AIAT 2026 · 6th International Conference on AI and Application Technologies · Hosei University, Tokyo · Dec 11–13, 2026

Session 8 · Embodied AI, Energy Systems and Cross-Domain Intelligent Applications · Room E106 · Paper T354-A

Edge-Based Multimodal Perception and Safety Arbitration for an Embodied AI Companion on a Quadruped Robot

Po-Yu Lu, Hsu Huang, Yu-Tong Wu, Ju-En Chen, Pei-Chen Yang, Wei-Feng Tung*

Department of Information Management, Fu Jen Catholic University, New Taipei City, Taiwan

Presented by **Prof. Wei-Feng Tung** · * corresponding author

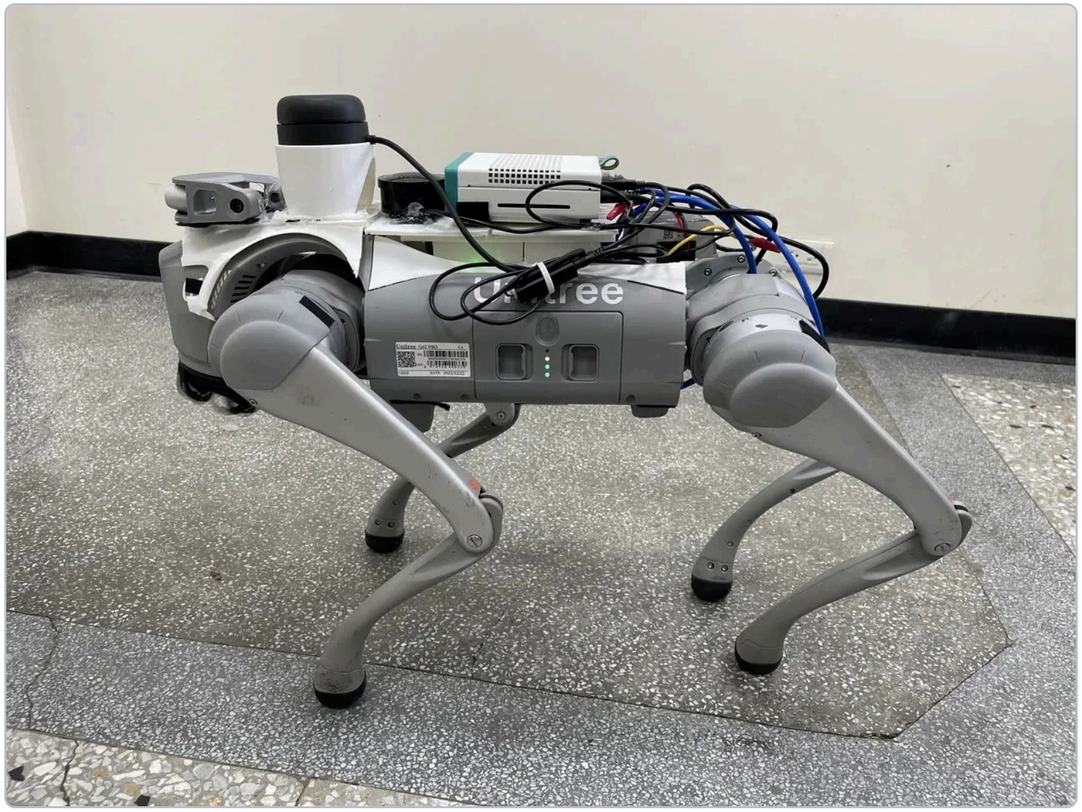


Fig. 1 PawAI: a Unitree Go2 Pro carrying an external NVIDIA Jetson, RealSense D435 and RPLIDAR. Source: PawAI project, roy422.dev

Keywords: embodied AI · human–robot interaction · multimodal perception fusion · large language models

Motivation: who has the final say over a moving robot?

