

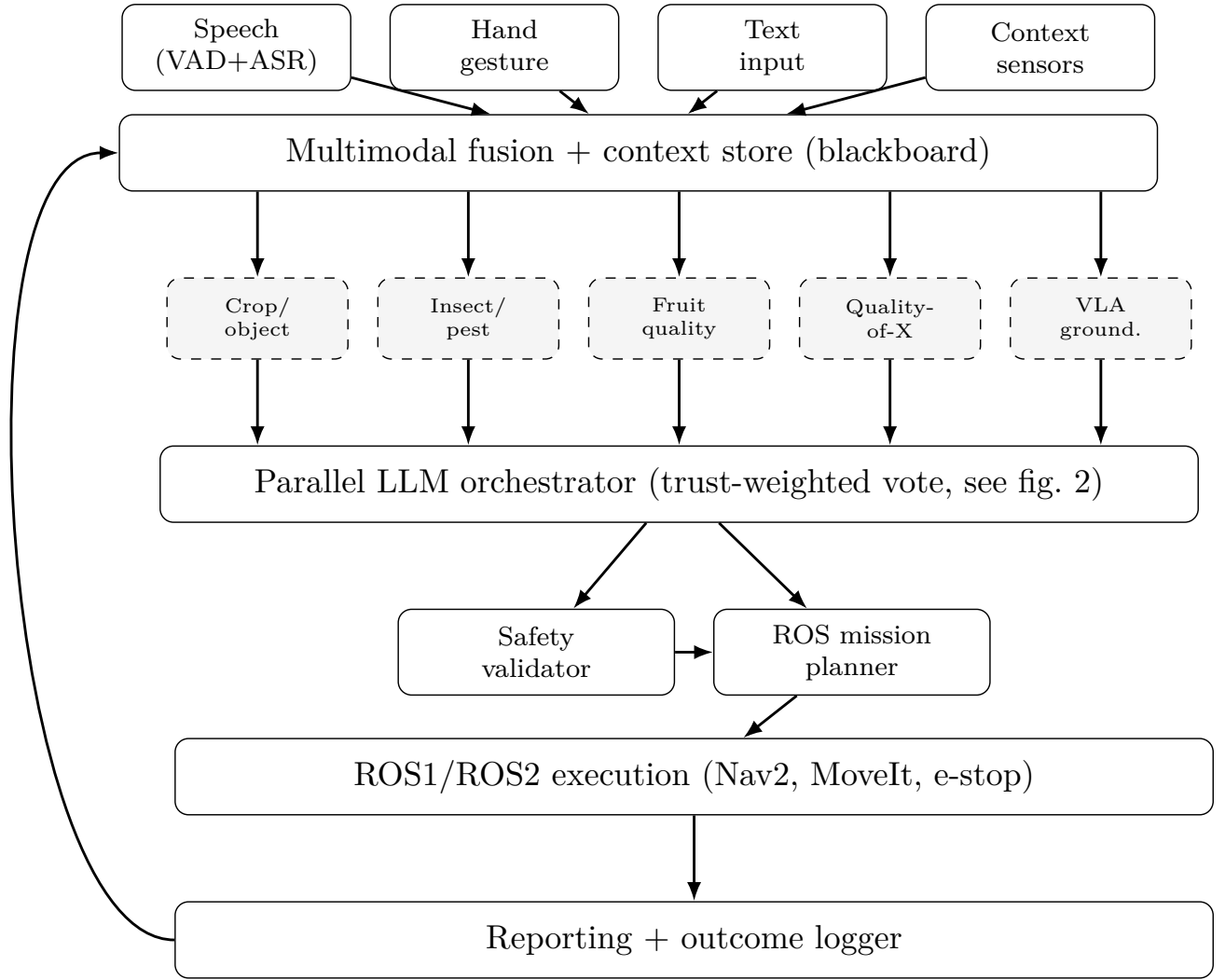


Figure 1: Overview: input, perception plugins, LLM orchestrator, ROS execution, reporting. Cf. [1], [2].

1 Concurrency

Input fan-in (speech/gesture/text/sensors) independent producers, no ordering needed.

Perception plugin bus parallel tasks/workers; timeout per plugin so one slow plugin can't block a mission.

LLM voters async fan-out (I/O-bound); a self-hosted model needing an accelerator gets a dedicated worker instead.

Trust table single writer (actor or atomic update), not a shared table behind a lock everyone fights over.

Blackboard snapshot reads; the feedback loop writes without invalidating requests already in flight.

Safety-critical fast path runs outside the ensemble's thread/task pool, so it can't be blocked by it.

2 Terms

VAD (Voice Activity Detection) detects speech vs. silence/noise in audio.

