EXECUTIVE SUMMARY / ENGINEERING DECISION

AI inference in a web-delivered Meta Quest experience

A short, account-free encounter with a locally generated persona is plausible on Quest 3S. The evidence supports building a prototype. It does not yet establish a believable, voiced persona that runs continuously beside a 72 Hz room.

Feasible today

Sonalon already has the most relevant measurement: on its own headset, an unnamed 350M model decoded at 3.8–4.3 tokens/s in single-threaded WebAssembly. A 1,700-token prefill took 266 seconds. The browser exposes a microphone, AudioWorklet, workers and WebGPU. A separate Quest 2 WebXR study demonstrated local speech recognition in WASM. Each component has a testable web path. [S1]

Plausible with engineering work

Start with two candidates. **LFM2.5-350M** is the best speed test: a published Android Chrome Adreno 7xx result shows about 86 prompt tokens/s and 5 generated tokens/s at a prior context of 2,048. **Qwen3-0.6B** is the first alternative for the persona: a Snapdragon 8 Gen 3 phone reached 22.7 generated tokens/s. Three of 12 short responses passed the developer’s checks. A separate prompt variant raised that to six. Neither ran in XR. Liquid itself discourages using its 350M model for creative writing. [A1] [A2] [M1]

Keep one model loaded across thought and action. Reuse the exact persona prefix, aim first for a 100-token thought and a 50-token action, and measure a 200-token situation update. On the LFM phone proxy, that update alone would take about 2.3 seconds and the two outputs about 30 seconds. Generating the longest pair takes about 82 seconds at the same rate, before prefill, speech or rendering. Those are estimates from rounded phone results. [A1]

Outside the demonstrated envelope

At the stated floor of 4 tokens/s, 150–410 generated tokens take **37.5–102.5 seconds**. Even a perfect prefix cache cannot fit the shortest pair into 30 seconds at that speed. I found no public Quest 3, 3S or Pro browser run that establishes the complete workload, no supported Quest browser API for a preinstalled generative model, and no defensible universal tab-memory limit. A visitor’s own provider can support longer reasoning as a separate option.

Scope: stock Quest Browser 154 on the project’s Quest 3S. Research current to 4 November 2026. No access to Sonalon code or headset. The five proposed test pages are on pages 10–11. Bracketed references open the sources.

What has actually run in a browser

Quest 3S, Sonalon’s 4 November brief. Android 14, XR2 Gen 2, Adreno 740. Non-isolated wllama, one thread: 350M, 3.8–4.3 decode and 6.4 prefill tokens/s (1,700 ÷ 266). A 1B model decoded at 1.3. IDs and quants unspecified. llama.cpp WebGPU and ORT WebGPU gave corrupt text. WebLLM loaded 360M/0.8B models, then lost its device on first inference. A correct standalone matvec reached 23–29 GB/s. The run did not characterize heat or concurrent immersive frame timing.

Closest published multi-model browser comparison

Levine et al. ran llama.cpp WebGPU on Android Chrome. These are the rounded numbers printed in the frozen paper’s Figure 8, page 16. The named row is Adreno 7xx; the exact handset, driver and browser build are unspecified. The separately listed Galaxy S24 is an Xclipse device. Five runs per test, 512 prompt or 128 generated tokens, without speech or XR. [A1]

Model and GGUF file	Prefill at depth 0	Prefill at depth 2,048	Decode at depth 0	Decode at depth 2,048
LFM2.5-350M · 0.23 GB	86 tok/s	86 tok/s	6 tok/s	5 tok/s
Gemma 3 270M · 0.25 GB	89 tok/s	68 tok/s	3 tok/s	3 tok/s
Qwen3-0.6B · 0.40 GB	31 tok/s	12 tok/s	3 tok/s	2 tok/s

All three are Q4_K_M. Depth counts prior cached tokens. Blank cells provide no rates; file sizes exclude live allocations. [A1]

Evidence on identified handsets

Galaxy S23, Snapdragon 8 Gen 2. An MLC contributor showed 4-bit Phi-2 generating locally in Android Chrome 121 in a real-time video. No logged prompt length or rates. In a separate test an MLC maintainer read a Vulkan binding range of 134,217,728 bytes on an S23, the same as Sonalon’s headset. [A3] [A4]

Galaxy S24 Ultra, Snapdragon 8 Gen 3. WebLLM Qwen3-0.6B q4f16: 1.208 s to first token, 22.7 tokens/s, 12/12 completions averaging 33 words and 3/12 passing mechanical checks. Prefilling a closed thinking block raised this to 6/12 in a separate run. Qwen3-1.7B emitted corrupt output. Qwen2.5-1.5B managed two readings at 10–13 tokens/s before a mapAsync failure. Exact token counts, thermals and Chrome build were unavailable. [A2]

Pixel 8 Pro, different GPU family. WebLLM Llama 3.2 3B Q4F16_1 processed 148 prompt tokens at 5.38 tokens/s and 47 generated tokens at 5.08. Changing the configured binding requirement or chunk size had little effect in this report. [A5]

No timed independent Quest browser LLM result surfaced. Aether uses an Ollama host. LlamaWeb publishes a browser benchmark. [A9] [A10]

DECISION TABLE / RANKED BY TEST PRIORITY

Runtime and model shortlist

Ranking weighs proximity of the evidence, usefulness for the two-call loop and the cost of learning whether the path works. Confidence concerns transfer to Quest. No row qualifies yet for a 72 Hz production claim.