- LLMs are moving from chat windows into the physical world (embodied AI).
- A consumer quadruped such as the Unitree Go2 Pro is script-driven out of the box.
- Adding face, gesture, pose, object, speech and an LLM creates many sources of action requests.

Research question. When several modules ask the robot to act at the same time, who has final control? And when the cloud fails, perception is wrong, or the LLM makes a mistake, how does the robot return to a safe state?

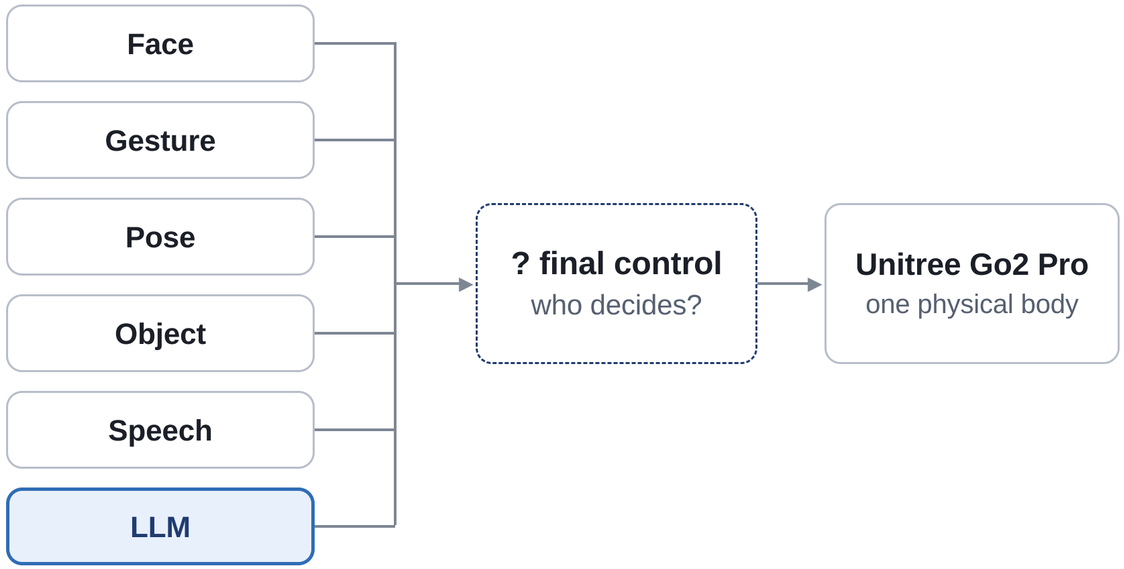


Fig. 2 Many perception and reasoning modules compete for one robot body.

Our design rule: the LLM only proposes. Only the Interaction Executive may release a physical motion.

Three challenges for an embodied companion at the edge

1 Limited edge compute

- One NVIDIA Jetson Orin Nano with 8 GB unified memory.
- Face, gesture, pose, object and speech must run at the same time.
- Example: RTMPose alone used 91–99% of the Jetson GPU.

Design response. Benchmark on the real device; prefer CPU models; keep the GPU for YOLO.

2 Heterogeneous perception

- Each module has its own model, frame rate and output format.
- 16 candidate models were benchmarked before the pipeline settled.
- Modules must not command the robot or call each other directly.

Design response. Shared ROS 2 event contracts: go2_interfaces, pawai_contracts.

3 Physical safety

- The robot really moves; a wrong action can hurt people or itself.
- On Go2 Sport Mode a very small Move{x} is ignored and the robot keeps sliding: zero speed ≠ stop.
- Cloud outages and LLM errors must fail safe.

Design response. One central arbiter, a deterministic Safety Gate, explicit StopMove.