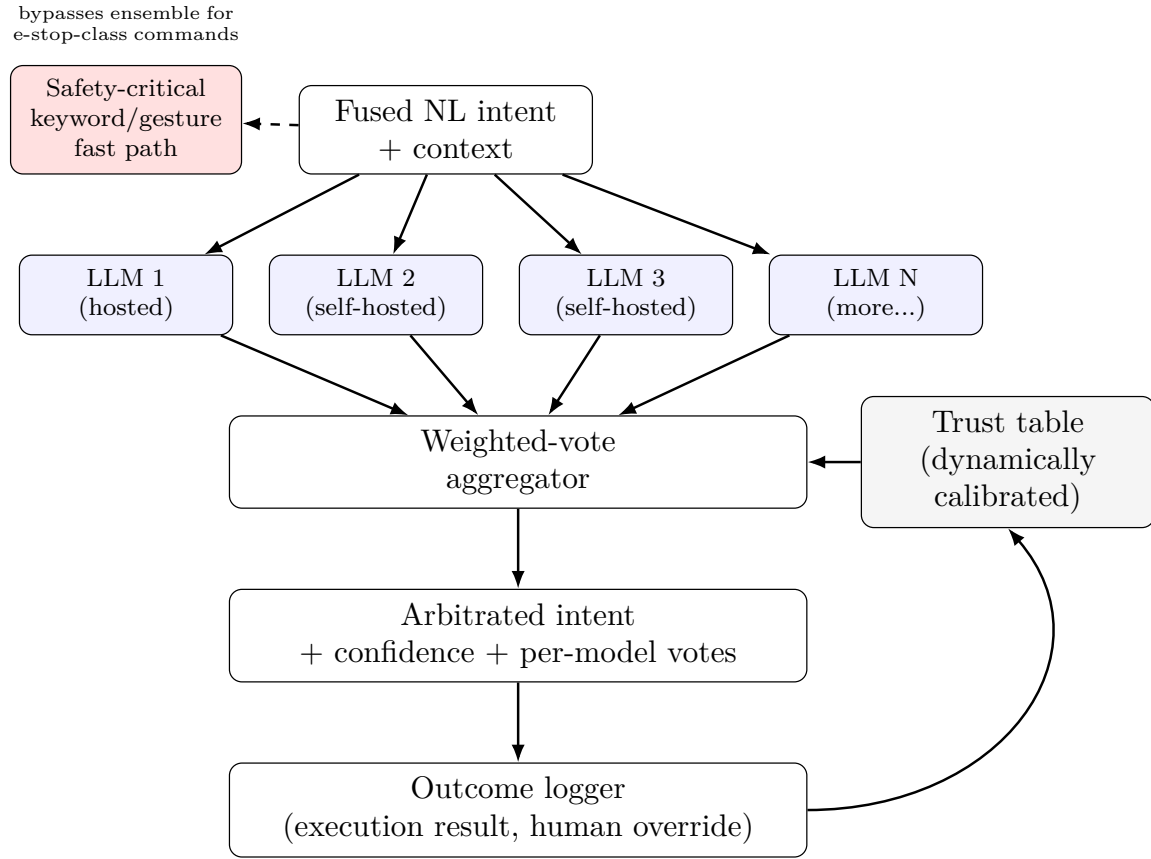


Figure 2: LLM orchestrator: parallel voters, trust-weighted vote [3], [4], [5], stop commands bypass the ensemble.

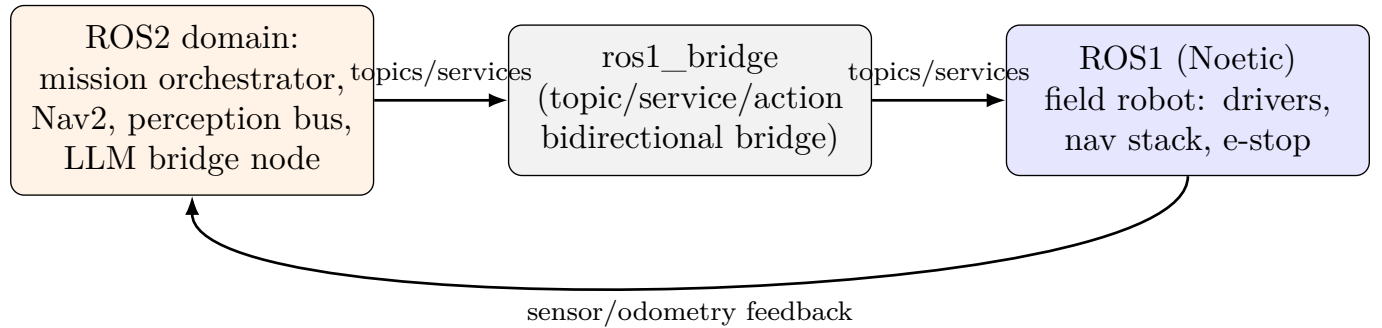
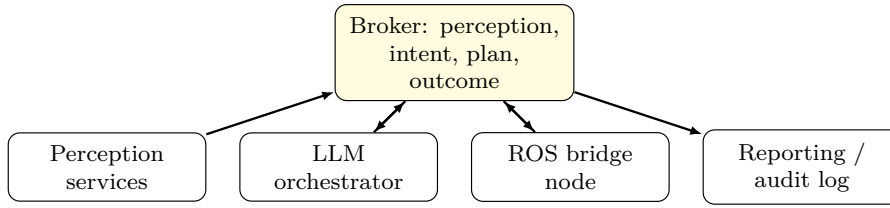
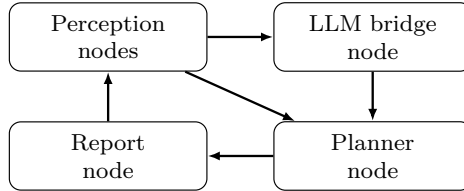