Priority and combination	Observed rate or Quest estimate	Decision and confidence
1 · wllama WASM SIMD + project's 350M GGUF	Quest 3S, one thread: 3.8–4.3 decode, ~6.4 prefill tok/s. Two- and three-thread rates unknown.	Reproduce with identical weights and enable exact-prefix reuse. Highest confidence in the baseline; low in the size of the speedup.
2 · llama.cpp WebGPU via wllama + LFM2.5-350M Q4_K_M	Adreno 7xx Chrome: 86 prefill / 5 decode tok/s at depth 2,048. Quest rate unmeasured. [A1]	Best GPU speed candidate, particularly for actions. Medium confidence a short loop merits testing; low confidence in prose and XR concurrency.
3 · WebLLM + Qwen3-0.6B q4f16, compiled build	S24 Ultra: 22.7 decode tok/s on short outputs. Different runtime on Adreno 7xx: 12 prefill / 2 decode at depth 2,048. [A1] [A2]	First alternative for prose. Large spread across hardware and runtime; low confidence in a Quest number. Test the chat template with thinking disabled.
4 · wllama + LFM2-700M, then 1.2B Instruct	No matching Quest browser rates. Project's unrelated 1B WASM control: 1.3 decode tok/s. [M2]	Quality challengers for the single-model loop if ranks 1–3 pass timing but fail blind scenes. Low speed confidence.
5 · LiteRT-LM JS + supported Gemma 3 1B artifact; 270M on llama.cpp as control	No Quest LiteRT rate. Gemma 3 270M via llama.cpp on Adreno 7xx: 68 prefill / 3 decode at depth 2,048. [A1]	Independent GPU implementation; confirm exact pack support and total allocation. The 270M result is a separate control. Low confidence.
6 · Transformers.js / ORT WASM and WebGPU + Qwen2.5-0.5B-Instruct	Project reports bad GPU output for an unspecified 0.5B. S24 Ultra app reports 3/3 sanity successes on WASM and 0/3 on GPU with its 0.5B q4f16 setup. [A2]	Use as a CPU/GPU differential test and speech runtime. No defensible Quest language-generation rate. Low confidence.

Retain the fastest configuration that passes the persona and frame tests. Additional candidates, exclusions and weight terms appear on page 7. A quantized model that fits in a download can still exceed peak memory during upload or prefill.

Which runtimes to test

wllama and llama.cpp · MIT. Use the same GGUF, template and tokenizer for CPU and GPU comparisons. wllama 3.1 introduced a unified WASM build with WebGPU. Set the CPU thread count explicitly under cross-origin isolation, call getNumThreads(), and use n_gpu_layers to vary offload. A GPU device and GGUF model buffers must coexist with WASM memory and KV. createChatCompletion supports cache_prompt, streaming and response_format with json_schema. Apply an enum of allowed actions on the second call and check world-state validity in JavaScript. [R1] [R2] [R15]

WebLLM / MLC · Apache-2.0. A precompiled model library, matching quantized weights and its configuration run on WebGPU via TVM. Start with an official small record and record its model ID, library URL, required features and context size. Measure peaks during upload, first prefill and decode separately. Worker engines and schema-constrained JSON are available. Its Snapdragon phone results and first-dispatch failures justify a correctness gate before a speed ranking. [R3] [A2] [G1]

Transformers.js · Apache-2.0. ONNX Runtime · MIT. Version 4 replaced the old WebGPU backend with a C++ execution provider. Retest the exported graph on pinned v4 and WASM, keeping tokenizer and inputs identical. Version 4.1 exposes past_key_values through the pipeline. Avoid large monolithic ArrayBuffers and budget external ONNX weights, intermediate tensors and KV separately. For speech models this runtime can serve both CPU and GPU controls. [R4] [R5] [R6]

MediaPipe and LiteRT-LM · Apache-2.0. The maintenance-only MediaPipe web API has WebGPU Gemma 3 270M and 1B artifacts. Google’s successor is its early-preview LiteRT-LM JavaScript package. It streams models from static URLs or accepts a local Blob, permits a bounded context, offers constrained decoding and can clone a conversation. Pin the selected web artifact. Gated Hub downloads require a visitor token. For account-free entry, host a redistributable copy under its weight terms. No Quest speed measurement surfaced for either engine. [R7] [R8] [R14]

An Android runtime comparison. A 22 September LocalMode run on Chrome 153 and Adreno 830 used SmolLM2-135M and 128-token prompt/output targets. WebLLM q0f16 achieved 0.34 s to first token and 238 characters/s. With the same Q4_K_M GGUF, wllama WASM measured 1.63 s and 171 characters/s; wllama WebGPU measured 0.61 s and 65 characters/s. The phone was charging, outside XR, with a binding range near 2 GiB. Its character rates do not directly compare with token rates. [A13]

Built-in AI and WebNN. Google excludes Android from Chrome’s Gemini Nano Prompt API. Probe globalThis.LanguageModel and availability() on Quest. For WebNN, probe navigator.ml, create a context and execute a small known-answer graph. Chromium’s Android paths include XNNPACK CPU and WebGPU/Dawn GPU with fallbacks. Its implementation table lists no NPU accelerator. I found no Meta commitment to expose an equivalent on-device language model or NPU web API. [R9] [R10]

CORRECTNESS AND COMPATIBILITY

What the GPU failures tell us

The tested Quest reports shader-f16 and subgroups. A correct matvec at 23–29 GB/s shows that one kernel works. A full model also exercises unpacking, gathers, attention, KV writes and repeated dispatches. WebLLM 0.2.84 users reported device loss during warm-up on Adreno 6xx and 810 even though small compute tests passed. Both q4f16 and q4f32 failed. This matches Sonalon's symptom without establishing its cause. [G1]

