

Sovereign by Construction

Privacy as a property of home-AI architecture, with a verification test the owner can run themselves.

Pieter Meyer · Elysium Labs · Zürich, Switzerland

ELY-TR-2026-01 · Version 1.6 · First published 11 July 2026 · Revised 11 October 2026

Self-published technical report · not peer reviewed

Abstract

A voice assistant is, physically, a set of microphones and cameras with a network connection. In the dominant commercial architecture, the intelligence that interprets those signals runs in a datacenter, so the signals must leave the house. Vendors govern what happens next through policy, such as retention rules, training opt-outs and deletion requests, and a vendor can rewrite its policy after the data has left the house. Our position is that privacy should be a property of the architecture, not a clause in a contract.

This paper describes the architecture Elysium uses to make that position concrete. The system runs in three placements: Sovereign and Hybrid on a hub in the home, and Cloud as a hosted deployment. Two invariants hold in every placement: only a hub in the home executes device commands, and raw video does not leave the home. In the Sovereign placement, which is how the hub ships, the complete product, including language-model inference, identity and memory, runs on one machine on the home network, and the system fails closed: a fault reduces what the hub can do and never moves a request to a cloud service. Because claims of this kind are easy to make and hard to evaluate, the design treats verifiability as a requirement. The owner can run an egress self-test on the hub, and software updates are signed releases that the hub installs only when the owner starts an update. We describe each mechanism, report what a capable local configuration costs in hardware, and close with an account of what is operational as of October 2026 and what is still in development.

1. Two ways to promise privacy

Privacy assurances in consumer smart-home systems come in two forms. The first is contractual: the vendor states what it collects, how long it retains it, and who may access it. The assurance is real but conditional. It depends on the vendor's current incentives, its security posture, the jurisdictions it operates in, and every future revision of its terms. Once data has crossed the home boundary, the resident's protection is whatever the policy says this quarter.

The second form is architectural: the data never crosses the boundary at all, so outside the house there is nothing for a vendor to retain, delete or hand over. This form has historically carried a cost. Local processing meant pattern-matched commands with little language understanding and no reasoning. Open-weight language models in the 10 to 30 billion parameter range [6], together with quantization methods that fit them into consumer GPU memory [5], change that calculus. Section 5 quantifies it.

2. One application, three placements

Elysium runs the same application code in three placements. On a hub in the home, the owner chooses between Sovereign, which is how the hub ships, and Hybrid. Cloud is a hosted deployment for households without a capable hub. Table 1 sets out where each stage runs in each placement.

Table 1. The three placements: where each stage runs, and what each needs from the network.				
PLACEMENT	PERCEPTION	REASONING	IDENTITY AND MEMORY	NETWORK DEPENDENCE
Sovereign (as shipped)	on the hub	open-weight model on the hub	on the hub	none
Hybrid	on the hub; cloud transcription optional	the model each resident selects: a frontier model in a datacenter or the hub's own	on the hub, linked to a cloud account	required for reasoning in a datacenter
Cloud	hosted; speech may be recognized remotely	frontier model in a datacenter	cloud account	required

Two inputs decide whether reasoning may leave the hub: what the hub is configured to serve, and what the owner has chosen. A small resolver combines them, and anything it cannot explain resolves to Sovereign. The reasoning service then applies a second floor, fixed at installation: on a Sovereign appliance it refuses cloud providers whatever the resolver says. In Hybrid, each resident's requests go to the model that resident has selected, and a resident who selects the hub's own model has their requests answered on the hub. Section 4 sets out where this does not hold: requests spoken to the hub follow the model of one resident, a request that fails on an OpenAI model can go to an Anthropic model, and some background flows reach the cloud whatever a resident selects. The resolver governs language-model reasoning and the optional cloud sign-in link. Speech recognition, speech synthesis and memory extraction are configured when the hub is installed, and each of them asks the resolver before it contacts a cloud service, so none of them reaches one while the hub is in Sovereign. Bringing their configuration under the same runtime choice is planned work.

On an unlocked hub, the owner switches placement at runtime, usually from the dashboard. Moving from Sovereign to Hybrid is the heaviest transition, because it is the only one that opens a path to the outside: the dashboard lists each category of data that will begin to cross, asks for the local password again and records when the owner accepted the list. An owner can also reach Hybrid without that screen, with a command run on the hub or by enabling the cloud link in the hub's configuration before any placement has been chosen. Neither path shows the list or records consent, and closing that gap is planned work. Appliance units ship locked in Sovereign, so cloud reasoning is something a household opts into.