Figure 3: ROS2 orchestration bridged to the existing ROS1 robot via ros1_bridge [6], [7].

(a) Kafka/Redpanda hub-and-spoke



(b) ROS2 DDS mesh only



(c) Hybrid: DDS for control loop, broker for analytics/orchestration

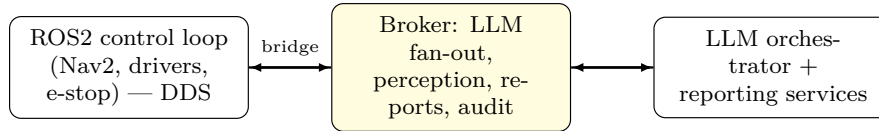


Figure 4: Messaging backbone options: (a) Kafka/Redpanda broker [8], (b) ROS2 DDS only, (c) hybrid.

ASR (Automatic Speech Recognition) converts speech to text.

NL natural language.

LLM large language model.

VLA (Vision-Language-Action model) a model that maps camera input plus an instruction directly to robot actions.

Blackboard shared store that every component reads from and writes to (pose, map, recent perception results, task history).

Quality-of-X generic plugin slot for a domain-specific inspection task (weld defects, thermal hotspots, etc.) — same interface as the crop/insect/fruit plugins, different label set.

Trust table per-model weight used to combine LLM votes, updated from past agreement/outcomes rather than fixed by hand.

ROS1 / ROS2 Robot Operating System, versions 1 and 2 (ROS2 uses DDS; ROS1 does not).

ros1_bridge bridges topics, services, and actions between a ROS1 and a ROS2 system.

DDS (Data Distribution Service) the publish/subscribe transport standard ROS2 is built on.

Nav2 the ROS2 navigation stack (path planning, obstacle avoidance).

MoveIt ROS motion-planning framework, mainly for robot arms.

e-stop emergency stop.

Kafka / Redpanda distributed message-broker systems; Redpanda is API-compatible with Kafka but lighter-weight.

Broker a middleman service that receives messages from senders and forwards them to whoever subscribed, so senders and receivers don't talk to each other directly.

Hub-and-spoke a layout where every node talks only to a central point (the broker), never directly to each other — like spokes on a wheel.

Mesh a layout where nodes talk directly to each other, with no central point.

Fan-out sending one message out to several receivers at once.

Audit log a record of every message/decision kept for later review.

3 References

1. C. E. Mower, Y. Wan, H. Yu, et al., “A robot operating system framework for using large language models in embodied AI,” *Nature Machine Intelligence*, vol. 8, pp. 313–325, 2026. <https://arxiv.org/abs/2406.19741>
2. M. A. Zuzuárregui and S. Carpin, “Leveraging LLMs for mission planning in precision agriculture,” 2025. <https://arxiv.org/abs/2506.10093>
3. X. Wang et al., “Self-consistency improves chain of thought reasoning in language models,” *ICLR* 2023. <https://arxiv.org/abs/2203.11171>

4. J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, and J. Zou, “Mixture-of-agents enhances large language model capabilities,” 2024. <https://arxiv.org/abs/2406.04692>
5. S. Arora, E. Hazan, and S. Kale, “The multiplicative weights update method: a meta-algorithm and applications,” *Theory of Computing*, vol. 8, no. 1, pp. 121–164, 2012. (not on arXiv) <https://theoryofcomputing.org/articles/v008a006/v008a006.pdf>
6. Open Robotics, “ROS 2 design,” 2024. <https://design.ros2.org/>
7. M. Arguedas, S. Ragnarok, D. Thomas, A. Nash, G. Biggs, and M. A. Gutierrez, “REP 2000 – ROS 2 releases and target platforms,” 2020. <https://ros.org/repos/rep-2000.html>
8. J. Kreps, N. Narkhede, and J. Rao, “Kafka: a distributed messaging system for log processing,” NetDB Workshop, 2011. (not on arXiv) <https://kafka.apache.org/books-and-papers/>