Not Every Call Needs a Frontier Model

Per-Call-Site Evaluation of Small Language Models in a Deployed Agentic Home-Automation System

Panagiotis Kasnesis[1,2,3], Christos Chatzigeorgiou[1,2], Lazaros Toumanidis[1,2], Amalia Contiero Syropoulou[1,3]

[1] University of West Attica, Athens [2] Waldiez PC [3] ThinGenious PC

{pkasnesis, chrihatz, laztoum, acontiero}@uniwa.gr

NEURAL INFORMATION PROCESSING SYSTEMS

Model choice is a per-call-site routing decision, not a single procurement choice

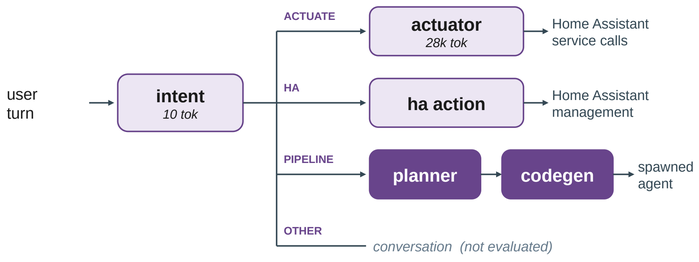
A running agentic framework issues several structurally different LLM calls per user turn, and their difficulty varies by an order of magnitude. Systems usually serve all of them from one model sized for the hardest call, and pay hosted prices for the easiest one, which is also the call made most often. We measure each call site separately in Wactorz, a deployed open-source home-automation runtime, using its unmodified production prompts.

9
models, 0.8B local to frontier hosted

5
call sites of one deployed system

2,520
scored calls, all released

Five call sites in one user turn



Intent is a static gate, so a per-call-site model assignment is a configuration value and cannot mis-route between sites. Actuator carries the whole entity registry (81% of hosted spend). Purple = the two generative sites.

Benchmark, models and protocol

- 280 cases over five call sites (36 intent, 28 HA action, 116 actuator, 50 planner, 50 codegen) grounded in two real Home Assistant installations, incl. cross-site flip, near-miss and refusal cases
- Local: Qwen3.5 0.8B / 2B / 4B and Gemma4 E2B / E4B / 12B / 26B, 4-bit GGUF on one NVIDIA DGX Spark via llama.cpp, no per-call cost. Hosted: Claude Haiku 4.5 and Claude Sonnet 5 (measured API spend)
- Temperature 0, one pass per case; exact label or exact action-set match (an unrequested extra action is a failure); paired McNemar tests on the same cases
- Live run: 46 end-to-end cases on the deployment under three routing configurations, judged by a person

No model wins every call site

Accuracy (%) per call site; best local model per site in purple

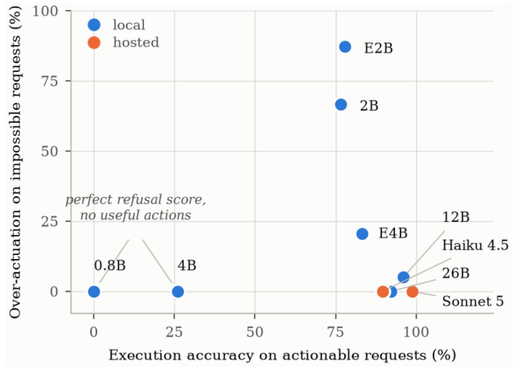
Model	Intent n=36	HA action n=28	Actuator n=116	Planner n=50	Codegen n=50	All n=280	Cost ($)
Qwen3.5 0.8B	50.0	71.4	33.6	8.0	24.0	33.2	0
Qwen3.5 2B	61.1	78.6	62.1	30.0	36.0	53.2	0
Qwen3.5 4B	**91.7**	92.9	50.9	52.0	38.0	58.2	0
Gemma4 E2B	83.3	89.3	56.0	46.0	34.0	57.1	0
Gemma4 E4B	91.7	**96.4**	81.9	50.0	80.0	78.6	0
Gemma4 12B	86.1	89.3	**95.7**	76.0	**84.0**	88.2	0
Gemma4 26B	**91.7**	**96.4**	94.8	**88.0**	80.0	**90.7**	0
Claude Haiku 4.5	80.6	92.9	93.1	82.0	94.0	89.6	3.06
Claude Sonnet 5	94.4	92.9	99.1	84.0	100.0	95.4	12.13

Orderings differ by site and are not monotonic in size. Three different local models take the five sites. Qwen3.5 4B scores below its 2B sibling at actuation (50.9 vs 62.1) and Gemma4 26B below 12B at codegen (80 vs 84).