3. Invariants at the boundary

Two rules hold in every placement.

1. **Device control is local.** Only a hub in the home executes device commands, over local radio and network protocols [1, 2], and the hub also holds the command queue, so an internet outage can take away a reasoning capability while lights and other devices keep responding. The hosted Cloud deployment runs no device controller; it records commands but executes none.
2. **Raw video does not leave the home.** Camera frames are processed, if at all, on the hub. In Hybrid, only derived data such as a text description of what a camera saw may cross the boundary.

A third rule depends on placement. On a hub, the listening for the wake word runs on the hub, and room audio stays there until the hub recognizes a request. In Sovereign, the hub also transcribes speech and speaks its replies itself, and no audio or transcript leaves the house. In Hybrid the owner can turn on a cloud recognizer for requests addressed to the assistant, and cloud speech for its replies; the hub's disclosure lists each one while it is on. In both placements the dashboard's microphone button listens only through the hub's own microphone and never uses the web browser's speech recognition. The hosted Cloud deployment may route speech to a remote recognizer, and it is documented as doing so.

Figure 1 summarizes what crosses the boundary in each architecture, compared with the conventional design.

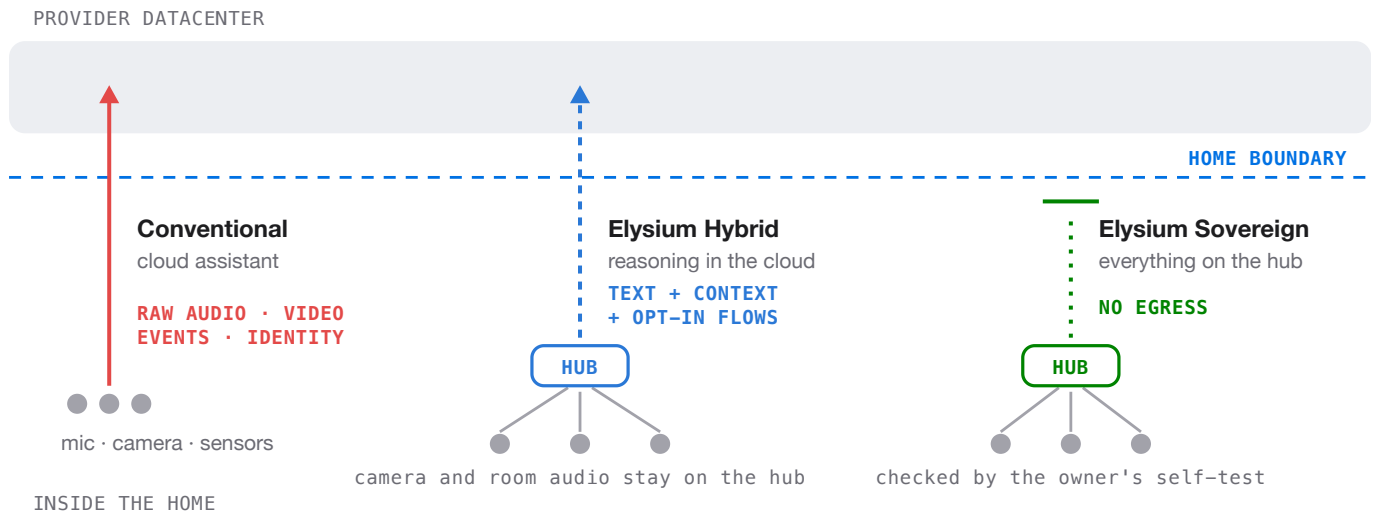


Figure 1. What crosses the home boundary, by architecture. A conventional cloud assistant transmits raw signals because interpretation happens in a datacenter. Elysium Hybrid keeps camera and room audio on the hub and transmits text and context, plus optional flows that are turned on, each one disclosed (Section 4). Elysium Sovereign, as it ships, transmits no household data, and the owner can test that claim (Section 6).

4. Hybrid, stated precisely

Hybrid is the placement a household chooses when it wants the strongest available reasoning while the camera, the wake-word check and device control stay on the hub. It is also the placement that most needs precise language, since it crosses the boundary by design.

