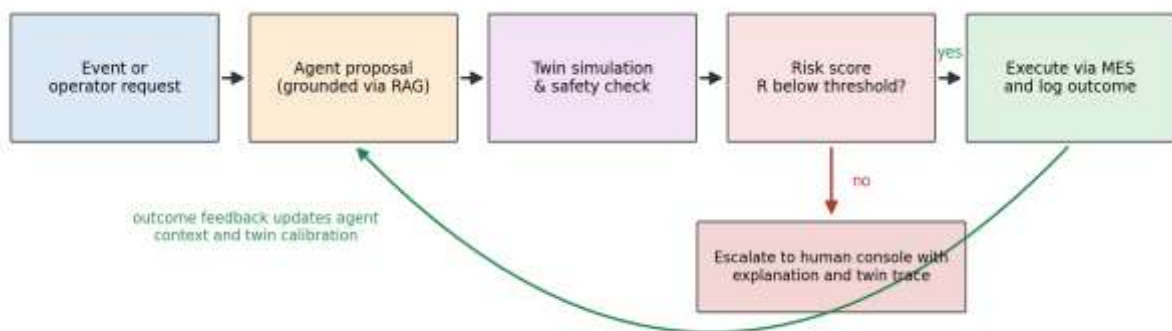


Fig. 2. The VeriMAS decision workflow. The green feedback path improves future proposals and twin fidelity; the red path escalates uncertain proposals to human review.

The escalation logic is deliberately simple enough to certify:

```

procedure GATE(proposal p, class c):
  if SafetyEnvelope(p) = FAIL:          return REJECT(p, reason)
  (effects, sigma) := TwinSimulate(p)
   $R := w1 \cdot \sigma + w2 \cdot (1 - \text{conf}(p)) + w3 \cdot \text{novelty}(p) + w4 \cdot \text{blast}(c)$ 
  if  $R < \text{theta\_lo}(c)$ :                return EXECUTE(p); log
  else if  $R < \text{theta\_hi}(c)$ :           return EXECUTE(p); notify; log
  else:                               return ESCALATE(p, effects, trace)
  
```

The weights w_1 – w_4 and thresholds θ_{lo} , θ_{hi} per decision class c are governance parameters exposed to plant management through the oversight console. Setting θ_{lo} to zero for a class disables autonomy entirely for that class—useful during commissioning—while raising it gradually as audit history accumulates provides a principled autonomy ramp. Novelty is computed as distance from the plant's historical action distribution, so the system is conservative precisely where its experience is thinnest.

VI. IMPLEMENTATION CONSIDERATIONS

Confidentiality and cost argue for locally deployed open-weight models fine-tuned on the plant's document corpus rather than frontier-scale APIs; the agent roles do not require frontier capability so much as reliable structured output and grounded reading, and evaluation environments for factory-context LLMs suggest that data preparation dominates model scale as a performance determinant [10]. Latency is managed by decision class: agents and the twin run on plant-edge hardware for intermediate-loop decisions (minutes), while nothing in VeriMAS sits inside real-time control loops—by design, the framework addresses the slow loop where deliberation is affordable. The digital twin need not be plant-wide at the outset; the architecture admits incremental coverage, with unverifiable proposals (no twin available) defaulting to escalation, which preserves O5. Integration uses standard industrial interfaces (OPC UA, MQTT, MES APIs), and the connector vocabulary should be reviewed as part of the plant's existing management-of-change process.

Model and prompt engineering deserve equal discipline. Each agent role is defined by a versioned system prompt and a JSON output schema that the gate validates before any other processing—malformed proposals are rejected at parse time, which converts a class of model failures into cheap, observable events. Fine-tuning targets structured-output reliability and plant vocabulary rather than reasoning ability, using the knowledge layer's corpus as training material so that model and retrieval remain consistent. Observability is treated as a first-class requirement: token-level logs, retrieval traces, gate verdicts, and escalation outcomes flow into the same audit store, giving engineers a single timeline from event to actuation. Finally, deployment follows the autonomy ramp described in Section V—the system runs in shadow mode first, generating and gating proposals without execution authority, so that interception statistics accumulate before any connector is enabled.

VII. PROPOSED EVALUATION METHODOLOGY

No experimental results are reported in this paper; this section defines the study we argue the framework deserves, in sufficient detail to be executed or challenged. The evaluation environment is a simulated discrete-manufacturing shopfloor with injected disruption scenarios (machine failures, rush orders, quality drift), extended with a deliberately imperfect twin—since gate value depends on twin fidelity, fidelity itself is an experimental variable. Adversarial scenarios seed agents with misleading retrievals and prompt perturbations to elicit unsafe proposals, following the spirit of safe evaluation sandboxes [10] and calibrated scheduling benchmarks [12].