Two failures with verified, specific mitigations

An older Gemma 3 270M ONNX export overflowed the fp16 residual stream around layers 7–8 and repeated unused tokens. Re-export cfd5c04f84 clipped residuals and passed the reporter's check. Separately, LiteRT.js on Galaxy Z Fold7, Adreno 830, Chrome 149 returned correct YOLOv5n results on first execution and corrupt results on subsequent executions. The reporter traced stale reads to reuse of a tensor texture and found that a fresh allocation or re-uploading the affected texture restored correct results in its reproducer. Neither result establishes a fix for Quest's Adreno 740. [G2] [G3]

What 128 MiB constrains

A Samsung S23 also exposed a Vulkan storage-buffer range of 134,217,728 bytes. Sonalon's 2 GiB maxBufferSize is a separate single-allocation limit. Each bound range must fit the 128 MiB limit at an aligned offset. Fetching a tensor in small files achieves nothing if the runtime then binds the full tensor. Check embeddings, output projection, scratch and KV allocations. Genuine row tiling or sharding requires kernel support. MLC previously shipped three explicitly compatible 1k-context builds. [A4] [G5] [G6]

On identical token IDs and a pinned export, compare CPU and GPU logits over three repeated runs. Capture the first divergent operator, nonfinite values, adapter and granted device limits, compilation messages, validation and out-of-memory error scopes, and device.lost. Vary f32 arithmetic, subgroup-free kernels, smaller workgroups and prefill chunks one at a time. These are diagnostic variants, each with a memory or throughput cost. [G6]

Quest Browser release history

Meta documented experimental WebGPU rendering in WebXR in 146.0 (21 April), foveation in 150.1 (28 August), and composition layers in 152.1 (2 October) behind "webxr experiments" in chrome://flags. The 154.1 notes (30 October) list a stereo auto-detection experiment and generic fixes. Version 154.0 (14 October) added experimental hit testing and body tracking. None identifies an inference regression or fix. Sonalon can test WebGPU compute alongside its existing WebGL XR renderer on visitors' default settings. [G7] [G8]

Two version probes are worth keeping. A WebLLM maintainer suggested comparing 0.2.79 and 0.2.80+ after a GPU-kernel change. A native Dawn report on S24 Adreno 750 compiled the same graph with v0.14.0 but crashed with v0.17.x. These are diagnostic leads; neither identifies a corrective Quest Browser version. [G1] [A11]

Making the persona prefix cheap

A cached model download reduces startup network traffic. A live inference cache stores the model's state after specific tokens. The latter can remove almost all repeated work on the 1,500–3,000-token prefix. Keep the exact token sequence unchanged, including role markers, whitespace and beginning-of-sequence tokens. Move clocks, retrieved memories and changing scene data behind it. Preserve the model's own chat template. [R2] [R11] [R13]

What still costs time

Sonalon's CPU baseline processed about 6.4 prompt tokens/s. Even perfect prefix reuse would leave **31 seconds** for 200 new tokens at that rate. To finish 200 or 400 new tokens in five seconds requires 40 or 80 prompt tokens/s, respectively. On the LFM Adreno proxy at depth 2,048, 200 new tokens / 86 prompt tokens/s gives **2.3 seconds**. A cold 1,700-token prefill at 86 tokens/s would take about 20 seconds. These extrapolations assume linear throughput. Measure both calls and any fresh action instruction. With 5 seconds reserved for everything else, four decode tokens/s buys 100 output tokens in a 30-second step or 220 in 60 seconds. [A1]

Where to keep the state

In wllama, retain the engine and compare cache_prompt true/false. Append the action request immediately after the actual thought as a normal next turn. Check the cached-token count before trusting fast timings, especially when replacing a suffix on a hybrid model. In LiteRT-LM, set prefillPrefaceOnInit, keep that base conversation and clone() it for a new situation. Measure clone latency and peak memory. Transformers.js 4.1 exposes past_key_values. In WebLLM, ordinary chat reuse requires the preceding history to match. Raw completions and some interrupted or length-limited turns reset it. PR 664 for arbitrary system prefixes remained open. [R2] [R5] [R8] [R11] [R12]

Persistence and alternatives. Native llama-server supports saving/restoring slot state. The wllama public class does not expose those file endpoints. A custom binding should invalidate snapshots when the weights, tokenizer, template, cache format or runtime changes. Chunked prefill can improve scheduling and peak workspace while still evaluating each new token. Speculative decoding can accelerate accepted output tokens, but a second model adds memory and verification work. No Quest speedup was established for either technique. [R2] [R13]

Phone evidence. Google estimates native Pixel 9 Gemini Nano latency at 2.11 s without prefix caching and 0.50 s on a hit for 1,000 fixed + 100 dynamic tokens. An independent native Android test of Qwen2.5-0.5B Q4_K_M used 4 CPU threads, a 123-token prefix and a 10-token suffix. Mean first-token time across 40 requests fell from 4.828 to 0.486 s with state reuse, including 0.123 s for restore. No equivalent 3,000-token Quest browser comparison surfaced. [A8] [A14]

MODEL SELECTION / DIALOGUE AND ACTIONS

Which models might sound like a persona

The strongest reason to try LFM2-700M for prose is a dialogue comparison. The strongest reason to try LFM2.5-350M first is speed. Those findings come from different checkpoints and answer different questions. [A1] [M1] [M2]

LFM2-350M, 700M and 1.2B. Liquid used five model judges and 1,000 real WildChat conversations. Its 350M roughly tied Qwen3-0.6B, its 700M was preferred over Qwen3-0.6B, and its 1.2B was comparable to Qwen3-1.7B and preferred over Gemma 3 1B IT and Llama 3.2 1B Instruct. These are vendor-run conversational preferences. They do not score first-person continuity in a simulated home. LFM2.5-350M improves tool use, but Liquid discourages creative writing with it. [M1] [M2]