Source: paper abstract; PawAI project, roy422.dev

Hardware platform: a consumer quadruped plus an edge computer

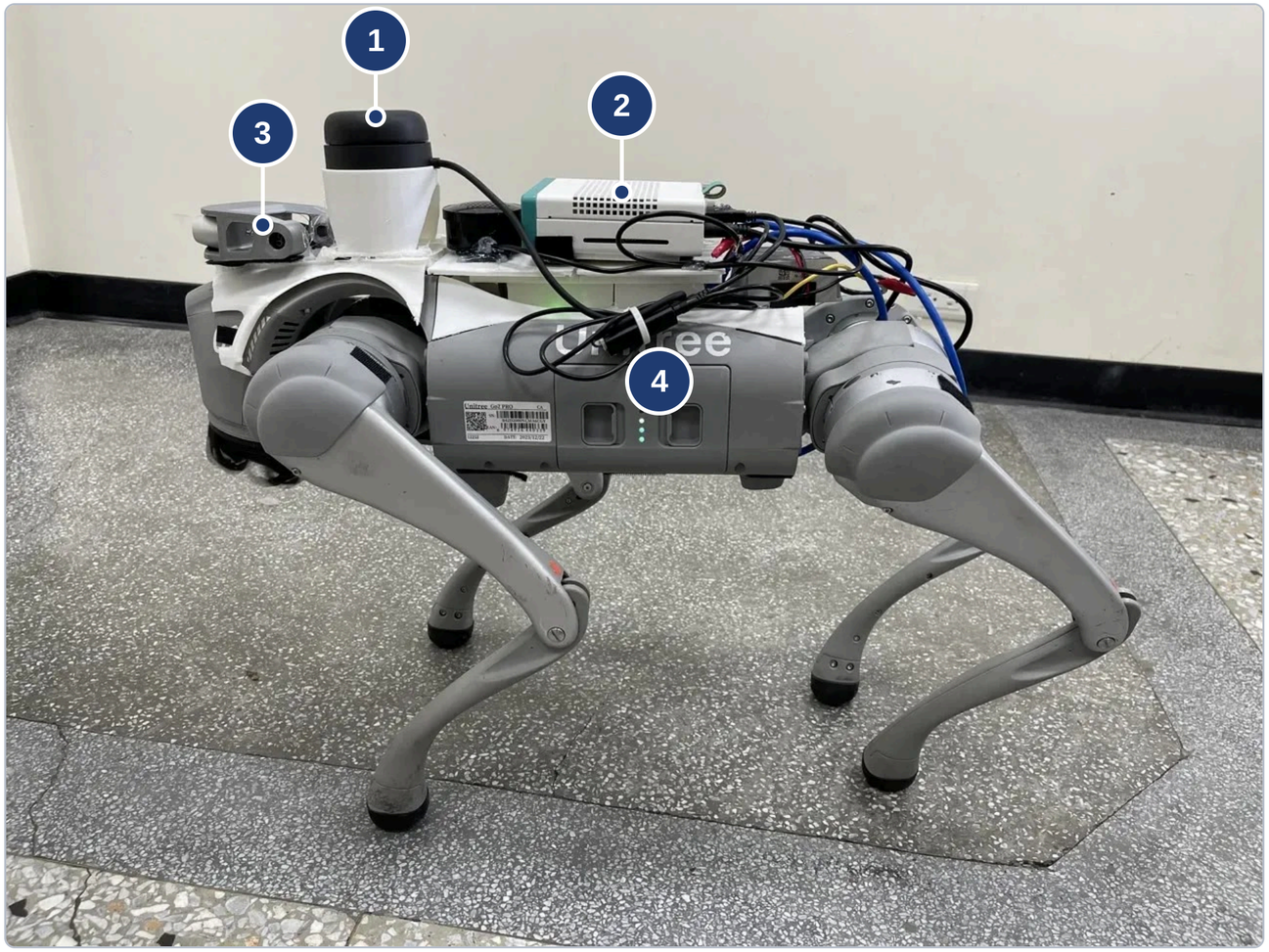


Fig. 3 PawAI hardware stack on the Unitree Go2 Pro (side view, same robot as Fig. 1). Numbers match the list on the right.