Only code generation separates local from hosted. Paired McNemar tests: Gemma4 26B is indistinguishable from both hosted models at four of five sites; codegen separates it from Haiku 4.5 ($p = 0.039$) and Sonnet 5 ($p = 0.002$), so the gap belongs to the call site, not to frontier scale.

The gap is narrow. Gemma4 26B matches hosted models on 5 of 12 API shapes and loses almost all ground on callback and lifecycle idioms (window, poll, cleanup, subscribe), the same primitives that failed on real hardware.

Accuracy hides the actuation trade-off



Actuator site: execution accuracy on 77 actionable requests vs. over-actuation on 39 impossible ones (devices only the other installation owns). Deployable is bottom-right; bottom-left is a model that never acts.

Small models fail in opposite directions. Qwen3.5 0.8B never acts (perfect refusal, no useful actions); Gemma4 E2B actuates on 87.2% of requests for devices the house does not own. Constrained decoding would not fix it: 76% of wrong service calls named a real device, so the failure is substitution, not hallucination. Calibration must be measured per model.

Routing beats model choice

Whole-system accuracy and hosted spend over the same 280 cases

Routing configuration	Acc. (%)	$ total	$ / call	Saving
All hosted (Sonnet 5)	95.4	12.13	0.0433	0%
All hosted (Haiku 4.5)	89.6	3.06	0.0109	75%
Hosted for generation only (Sonnet 5)	93.9	2.15	0.0077	82%
Hosted for generation only (Haiku 4.5)	**92.5**	**0.56**	**0.0020**	**95%**
All local, best model per site	91.8	0.00	0.0000	100%
All local, single model (Gemma4 26B)	90.7	0.00	0.0000	100%

81%
of hosted spend goes to the actuator site, which Gemma4 12B already serves at 95.7%

95%
of the spend cut by hosting only planner and codegen (Haiku 4.5), at 92.5% accuracy

$0
per call with all-local per-site routing, 3.6 points below all-hosted Sonnet 5

The live run agrees with the benchmark

46 end-to-end cases on one installation, judged by a person

	all-hosted	gen-hosted	all-local
Direct (actuate, refuse) n=26	25/26	24/26	24/26
Pipeline (scheduled, reactive, spawn) n=20	15/20	15/20	12/20
All cases	40/46	39/46	36/46
Excluding 3 system-limited cases	39/43	39/43	35/43
Hosted spend ($)	2.47	0.70	0.00

- Hosting only the two generative sites matches hosting everything (39/43 vs 39/43) for 28% of the spend
- The offline benchmark predicted the deployed outcome: moving actuation to a local model cost exactly one case in 26
- Scale is not monotone live either: all-hosted was the only configuration to fail “in two minutes turn off the livingroom lights”; both local configurations handled it

Takeaways

- Measure each call site separately; the expensive call is rarely the hard one
- For safety, report the pair (acts when asked, refuses when the device does not exist), not a single accuracy
- Once the expensive calls are separated from the difficult ones, most traffic can run locally at no per-call cost

References

[1] Belcak et al. (2025). Small language models are the future of agentic AI. arXiv:2506.02153

[9] Li et al. (2025). HomeBench: Evaluating LLMs in smart homes with valid and invalid instructions. ACL 2025

[23] Wactorz Contributors (2026). Wactorz: actor-model multi-agent framework, v0.5.3. github.com/waldiez/wactorz

github.com/waldiez/slm-callsite-eval

Research supported by the NVIDIA Academic Grant Program (two NVIDIA DGX Spark systems).

code, benchmark & all records