Qwen3-0.6B and 1.7B. Qwen claims role-play and creative-writing ability; the S24 Ultra 0.6B run gives concrete evidence of occasional unusable prose or format. Use the template's hard non-thinking mode where the export supports it and inspect the resulting tokens. The S24 developer suppressed reasoning with a raw completion that reset the cache. Reproduce both the quality gain and the cost of rebuilding Sonalon's longer prefix. Include Qwen3.5-0.8B as an alternate only after its hybrid architecture and text-only export pass correctness. [M3] [M4] [A2] [R11]

Qwen3.5-2B and Llama 3.2 1B. TRACE Bench tested 200 multi-turn role scenarios. Qwen3.5-2B scored 67.22 overall and 63.50 on cross-turn memory. EQ-Bench 3's Llama 3.2 1B run scored 12/100 on its emotional-intelligence rubric. In its lunchroom scenario it began in first person, then repeated the formatting instructions. Both evaluations publish the task or transcript; neither tests Sonalon's shorter private thoughts at Q4. [M5] [M6]

SmolLM2-360M, Gemma 3 270M and Gemma 3 1B. SmolLM2's card reports MT-Bench 3.66 against 4.16 for Qwen2.5-0.5B. This supports a compact dialogue baseline. Google positions the 270M model for task specialization. FunctionGemma-270M's Mobile Actions score rose from 58% to 85% after tuning on that task. It is a useful action-only comparison if a single resident model fails. [M7] [M8] [M9]

Newer and larger options. Liquid released LFM3-1.7B on 27 October 2026 with native llama.cpp support; I found no matching Quest browser benchmark. Gemma 3n E2B has 5B physical parameters and a claimed optimized floor of 2 GB. Gemma 4 E2B in the browser paper was a 3.11 GB Q4_K_M file. SmolLM3 is 3B; Phi-4-mini is 3.8B. Spend the initial Quest budget on the smaller candidates. [M10] [M11] [A1] [M12] [M13]

What a visitor can say and hear

On Quest 2, Berglin ran Silero speech recognition locally in the browser with ONNX Runtime WASM. Mean processing time was about 3.5 seconds per command across 120 recordings, each replayed ten times. The study did not report a real-time factor. [S1]

Sonalon’s browser exposes neither SpeechRecognition nor speechSynthesis. We found no timed demonstration of faster-than-real-time open-vocabulary recognition or synthesis in stock Quest 3, 3S or Pro Browser, in a flat tab or during immersive WebXR. The phone measurements below identify candidates. Real-time factor (RTF) is wall time divided by audio duration. A value of 0.5 processes ten seconds of audio in five seconds.

Device and run	Recognition	Synthesis
Galaxy SM-S918B, Snapdragon 8 Gen 2, native ARM64 CPU [S2]	Parakeet-EOU-120M ONNX int8, 0.21 RTF, 232 MB peak resident memory. Input was a 20-second clip.	Kokoro-82M ONNX fp32, 0.53 RTF, 640 MB peak resident memory.
Galaxy S10, Exynos 9820, Android 12, native sherpa-onnx [S3] [S4]	One 30-second English clip. Moonshine Tiny 27M, 0.05 RTF. Whisper Tiny 39M, 0.07. Streaming Zipformer 20M, 0.12.	Four threads, median RTF. Piper amy-low 0.076, Kitten Nano v0.2 fp16 0.387, Kokoro EN v0.19 1.133.

Browser candidates for the first voice test

For English, start with MoonshineJS Tiny (MIT code and English weights). Its microphone API can commit a transcript after a pause. Compare it with sherpa-onnx’s streaming English Zipformer WASM SIMD demo for partial text while the visitor speaks. Use Whisper Tiny as an accuracy control and pin its backend and quantization. The S10’s second Whisper Tiny configuration took 105.6 seconds for the same clip. [S3] [S5] [S6]

A separate Galaxy S24 Chrome report describes corrupted WebGPU Kokoro audio and slow but intelligible WASM output; it supplies no RTF or GPU SKU. [A12] For her voice, try Piper-low with eSpeak NG phonemization and WASM synthesis, then Kitten Nano int8 (15M, 25 MB download). Check each Piper voice’s terms. Test Kokoro once the room and mind are resident. Time from the end of the visitor’s speech to first audible reply matters even when average RTF is good. [S2] [S4] [S7] [S8]

Capture audio in an AudioWorklet, resample to 16 kHz mono, and keep recognition off the audio thread. Compare push-to-talk with Silero VAD. Replay the persona’s voice through the headset speakers while someone interrupts her. Count false triggers, missed interruptions and errors on names. [S9]

Keeping the room responsive

At 72 Hz, each stereo frame has 13.89 ms. Meta’s Project Flowerbed team met its WebXR budget by reducing WebGL calls, shader work, overdraw and costly texture resolves. Inference shares the GPU and memory bandwidth with that render. [S10]

How much can fit

The 8 GB Quest 3S reported 2.3–2.9 GB free, about three logical cores and a roughly 2 GB JS heap ceiling. We found no documented Quest 3S tab-kill threshold. Model downloads, WASM memory, CPU staging copies, GPU buffers, XR render targets, textures, KV caches and speech workspaces have different lifetimes. Measure peak residency during upload and speech startup. Include the 23 MB embedding model’s working memory.