1. **RPLIDAR (2D LiDAR)**
   Used only for supervised, experimental navigation.

2. **NVIDIA Jetson Orin Nano · 8 GB**
   ROS 2 runtime, perception, Brain, Executive, local speech fallbacks.

3. **Intel RealSense D435**
   RGB + depth input for all vision modules.

4. **Unitree Go2 Pro**
   Sport API, Megaphone audio and motion, reached over WebRTC.

5. **Audio (on board)**
   External microphone and USB speaker (not labelled in the photo).

Why it is hard: the Go2 Pro has no official EDU SDK path, so control goes through a community WebRTC bridge. Power, vibration, USB contacts and cooling all had to be solved on the robot; the team 3D-printed a custom back cover to mount the stack.

System architecture: perception, decision, action

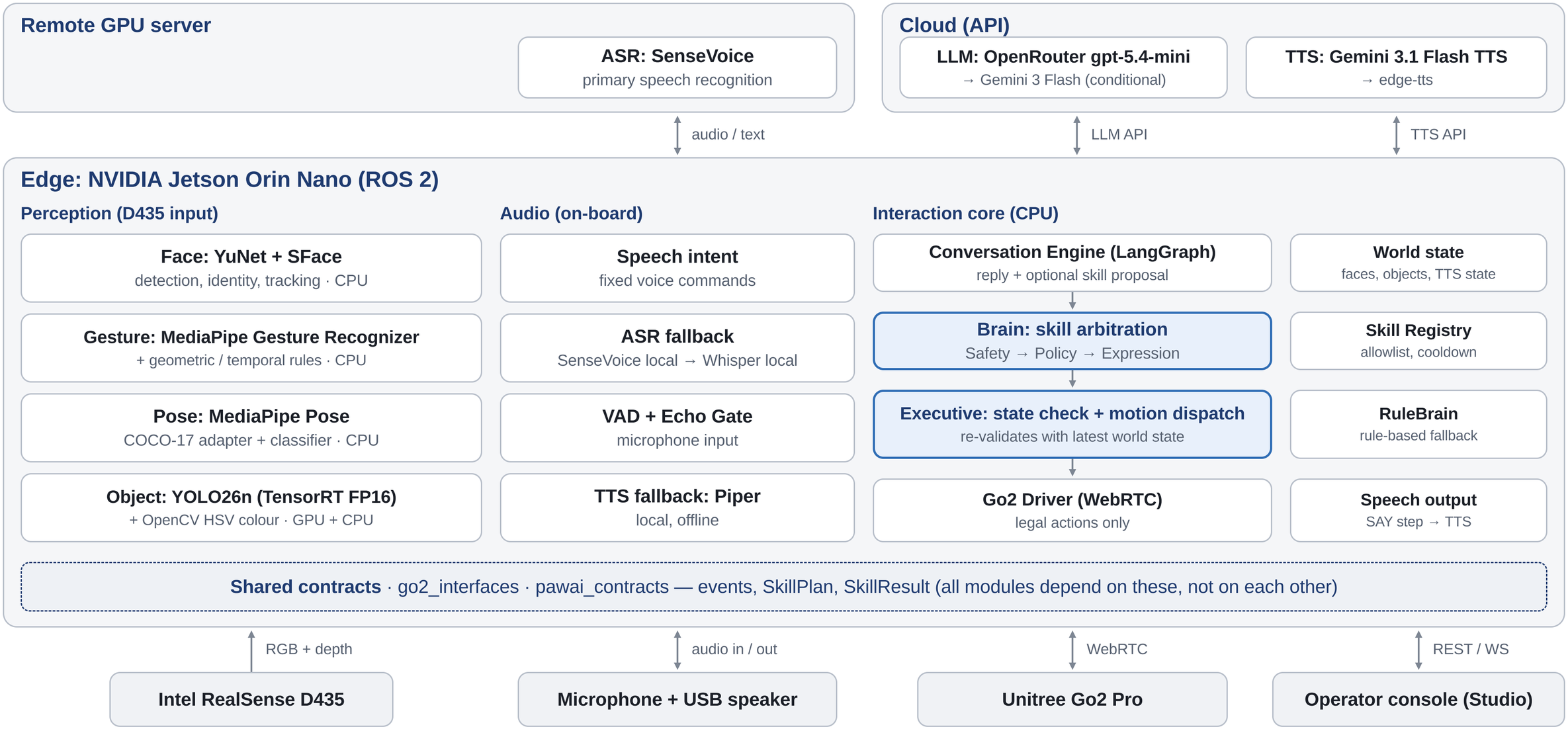


Fig. 5 System architecture. Ten ROS 2 packages grouped by deployment tier; perception, decision and action communicate only through shared contracts. Highlighted: the two arbitration points.

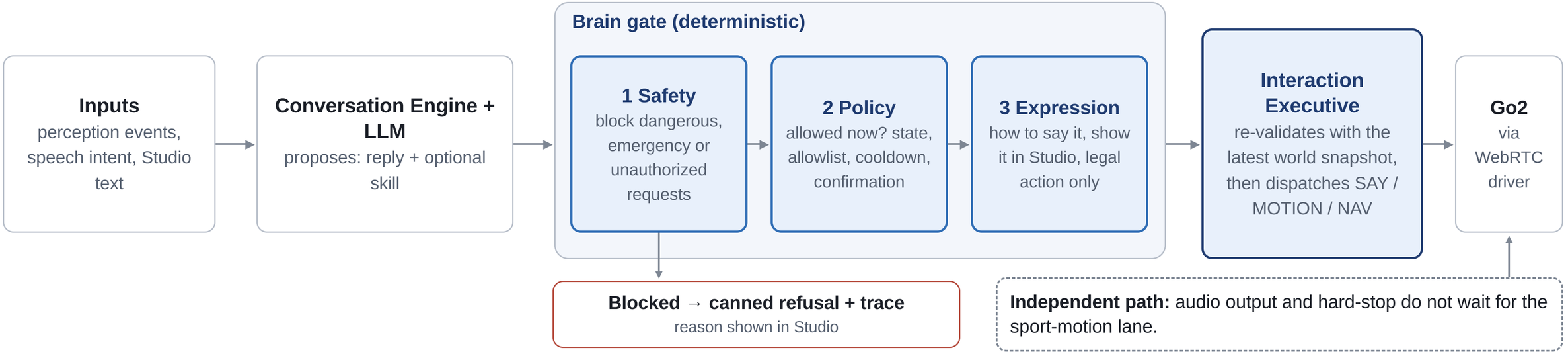
Multimodal perception: models chosen for a shared 8 GB budget

Table 1 Perception and speech models, where they run, and reported rates.