When a request goes to a cloud model, the following leave the house with it: the message or the transcript of a spoken request, with the recent messages of that conversation; memories chosen for the request; the rooms and devices with their current state; house instructions and open goals; what the assistant's tools send and return; the resident's display name, preferred reply style and, if they wrote one, the short description in their profile; and the current time and the home's time zone. Until the hub can tell voices apart, it treats every request spoken to it as a request of one resident chosen at setup: the request runs on that resident's model and carries that resident's memories and profile, whoever is speaking, so it leaves the house whenever that resident picked a cloud model. When a resident's OpenAI model fails before it starts to answer, because OpenAI is out of quota, refuses the key or cannot be reached, a hub that also has an Anthropic key sends the same request to Anthropic's Claude Sonnet 4.6. Web research, which any adult resident can turn on with web search, runs on the model of the resident who asks for it: the question and excerpts of the pages read for it leave the house only for a resident who picked a cloud model, and stay on the hub when the hub cannot read that resident's choice. When the hub has an Anthropic key, two more flows go to a small Anthropic model whatever model a resident picks: conversation text, to write titles and summaries; and each reply that changed something in the home, with a short record of the tool calls behind it, so the model can check the reply against what was done. Some flows run only when they are turned on, by the owner in the hub's configuration or, for the morning summary, by any adult resident: the audio of spoken requests, for a cloud recognizer; the text of spoken replies, for cloud speech; the text the hub hears for a few seconds after a spoken reply, with the last request and the reply, so a model can judge whether it was meant for the assistant; spoken requests, for a fast cloud model that tries to answer first; exchanges,

with the resident's saved memories, for cloud memory extraction; memory and request text, for a cloud embedder; and the night's events, for the morning summary. Camera frames, room audio until a request is recognized, device commands and the credentials that control the home stay on the hub. The hub shows these categories, and what stays on the hub, on the screen where the owner confirms the switch and, while cloud reasoning is on, on its trust page and in its self-test; tests keep the three copies word for word identical.

Two properties keep Hybrid from becoming an ordinary cloud product. The wake decision and room audio stay on the hub, so the raw-signal invariants of Section 3 hold wherever reasoning runs. And the crossing is inspectable: the disclosure list is part of the product.

5. Running the whole product at home

Sovereign is useful only if a capable assistant fits on hardware a household can buy. Figure 2 shows how we budget the memory of our reference workstation: the language-model share is what the serving engine reserves, and the speech, vision and synthesis shares are budgets for components designed to run alongside it.

The reference hub is a single workstation-class machine with one consumer GPU carrying 32 GB of memory. The stack is budgeted to run concurrently on that card: speech recognition (about 3.5 GB, sized for a large multilingual recognizer [3]; the voice front end now in integration runs a small recognizer on the main processor), a 14-billion-parameter language model quantized to 4-bit weights [5, 6], a vision model for camera perception (about 1 GB), and neural speech synthesis (about 1.2 GB). The language model's weights occupy about 10 GB. The serving engine [4] reserves 55 percent of the card, roughly 17 to 18 GB, which holds the weights, about 2.7 GB of key-value cache for one full 16,384-token context, runtime workspace, and spare cache that lets several conversations run at once. The cache figure follows from the model's shape: 40 layers, 8 key-value heads of 128 dimensions, two bytes per value for keys and for values, or 160 KiB per token. Memory search on the hub matches text directly and needs no embedding model. The budget totals roughly 23 GB and leaves about 9 GB unallocated, room to raise the serving engine's share for more concurrent conversations or longer contexts.

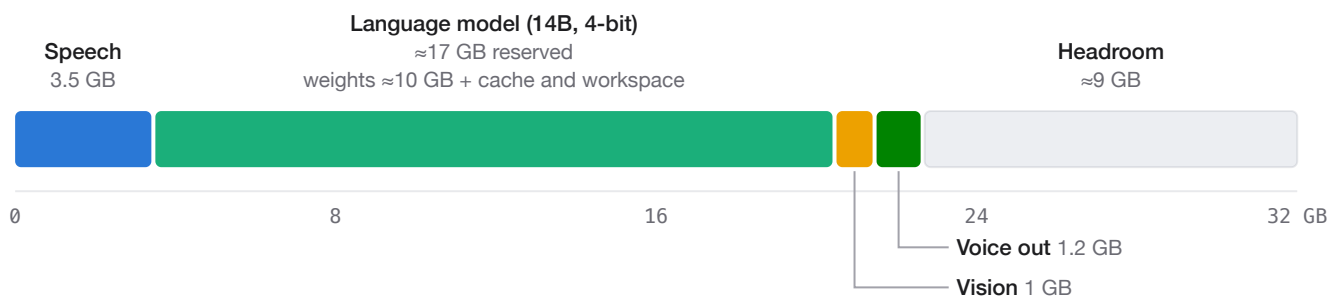


Figure 2. GPU memory budget for the reference hub (32 GB). The language-model bar is the serving engine's reservation, which includes the weights and the key-value cache; the speech, vision and synthesis bars are budgets for components designed to run alongside it. Roughly a quarter of the memory remains to spare.