A Quest 2 developer reproduced an out-of-memory crash by switching scenes and traced most growth to recreating the XR layer. The JS profiler missed graphics allocations. In the lab, sample Android proportional set size for the renderer and GPU processes, system memory available, and graphics allocations where exposed. Keep the XR layer alive across room changes and release unused model sessions and upload buffers. [S11]

Sustained rendering and inference

Probe navigator.gpu in a dedicated worker and create the compute device there. Keep WebGL and XRSession.requestAnimationFrame on the main thread. Workers can free CPU time on that thread. Long GPU submissions can still delay eye frames. Bound prefill chunks, keep few submissions in flight and compare serialized speech and thought against overlap. [G6]

We found no quantified Quest 3S browser LLM throttling curve. A Quest Pro Browser 149.1 WebGL2 report shows 54 then 46 fresh frames/s at 72 Hz, with GPU clock steady at 525 MHz and reported temperature steady at 23 °C. FPS alone cannot identify thermal throttling. Compare the first and last five minutes on battery, matching initial heat and brightness. Record tokens/s, stale frames, clocks and thermal status. [S12] Meta’s OVR Metrics Tool can supply device-side counters during lab runs. [S13]

Launching again

The v74 browser shortcut opens a website in its own window. Meta also documents a separately packaged store PWA. The public URL already meets Sonalon’s delivery constraint. Cache versioned model files with a service worker and checkpoint conversation state when the page hides or the XR session ends. Plan to rebuild GPU/KV state on resume. We found no documented extra RAM, continuously running background worker or browser-callable Meta model granted by installation. [G6] [S14] [S15]

Establish correctness and turn latency

Serve static HTTPS pages with COOP: same-origin and COEP: require-corp. Self-host assets or configure CORS/CORP. Log crossOriginIsolated, real thread count and OS/browser build. Pin runtime, model, quant, tokenizer and template revisions. Export per-run JSON. [A7]

1 /cpu-prefix · Can a warm CPU session meet the budget?

In a wllama worker with zero GPU layers, test the original 350M GGUF then LFM2.5-350M at 1, 2 and 3 threads. Use 1,700- and 3,000-token personas plus 64, 200 or 300 fresh tokens. Generate a 100-token thought, append the action instruction, then generate 50 action tokens. Repeat five times, then change an early prefix token. Log actual output length. [A1]

Expected result. The one-thread cold control should approach the original timing. Changing an early token should erase most of a cache hit. Select the fewest threads with ≤5 s warm prefill at 200 fresh tokens and ≥4 decode tokens/s. If all fail, record the largest suffix that passes.

2 /gpu-parity · Which backend produces trustworthy tokens?

First run upstream wllama WebGPU LFM2.5-350M Q4_K_M, then WebLLM Qwen3-0.6B via its official worker, then ORT’s 0.5B export with explicit WASM and WebGPU providers. Use identical input IDs and forced continuations on the CPU reference for each artifact or an equivalent verified implementation. Test lengths 16, 257 and 1,700, repeated three times per loaded session. Record top logits and ranks where exposed, NaNs, first major divergence, device loss and error scopes. A near-tied top token may change with rounding.

Expected result. Look for repeatable, sensible answers and similar high-probability tokens. Isolate precision, subgroup use, context and batch on failures. Run PP512/TG128 at cache depths 0 and 2,048. A correct LFM result near 5 decode tokens/s warrants the XR test. [A1] [G1] [G2]

3 /persona · Can the same model think and act in character?

Put the leading two timed models and one larger quality comparator through 12 fixed situations × 2 seeds. Include a renamed visitor, a moved object, a long silence and an impossible request. Save both raw outputs. Blind two readers to model identity; score voice, continuity and room grounding from 1–5. Separately validate schema, allowed action, real entity IDs and consistency with the thought. Show representative failures alongside successes.

Proposed pass. Expect more failures in voice or intent than in grammar-constrained syntax. Target median ≥4/5 on each prose dimension, at least 23/24 usable actions, and zero actions executed against missing entities. A grammar can fix malformed JSON. A perfectly formatted action directed at the wrong object still fails.

HEADSET EXPERIMENTS / 4 AND 5

Verify conversation inside the room

4 /voice · Can the visitor interrupt and get a timely reply?

Capture mic PCM in an AudioWorklet. Send 16 kHz mono frames to a recognition worker. Compare Moonshine Tiny with streaming sherpa-onnx Zipformer, using Whisper Tiny as an accuracy control. Pair the fastest accurate recognizer with one Piper-low voice and then Kitten Nano int8. Try Kokoro only after recording spare memory with the mind loaded. Test VAD and push-to-talk on identical audio. [S5] [S6] [S7] [S8] [S9]

Replay ten utterances at each of 3, 6 and 12 seconds, with silence, background noise, names and interrupted speech. Then repeat live using the headset speakers and requested echo cancellation. For ASR, report time spent decoding and time from end of speech to a final transcript. For TTS, time first PCM, first audible playback and completion. On unpaced replay, divide ASR processing time by input duration and synthesis time by generated audio duration. Exclude recording and playback time. Count word errors, missed names, self-triggers and underruns.

Expected result. On native phones small recognizers and Piper have enough speed to justify a browser trial. A reasonable initial gate is warm RTF ≤0.7 for both speech stages, median end-of-speech to transcript ≤1 s, and intelligible uninterrupted playback. Score streaming endpoint latency separately. [S2] [S3] [S4]

5 /room-soak · Can it hold 72 Hz after the headset warms?