Capability	Model and pipeline	Runs on	Reported rate	Known limit
Face	YuNet 2023mar detection + SFace 2021dec embedding + IOU tracking / identity hysteresis	Jetson CPU	**71.3 FPS (YuNet, single-model benchmark)**	Low light, distance, tracking jitter
Gesture	MediaPipe Gesture Recognizer + custom geometric / temporal rules	Jetson CPU	**≈ 7.2 FPS, GPU 0%**	Static gestures more reliable; dynamic wave not yet passed
Pose	MediaPipe Pose + COCO-17 keypoint adapter + geometric classifier	Jetson CPU	**≈ 18.5 FPS**	Awkward viewpoints → false landmarks
Object & colour	YOLO26n ONNX, onnxruntime-gpu TensorRT EP FP16 + OpenCV HSV (12 colours)	Jetson GPU + CPU	**≈ 15 FPS**	Small far objects; cup / bottle confusion
Speech (ASR)	SenseVoice (GPU server) → SenseVoice local (Jetson) → Whisper local	Remote GPU / Jetson	**— (fallback chain)**	Mic, VAD, echo, network

Selection rule. Not single-model accuracy, but: does the model still survive on 8 GB unified memory when face, pose, gesture, object and speech run together? CPU-first choices leave the GPU to YOLO. Rates are as reported in project documentation, not a controlled benchmark.

