Case Study

Internal Production Pilot — "HumanityAI'd Office" Museum Instance

HumanityAI'd — July 2026

CONFIDENTIAL

Summary

To prove the Museum AR Smart Glass Guide end-to-end before customer pilots, HumanityAI'd stood up a real, running museum instance on its own production infrastructure: **10 exhibits, 12 enrolled photos, 8 languages, 80 professionally narrated audio guides**, all served by the exact stack a museum would deploy — on a single shared GPU. This case study documents what was built, the measured results, and the engineering issues surfaced and fixed along the way.

The setup

- **Venue**: the HumanityAI'd office, catalogued as a museum instance with 10 "exhibits" — a mix of equipment, furniture, motivational artwork, and a brand landmark (a deliberately varied set to stress-test recognition on non-artwork objects).
- **Content**: each exhibit given a professional title, short and full description, a fun fact, and a narration script — authored (not machine-stubbed) in **English, Arabic, French, Spanish, Chinese, Russian, Hindi, and German**.
- **Infrastructure**: the production stack (FastAPI + CLIP ViT-L/14 + FAISS + Chatterbox multilingual TTS + PostgreSQL/Redis/nginx) on a single **NVIDIA RTX 4060 Ti (8 GB)** — sharing that GPU with two other live AI workloads, i.e. a deliberately constrained, realistic environment.

What was measured

Metric	Result
Exhibits enrolled	10 (12 photos, incl. a 3-angle exhibit)
Languages	8
Narrations generated & **approved**	80 / 80
Recognition accuracy (enrolled exhibits)	Verified match at **0.995** similarity
Visual index	12 office vectors added live, no retraining
Enrolment-to-live per exhibit	**< 5 minutes** incl. 8-language narration
Visitor data sent to cloud	**Zero** — fully on-prem

Engineering hardening surfaced by the pilot

Running on real, constrained hardware surfaced and fixed a series of production issues — the kind of shakedown that only a live deployment reveals:

- **Persistent-storage bug**: container volume mounts pointed one level above the app's data directory, so images, the FAISS index, and audio were being written to ephemeral storage and lost on restart. Corrected to persist across restarts and upgrades.
- **Multilingual TTS**: switched to the multilingual narration model and added a fallback for a broken audio-watermarker in the base image; long Chinese scripts that exceeded GPU memory were tuned to fit.
- **Stability**: model loading and speech synthesis were moved off the request thread so the service healthcheck no longer killed the container mid-generation.
- **GPU budgeting**: with recognition and speech models competing for 8 GB, narration generation was established as an offline/admin activity — informing the "16 GB recommended" guidance for customers who generate audio on the same box.

Every issue was diagnosed, fixed, redeployed, and re-verified on the live system.

Why it matters to a museum

This was not a slideware demo. It is the **same software, same deployment model, same content workflow** a museum receives — proven to:

1. Recognize real objects reliably from a couple of photos, with no training step.
2. Let non-technical staff enrol an exhibit and publish 8-language narration in minutes.
3. Run entirely on one on-prem box, with no visitor data leaving the network.
4. Withstand real hardware constraints and a full hardening cycle.

From here to your museum

The POC Playbook turns this into a 4-week pilot in one of your galleries: ~25 exhibits, your languages, your Wi-Fi, your success metrics — with the pilot's content and analytics carrying straight into rollout.