Use the representative textured stereo room, actual avatar and a 72 Hz immersive-vr session with WebGL2 multiview. Log layer and resolution settings. Run the same scripted movements and visitor utterances in three conditions: room alone, one worker at a time for speech/LM/embedding, and controlled overlap. Use a 1,700- and then 3,000-token persona with 200 changing tokens. Exercise 100+50 and 250+160 generated tokens; report each separately. Keep the 23 MB embedding model resident or log its load time on demand.

Run each condition for 30 minutes after a comparable start, with two repeats in mixed order. Record p50/p95 prefill, decode, full step and voice response, compositor misses, p95/p99 page-frame spacing, peak memory and reloads. Log clocks where exposed, charge state, brightness and room temperature. Re-enter VR and suspend/resume once. Browser-only users need no lab connection. [S10] [S11] [S13]

Proposed pass. Sustain 72 Hz, keep stale/dropped frames below 1% and within one percentage point of the room-only baseline, and keep the short complete step’s p95 ≤60 s, with no corruption, audio underruns or reloads. Report the first and last five minutes separately. If only serialized work passes, ship the tested scheduling policy for the pilot. If prose or timing fails, keep that combination out of the no-key default.

The five-second prefill, voice, frame and quality cutoffs are proposed acceptance criteria. The project supplied the target cadence and four-token/s floor.

Open risks and weight terms

How much memory does a longer persona add?

For Qwen3-0.6B, 28 layers × 8 KV heads × 128 dimensions × key and value × 2 bytes gives **112 KiB per token** in a conventional fp16 KV cache. That is 224 MiB at 2,048 tokens and 448 MiB at 4,096, before weights, scratch and XR. An implementation using other cache types or layouts will differ. Two separate model contexts can pay twice. A 3,000-token persona plus 300 changing tokens and a 410-token output pair already approaches a 4,096-token configuration after chat formatting. Measure the actual formatted length and reserve space for the action. [A6]

What would still prevent release?

Cold arrival. Even a successful warm loop can leave a first-time visitor waiting for downloads, compile and initial prefill. Record first download bytes, first usable frame, first local speech, first fresh thought and second-visit times. If prefill is warmed while the room opens, include that work in the first-visit clock. Test an interrupted download, evicted cache and a headset resume.

Semantic and conversational latency. A model may maintain the background cadence but make a visitor wait through a full private thought before acknowledging them. Time each step to the first audible reply. Keep the visitor's own model provider explicitly selectable, and keep account-free qualification tied to the fully local test.

Headroom and upgrades. An OS update, different driver or downloaded graph revision can change correctness or peak memory. Keep a fixed short golden transcript plus a synthetic 3,000-token case. Re-run both and the immersive soak before changing the default. Local speech tests here assume English; other languages and voices need their own results.

Weight licences for the evaluated families

Weights	Distribution terms to track separately from Sonalon's GPL-3.0 code
Qwen3 / Qwen3.5, SmolLM2 / SmolLM3	Apache 2.0 for the listed checkpoints. Preserve notices in each actual quantized export. [M3] [M4] [M7] [M12]
LFM2, LFM2.5 and LFM3	LFM Open License v1.0. Its commercial-use threshold is defined as US$10M or more in annual revenue. [M14] [M10]
Gemma 3, 3n, FunctionGemma; Gemma 4	The first three use Google's Gemma Terms, with redistribution conditions and prohibited-use terms. Gemma 4 E2B uses Apache 2.0. [M15] [M16]
Llama 3.2 1B; Phi-4-mini	Meta Llama 3.2 Community License and acceptable-use policy for Llama. MIT for Microsoft's Phi-4-mini. [M17] [M13]

For the voice stack, licences may attach separately to engine, phonemizer, voice/model files and the particular distribution. Record the exact assets shipped as the smoke tests become a public build.

BIBLIOGRAPHY / DEVICE EVIDENCE

Headset and phone comparisons

The project's Quest 3S measurements come from the Sonalon research brief dated 4 November 2026. Entries below are primary measurements, demonstrations or product source. Developer observations are identified in the text. Source dates refer to publication, update or the cited run. Undated and living documents carry the date checked.

A1 Levine et al. · Llamas on the Web, frozen benchmark paper · 20 May 2026 · v1, §4, Table 3 and Fig. 8
arxiv.org/pdf/2605.20706v1#page=16

A2 Musa V. Ahmed · Major ArcanAI Android evaluation on the Galaxy S24 Ultra · 11 Aug 2026 · pinned developer notes
github.com/MusaVAhmed/major-arcanai/blob/9ca9c422e4d3a0edfa080a984437012f204474fa/CLAUDE.md#L623-L845

A3 MLC contributor · Phi-2 in Android Chrome on a Galaxy S23 · 24 Jan 2024 · firsthand demonstration
www.reddit.com/r/LocalLLaMA/comments/19envw6/

A4 MLC · WebLLM issue 209, Android storage-buffer limits · Nov 2023–May 2024 · S23 reading 17 Jan 2024
github.com/mlc-ai/web-llm/issues/209

A5 MLC · WebLLM issue 759, Pixel 8 Pro prompt and decode measurements · 3 Jan 2026 · developer measurements
github.com/mlc-ai/web-llm/issues/759

A9 Ibrews · Aether Companion, WebXR client and Ollama host architecture · repository checked 4 Nov 2026
github.com/ibrews/aether-companion

A10 Levine et al. · Browser benchmark, downloadable GGUFs and runnable demo · 20 May 2026 · study companion
reeselevine.github.io/llamas-on-the-web/

A13 LocalMode Bench · Android Chrome 153, Adreno 830, three SmolLM2-135M paths · 22 Sep 2026 · raw run
github.com/LocalMode-AI/LocalMode-Bench/blob/main/runs/2026/09/0905c473-e5a7-4ad5-9b81-e3c4b8280c43.json