TABLE 1. PROPOSED EVALUATION DESIGN

Component	Specification	Purpose
Baselines	B1: rule-based dispatching; B2: single LLM agent, ungated; B3: multi-agent, ungated [5]; B4: VeriMAS	Isolate the contribution of multi-agent structure vs. the gate
Primary metrics	Unsafe-proposal interception rate; task success; makespan/tardiness; escalation precision and recall; decision latency percentiles	Measure safety and cost of verification jointly
Ablations	Remove twin check; remove safety envelope; remove risk scoring; vary twin fidelity; vary thresholds θ	Attribute value to each gate component
Hypotheses	H1: gate intercepts a large majority of induced unsafe proposals with bounded throughput cost; H2: escalation quality	Falsifiable claims the study confirms or refutes

	improves with novelty term; H3: value of gate rises as twin fidelity rises	
--	--	--

Table 1 specifies baselines, metrics, ablations, and hypotheses for the proposed study. All rows describe planned methodology; no results are claimed.

The decisive comparison is B3 versus B4 under adversarial injection: if the gate does not intercept substantially more unsafe proposals than it costs in throughput and latency, the framework's premise fails. We consider publication of that negative result as valuable as its confirmation.

VIII. LIMITATIONS AND FUTURE WORK

Three limitations bound the current proposal. First, the gate's probabilistic component inherits the twin's fidelity: a stale or miscalibrated twin can certify infeasible actions, which is why twin-fidelity variation is central to the proposed evaluation and why the safety envelope remains deterministic. Second, risk-score calibration requires incident-labeled history that new deployments lack; conservative cold-start thresholds and the autonomy ramp mitigate but do not remove this. Third, the framework addresses intermediate-loop orchestration and deliberately excludes real-time control, so its benefits do not extend to cycle-time-critical decisions. Future work includes a reference implementation on an open simulation testbed, formal verification of the safety envelope and escalation logic, extension of the risk model with environmental-impact terms so that sustainability enters agent deliberation directly [6], and field studies of the oversight console with production staff—since the framework's ultimate test is not whether the gate works, but whether operators trust what passes through it.

IX. CONCLUSION

The multi-agent LLM literature has shown that language models can fill the roles classical manufacturing agents defined; it has not shown that plants can afford to let them. VeriMAS reframes the problem: autonomy is not a property of capable agents but of a system that can bound what capable agents are permitted to do. By making every agent output a proposal, every execution a verified event, and every uncertain case a human decision supported by simulation evidence, the framework converts trust from an assumption into an architecture—and, through its proposed evaluation, into a measurable one. If verification gates prove cheap relative to the failures they intercept, the path from simulated shopfloors to real ones becomes an engineering program rather than a leap of faith.

REFERENCES

- [1] P. Leitão, “Agent-based distributed manufacturing control: A state-of-the-art survey,” *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 979–991, 2009.
- [2] S. Yao et al., “ReAct: Synergizing reasoning and acting in language models,” in *Proc. International Conference on Learning Representations (ICLR)*, 2023.
- [3] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [4] M. Breque, L. De Nul, and A. Petridis, “Industry 5.0: Towards a sustainable, human-centric and resilient European industry,” *European Commission, Directorate-General for Research and Innovation*, Brussels, 2021.
- [5] “A Large Language Model-based multi-agent manufacturing system for intelligent shopfloor,” *arXiv preprint arXiv:2405.16887*, 2024. [Online]. Available: <https://arxiv.org/abs/2405.16887> *
- [6] “Large language model-driven multi-agent systems for improving production efficiency and reducing carbon emissions in manufacturing,” *Computers & Industrial Engineering*, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0360835225004450> *
- [7] “Self-Evolving Multi-Agent Network for Industrial IoT Predictive Maintenance,” *arXiv preprint*, 2026. [Online]. Available: <https://arxiv.org/pdf/2602.16738> *
- [8] “Foundation-Model-Based Agents in Industrial Automation: Purposes, Capabilities, and Open Challenges,” *arXiv preprint*, 2026. [Online]. Available: <https://arxiv.org/pdf/2605.02592> *
- [9] “Integrating Large Language Model Agents with Digital Twins for Industrial Autonomous Systems,” *arXiv preprint*, 2026. [Online]. Available: <https://arxiv.org/pdf/2606.20761> *

- [10] “FactoryLLM: A Safe and Open-Source AI Playground for Evaluating LLMs in Smart Factories,” arXiv preprint, 2026. [Online]. Available: <https://arxiv.org/pdf/2606.14119> *
- [11] “Golden-Retriever: High-Fidelity Agentic Retrieval Augmented Generation for Industrial Knowledge Base,” arXiv preprint, 2024. [Online]. Available: <https://arxiv.org/pdf/2408.00798> *
- [12] “DynaSchedBench: Calibrated Dynamic Scheduling Benchmarks and Observability Paradox in LLM-based Scheduling Agents,” arXiv preprint, 2026. [Online]. Available: <https://arxiv.org/pdf/2605.27566> *
- [13] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2405–2415, 2019.
- [14] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, “Industry 4.0 and Industry 5.0—Inception, conception and perception,” Journal of Manufacturing Systems, vol. 61, pp. 530–535, 2021.
- [15] “Explainability as the key ingredient for AI adoption in Industry 5.0 settings,” Frontiers in Artificial Intelligence, vol. 6, 2023. [Online]. Available: <https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2023.1264372/full> *
- [16] “Toward Epistemic Stability: Engineering Consistent Procedures for Industrial LLM Hallucination Reduction,” arXiv preprint, 2026. [Online]. Available: <https://arxiv.org/pdf/2603.10047> *