Configurations are not hand-tuned per machine. The hub measures its own accelerator memory and selects the most capable model that has passed our verification on that kind of hardware. On NVIDIA hardware that is currently the 14-billion-parameter model above, which needs about 26 GB. On Apple-silicon machines, three verified models range from 4 billion parameters to a mixture-of-experts model of about 106 billion parameters. Larger mixture models for NVIDIA cards and a small model for compact hubs are candidates still awaiting verification. Larger hardware runs a more capable local model under the same placement rules.

A 14-billion-parameter model running locally reasons less well than the frontier models available in Hybrid. A Sovereign household accepts that loss to keep every request inside the house. A household that wants the stronger model can switch to Hybrid and still keep sign-in, the memory store and device control on the hub.

6. Failing closed, and proving it

Failure behavior is where privacy architectures are usually falsified in practice. The common pattern is a silent fallback: the local path errors, the software retries against a cloud endpoint, and the request leaves the house without anyone being told. Elysium's placement layer applies the older principle of fail-safe defaults [7]. In Sovereign, a failure of the local model reaches the resident as an error message, and the request goes nowhere else. In Hybrid, a failure of the hub's own model ends the same way; the one retry that moves a request to another provider is the OpenAI-to-Anthropic retry described in Section 4.

During development we found and closed more than one path by which a Sovereign hub could still reach a cloud model: an automatic fallback to a hosted model after a local error, background title generation, and memory extraction on hubs installed without the Sovereign configuration. Title generation was caught by an end-to-end test that forces the local model to fail with a cloud credential planted on the hub and samples the reasoning service's network connections every 0.2 seconds; after the fix, 15 samples taken across a failing conversation showed no public connection. In October 2026 cloud speech and cloud follow-up detection, which judges whether what the hub hears just after a spoken reply was meant for the assistant, were brought under the same rule. We treat any such path as a release-blocking defect.

Residents cannot audit our source code, so the hub gives its owner two checks.

The egress test. The hub ships a self-test the owner can run at any time. In the Sovereign placement it confirms that sign-in is local and that no cloud credentials are configured, reads the connection table of every application service and fails if any holds a connection to a public address, and checks that no service listens on a public interface. With the network cable unplugged, it also confirms that every service is cut off, while the owner checks that the house still answers, signs in and controls devices. The unplugged run shows that the house does not depend on a remote service; the connected run shows that nothing is talking to one at that moment. An owner who wants evidence independent of our software can watch the hub's traffic on their own router.

Updates the owner starts. Consumer devices routinely contact remote services on their own [9]. Elysium releases are signed and verified on the hub against a public key written into the system image [8]. Elysium's software does not poll for releases and accepts nothing pushed to it, and the appliance image switches off the host operating system's own update timers, so host security updates install only when the owner starts them. The owner either downloads a release over the home's own connection, which opens one connection

to the release server for the length of the download, or carries it in on removable media and applies it with no uplink at all. Verification runs in a network namespace with no route to the outside. This mechanism has been exercised end to end on a development unit: releases that were tampered with, wrongly signed or older than the installed one were refused before anything changed, and a valid release was installed, migrated and health-checked. An update whose services then fail that health check is rolled back to the previous release; that path has so far been tested against a simulated container runtime.

A few kinds of egress can be turned on in Sovereign, each off by default: public-data lookups that the owner allows, such as the weather for the home's city or a subscribed calendar, which carry the city name or the calendar address; web search, which any adult resident can turn on and which sends each query to a search engine; an email account that an administrator connects, which the hub then reads and sends from; a tool server that an administrator connects, which receives each call the assistant makes to it; the remote-access tunnel; and the download of model weights when the owner switches local models. The self-test reports whether public-data lookups are on and names the search engine that web search would use. It does not list connected email accounts, tool servers or model downloads, and it accepts a tunnel's connections only when the owner gives it the tunnel's address.

The per-category list from Section 4 appears on the screen where the owner confirms a switch to Hybrid. The self-test and the trust page show it only while cloud reasoning is on; on a Sovereign hub they report instead that reasoning stays on the hub.