R1 Nguyen Xuan Son · wllama 3.1.1, WebGPU and unified WASM build · 13 May 2026
github.com/ngxson/wllama/releases/tag/3.1.1

R2 wllama · Wllama class, chat response formats and cache_prompt · API checked 4 Nov 2026
github.ngxson.com/wllama/docs/classes/Wllama.html

R3 MLC · WebLLM, compiled model libraries, JSON constraints and workers · repository checked 4 Nov 2026
github.com/mlc-ai/web-llm

R15 wllama · MIT licence, shared memory, workers and GPU layer offload · repository checked 4 Nov 2026
github.com/ngxson/wllama

BIBLIOGRAPHY / RUNTIME DOCUMENTATION

Web engines and prompt caching

R4 Hugging Face · Transformers.js v4 and C++ WebGPU execution provider · 9 Feb 2026 · updated for v4
huggingface.co/blog/transformersjs-v4

R5 Hugging Face · Transformers.js 4.1.0, cached generation API · 23 Apr 2026
github.com/huggingface/transformers.js/releases/tag/4.1.0

R6 Microsoft · ONNX Runtime Web, working with large models · guide checked 4 Nov 2026
onnxruntime.ai/docs/tutorials/web/large-models.html

R7 Google AI Edge · Maintenance status and MediaPipe web LLM guide · 12 Jun 2026
developers.google.com/edge/mediapipe/solutions/genai/llm_inference/web_js

R8 Google AI Edge · LiteRT-LM JavaScript, conversation branches and model delivery · 30 Oct 2026
developers.google.com/edge/litert-lm/js

R9 Chrome Developers · Prompt API and hardware requirements · published 20 May 2025 · header revised 21 Oct 2026
developer.chrome.com/docs/ai/prompt-api

R10 WebNN · Browser compatibility and device selection in Chromium · 9 Oct 2026
webnn.io/en/api-reference/browser-compatibility/api

R11 MLC · engine.ts, conversation matching in preparePrefillInput · source checked 4 Nov 2026
github.com/mlc-ai/web-llm/blob/main/src/engine.ts

R12 MLC · PR 664, proposal for reusable system prefixes · 4 Feb 2025 · open on 4 Nov 2026
github.com/mlc-ai/web-llm/pull/664

R13 ggml · llama.cpp server, cache settings, slots and speculative methods · reference checked 4 Nov 2026
github.com/ggml-org/llama.cpp/blob/master/tools/server/README.md

R14 Google AI Edge · LiteRT-LM licence and JavaScript preview status · repository checked 4 Nov 2026
github.com/google-ai-edge/LiteRT-LM

A6 Qwen · Qwen3-0.6B architecture and KV dimensions · configuration checked 4 Nov 2026
huggingface.co/Qwen/Qwen3-0.6B/blob/16706fc57485378d4ffaf54139b29ccc66ae08fa/config.json

A7 Chrome Developers · Enabling cross-origin isolation · living guide · checked 4 Nov 2026
web.dev/articles/coop-coep

A8 Google ML Kit · Pixel 9 prefix-caching estimates for native Gemini Nano · 27 Oct 2026
developers.google.com/ml-kit/genai/prompt/android/prefix-caching

A14 EdgeSync author · Measured TTFT on Android using llama.cpp state restore, with correctness controls · 11 Jul 2026
dev.to/bossandboss/i-got-99x-lower-ttft-on-a-real-android-phone-by-reusing-llamacpp-kv-state-1ngi

GPU and headset references

G1 WebLLM issue 836. Adreno device loss during engine warm-up · 21 Jun 2026
github.com/mlc-ai/web-llm/issues/836

G2 ONNX Runtime issue 26732. Gemma 3 fp16 export and residual overflow · 5 Dec 2025, follow-up Jul 2026
github.com/microsoft/onnxruntime/issues/26732

G3 LiteRT issue 8065. Repeated-run corruption and texture-reuse workaround on Adreno 830 · 12 Jun 2026, follow-up Jun 2026
github.com/google-ai-edge/LiteRT/issues/8065

G5 MLC WebLLM PR 213. Three 128 MiB-compatible model builds · 12 Nov 2023
github.com/mlc-ai/web-llm/pull/213

G6 W3C. WebGPU Candidate Recommendation Draft, limits, buffer bindings and errors · 15 Sep 2026
www.w3.org/TR/2026/CRD-webgpu-20260915/

G7 Meta. Quest Browser release notes by version · 21 Apr to 30 Oct 2026
developers.meta.com/vr/release-notes/?search_key=browser

G8 Meta. Quest Browser 152.1 and the WebXR experiments flag · 2 Oct 2026
developers.meta.com/vr/downloads/package/browser/152.1/?view=full_width

A11 Google AI Edge · LiteRT-LM issue 3867, Adreno 750 native Dawn compiler comparison · Oct 2026 issue · viewed 4 Nov 2026
github.com/google-ai-edge/LiteRT-LM/issues/3867

A12 Hugging Face · Transformers.js issue 1320, Kokoro on Galaxy S24 Chrome · May 2025 · developer report
github.com/huggingface/transformers.js/issues/1320

S10 Meta, Project Flowerbed WebXR case study and frame budget · 23 Feb 2023
developers.meta.com/vr/blog/project-flowerbed-a-webxr-case-study/

S11 Alberto Gnoato, Babylon.js forum, Quest 2 XR scene-switch memory investigation · 6–7 Feb 2024
forum.babylonjs.com/t/switching-scenes-in-a-webxr-session-makes-quest-2-go-out-of-memory/47868