Interaction Executive: the LLM proposes, the Executive decides



Output contract of the Conversation Engine

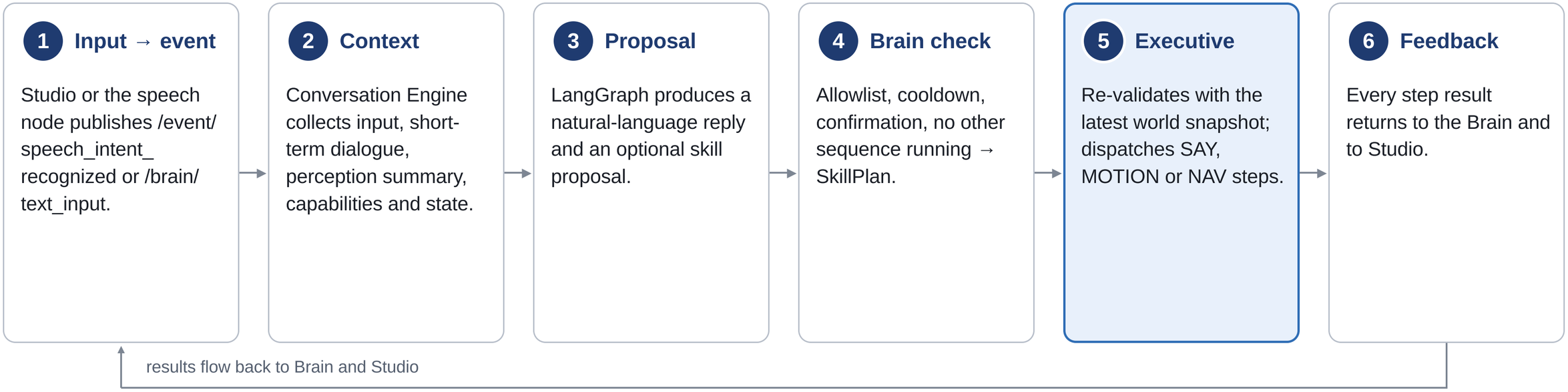
```
{
  "reply": "I am PawAI, ...",
  "skill": "self_introduce",
  "args": {}
}
```

- The reply can become speech. The skill is only a proposal.
- The LLM never has direct actuation rights. Navigation and high-risk motions are never opened just because the LLM suggests them.
- Low-risk skills may run after the gate; long motion sequences stay trace-only and are triggered from Studio under human control.

Fig. 6 Safety arbitration path. Status: the deterministic Safety Gate and skill allowlist exist and are covered by unit tests; this does not mean the LLM never hallucinates, and not every safety case is validated end-to-end on the real robot. The hard-stop path still needs an independent audit.

From event to action: six observable steps

Example: the user says **“Introduce yourself.”** This is not one LLM call; it is six steps, each one logged.



Each step returns: accepted · started · blocked · failed · completed

“Having an answer” and “having permission to move” are separate. A spoken reply can play while the motion proposal is still rejected.

Fig. 7 Event-to-action flow for one user request.

Graceful degradation: every cloud path has a local fallback

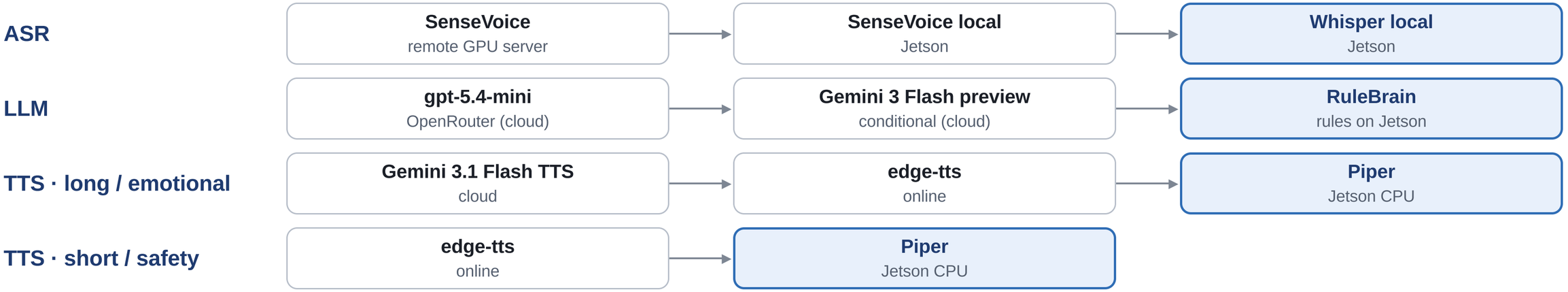


Fig. 8 Fallback chains (left = preferred, right = last resort, highlighted = runs locally or offline-capable). **Table 2** Deployment roles and behaviour when a link is lost.

Location	Main work	Behaviour on disconnection
Jetson Orin Nano	ROS 2 runtime, perception, Brain, Executive, part of local speech	Keeps local perception, rules and the safety path
Operator laptop	PawAI Studio, projection, manual operation	A Studio outage cannot bypass or change the Executive
Cloud model services	High-quality ASR, LLM and TTS	Falls back to an alternative model, a local model, or RuleBrain
Unitree Go2	Sport API, Megaphone, physical motion	Sport motion goes through the Executive; audio and hard-stop have independent paths

Limit. A fully offline voice loop is not yet validated: the Studio laptop microphone path still uses cloud SenseVoice only. Offline-first is a design direction, not a passed baseline.

Observability: PawAI Studio makes each decision traceable

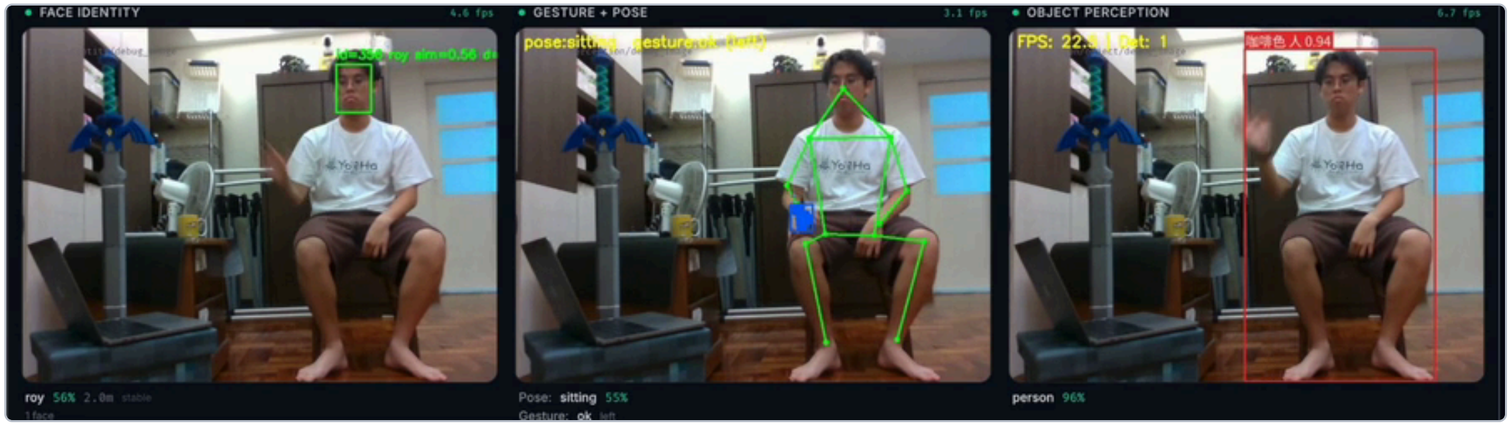


Fig. 9 Live debug panels in PawAI Studio: face identity, gesture + pose, object perception (screen capture).