S12 Meta Community Forums, Quest Pro Browser 149.1 frame production report and device logs (thread 1380174) · 9 Aug 2026 captures
communityforums.atmeta.com/discussions/Questions_Discussions/quest-pro-browser-webxr-session-frame-production-permanently-drops-13-after-20%E2%80%935/1380174

S13 Meta, VR App Testing Guide, automation and performance · n.d., checked 4 Nov 2026
developers.meta.com/vr/resources/automation-performance-testing/

S14 David Heaney / UploadVR, Horizon OS v74 browser web shortcuts · 18 Feb 2025
www.uploadvr.com/meta-quest-v74-update/

S15 Meta, quarterly developer recap, WebXR PWAs and store packaging · 27 Mar 2025
developers.meta.com/vr/blog/quarterly-developer-recap-tracked-keyboard-hand-gestures-webxr-spatial-sdk/

BIBLIOGRAPHY / MODEL EVIDENCE

Dialogue evaluations and model cards

M1 Liquid AI · LFM2.5-350M results and intended tasks · 31 Mar 2026
www.liquid.ai/blog/lfm2-5-350m-no-size-left-behind

M2 Liquid AI · LFM2 launch and 1,000-dialogue WildChat preference test · 10 Jul 2025
www.liquid.ai/blog/liquid-foundation-models-v2-our-second-series-of-generative-ai-models

M3 Qwen · Qwen3-0.6B card, thinking control and licence · Apr 2025 · checked 4 Nov 2026
huggingface.co/Qwen/Qwen3-0.6B

M4 Qwen · Qwen3.5-0.8B model card and comparison with the 2B model · Mar 2026 · checked 4 Nov 2026
huggingface.co/Qwen/Qwen3.5-0.8B

M5 Kuaishou GameMind · TRACE Bench, role-play protocol and model results · 2026 · checked 4 Nov 2026
kuaishou-gamemind.github.io/projects/trace_bench/paper/main.pdf

M6 EQ-Bench 3 · Meta Llama 3.2 1B Instruct, rubric and scenario transcripts · run undated · checked 4 Nov 2026
eqbench.com/results/eqbench3_reports/meta-llama__llama-3-2-1b-instruct.html

M7 Hugging Face · SmolLM2-360M-Instruct card and MT-Bench results · 2024 release · checked 4 Nov 2026
huggingface.co/HuggingFaceTB/SmolLM2-360M-Instruct

M8 Google · Gemma 3 270M design and specialization · 14 Aug 2025
developers.googleblog.com/en/introducing-gemma-3-270m/

M9 Google · FunctionGemma 270M IT, Mobile Actions evaluation · Dec 2025 · checked 4 Nov 2026
huggingface.co/google/functiongemma-270m-it

M10 Liquid AI · LFM3 release, architecture and native runtimes · 27 Oct 2026
www.liquid.ai/blog/lfm3

M11 Google · Gemma 3n developer guide, physical and effective parameter counts · 26 Jun 2025
developers.googleblog.com/en/introducing-gemma-3n-developer-guide/

M12 Hugging Face · SmolLM3-3B model card and licence · Jul 2025 · checked 4 Nov 2026
huggingface.co/HuggingFaceTB/SmolLM3-3B

M13 Microsoft · Phi-4-mini-instruct model card and licence · Feb 2025 · checked 4 Nov 2026
huggingface.co/microsoft/Phi-4-mini-instruct

Weight terms and browser speech

M14 Liquid AI · LFM Open License, definitions and commercial terms · licence text · checked 4 Nov 2026
www.liquid.ai/lfm-license

M15 Google · Gemma Terms of Use and applicable models · modified 1 Apr 2026
ai.google.dev/gemma/terms

M16 Google · Gemma 4 E2B IT model card and licence · Apr 2026 · checked 4 Nov 2026
huggingface.co/google/gemma-4-E2B-it

M17 Meta · Llama 3.2 1B Instruct model card and Community License · 25 Sep 2024 · checked 4 Nov 2026
huggingface.co/meta-llama/Llama-3.2-1B-Instruct

S1 Elias Berglin, WebXR Voice Assistant, figures 13–16 · 20 Jun 2022
www.diva-portal.org/smash/get/diva2%3A1698758/FULLTEXT01.pdf

S2 Soniqo, Android on-device benchmarks, Galaxy SM-S918B · n.d., checked 4 Nov 2026
cloud.soniqo.audio/benchmarks

S3 Akinori Nakajima / VoicePing, offline speech-to-text benchmark and app source · updated 21 Sep 2026
voiceping.net/en/blog/research-offline-speech-transcription-benchmark/

S4 Akinori Nakajima / VoicePing, offline text-to-speech benchmark and app source · 15 Feb 2026
voiceping.net/en/blog/research-offline-tts-eval/

S5 Moonshine AI, MoonshineJS browser API and English-model licence · n.d., checked 4 Nov 2026
dev.moonshine.ai/moonshine-js/

S6 k2-fsa, sherpa-onnx browser Zipformer demos and WASM SIMD build workflow · n.d., checked 4 Nov 2026
k2-fsa.github.io/sherpa/onnx/wasm/hf-spaces.html

S7 Piper browser port, eSpeak NG and ONNX Runtime Web implementation · n.d., checked 4 Nov 2026
github.com/piper-tts/piper-tts.github.io

S8 KittenML, KittenTTS Web model sizes and browser implementation · n.d., checked 4 Nov 2026
github.com/KittenML/KittenTTS-web

S9 Ricky Samore, browser voice activity detection and self-hosting guide · n.d., checked 4 Nov 2026
github.com/ricky0123/vad/blob/master/docs/user-guide/browser.md