The problem. In an integrated system the data arrives, a module decides, but nothing moves. From the terminal alone you cannot tell which layer ignored, degraded or blocked the event.

Evidence chain shown side by side in Studio

Perception events
/state/perception/face, /event/object_detected, ...

→

Brain proposals
/brain/chat_candidate, conversation trace

→

Gate results
accepted · trace-only · blocked · not in allowlist

→

Executive decisions
/brain/skill_result lifecycle

→

Browser UI
via Studio Gateway (FastAPI + WebSocket)

Fig. 10 What an operator can trace for every event (redrawn from the PawAI Studio architecture).

- Studio is also the human-in-the-loop console: skill buttons trigger longer sequences under human control.
- Its value is not a nice UI that proves success. It is an evidence chain: why the robot did or did not react, and which layer stopped it.

Capability status: we claim only what the evidence supports

Table 3 Current status of each capability, kept separate from “the code exists”.

Capability	Status	How to read it
Registered-user recognition	Pass (controlled)	One registered user, limited distance and lighting; not access control or general identity verification
Close-range cup detection	Pass (controlled)	About 1 m, cup-only scene; no claim that all 80 classes are stable
Fixed voice commands	Pass (controlled)	Limited commands mapped to system intents; free-form voice control not yet reliable
Brain Safety Gate	Mechanism + unit tests	Blocks unauthorized skills; we cannot claim the Brain never hallucinates
Dynamic wave, voice stop	Not yet passed	Failure records kept; not presented as results
Pose / fall scenarios	Insufficient evidence	Development signals visible in Studio; no care or safety claim
Autonomous navigation, dynamic obstacle avoidance	Experimental	Tested only under human supervision with an emergency-stop procedure

Lesson. A mechanism that exists (Navigation Action, Safety Gate, Studio view) is not the same as a validated capability.

Conclusion and future work

Contributions

1. **Edge multimodal perception**
   Face, gesture, pose, object and speech on one Jetson Orin Nano, joined by shared ROS 2 event contracts.

2. **Deterministic safety arbitration**
   The LLM only proposes; Safety → Policy → Expression gates it, and the Interaction Executive alone releases motion.

3. **Robust and observable operation**
   Graceful degradation for ASR, LLM and TTS, Studio traces, and evidence-based capability reporting.

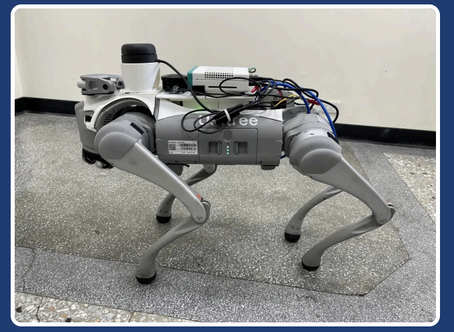
> **Potential.** A practical perception–cognition–action architecture that turns a script-driven consumer quadruped into a context-aware companion, with potential for home and eldercare use.

Future work

- Perception baselines for multiple users, distances and lighting.
- Better voice stop, gesture, and full end-to-end voice tests.
- Link Studio evidence directly to trusted baselines.
- Re-validate navigation under no-motion, e-stop and human-in-the-loop conditions.

Thank you · Questions?

Prof. Wei-Feng Tung · 076144@mail.fju.edu.tw
Dept. of Information Management, Fu Jen Catholic University · Code: github.com/roy4222/PawAI



Sources: paper abstract; PawAI project, roy422.dev