7. From self-hosted system to appliance

Architectural privacy is only meaningful to the general public if it does not require a systems administrator. The deployment path therefore has three stages. Today the system runs as a hardened self-hosted installation on the owner's hardware. A one-click installation for common home-server platforms is planned. The last stage is an appliance that arrives with the system installed and receives its application updates through the signed path in Section 6. We have built the appliance disk image and verified, in a virtual machine, that it loads a signed release; booting it on the reference hardware is the next step. Updates to the host operating system and the graphics driver are handled separately, outside the signed path.

The self-hosted installation and the appliance run the same software and placement rules and take the same signed releases.

8. Status and limitations

This section records the state of the system as of October 2026.

Operational: the placement resolver, with runtime switching between Sovereign and Hybrid that, from the dashboard, requires re-authentication and records consent; local identity with offline sign-in; local storage of memory, conversations and configuration; local language-model inference with hardware-fitted model selection; fail-closed behavior in Sovereign, including the speech and memory paths; the egress self-test; the signed offline update path, exercised on a development unit; and control of a Thread light through the hub's own Matter controller, with each command acknowledged by the hub's device bridge within about a second.

In integration: an always-on voice front end, with wake-word detection on the device, speech recognition and spoken replies, and the room-level voice satellite that will carry it into each room. In development: camera perception, a verified local model for compact hubs, and booting the appliance image on the reference workstation. Signed updates for the host operating system are designed, and building them is deliberately deferred; until then the owner applies host updates by hand.

Some limitations will remain as the engineering matures. At any given time a local model will reason less well than the strongest datacenter models. The self-test judges the application services and the connections they hold at the moment it runs; it reports the host operating system's own connections without failing on them, and side channels such as traffic timing [10] are outside its reach. The design treats the home network as trusted, and an attacker already on it is outside the scope of this paper. Every update also rests on our signing key [8], a narrower commitment than trusting a vendor's whole data-handling operation, though the owner extends that trust without being able to check it.

References

- [1] Connectivity Standards Alliance. *Matter Core Specification*, Version 1.4. 2024. <https://csa-iot.org/all-solutions/matter/>
- [2] Thread Group. *Thread 1.4 Specification*. 2024. <https://www.threadgroup.org/>
- [3] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust Speech Recognition via Large-Scale Weak Supervision. *Proceedings of the 40th International Conference on Machine Learning (ICML)*, PMLR 202:28492-28518, 2023. arXiv:2212.04356
- [4] Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient Memory Management for Large Language Model Serving with PagedAttention. *Proceedings of the 29th ACM Symposium on Operating Systems Principles (SOSP '23)*, 611-626, 2023. <https://doi.org/10.1145/3600006.3613165>
- [5] Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration. *Proceedings of Machine Learning and Systems 6 (MLSys 2024)*, 2024. arXiv:2306.00978
- [6] Qwen Team. Qwen3 Technical Report. arXiv:2505.09388, 2025. <https://arxiv.org/abs/2505.09388>
- [7] Saltzer, J. H., and Schroeder, M. D. The Protection of Information in Computer Systems. *Proceedings of the IEEE* 63(9):1278-1308, 1975. doi:10.1109/PROC.1975.9939
- [8] Samuel, J., Mathewson, N., Cappos, J., and Dingledine, R. Survivable Key Compromise in Software Update Systems. *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS '10)*, 61-72, 2010. <https://doi.org/10.1145/1866307.1866315>

- [9] Ren, J., Dubois, D. J., Choffnes, D., Mandalari, A. M., Kolcun, R., and Haddadi, H. Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach. *Proceedings of the Internet Measurement Conference (IMC '19)*, 2019. <https://doi.org/10.1145/3355369.3355577>
- [10] Aphorpe, N., Reisman, D., and Feamster, N. A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic. arXiv:1705.06805, 2017. <https://arxiv.org/abs/1705.06805>
-

How to cite

Meyer, P. (2026). Sovereign by Construction. Technical report ELY-TR-2026-01, version 1.6. Elysium Labs, Zürich, Switzerland. <https://elysium-labs.ai/research/sovereign-by-construction>

Revision history

Version 1.6 · 11 October 2026

Corrected Table 1, which gave Hybrid's reasoning as a frontier model in a datacenter that always needs the network. As Section 2 says, each resident's requests go to the model that resident selects, which can be the hub's own, and reasoning needs the network only when it runs in a datacenter.

Version 1.5 · 11 October 2026

Version 1.4 said that each resident's requests go to the model that resident selected, and left out two cases where they do not. Every request spoken to the hub runs on the model of the one resident the hub files spoken requests under, with that resident's memories and profile, whoever is speaking. And on a hub that also has an Anthropic key, a request that a resident's OpenAI model fails to begin answering is sent again to Anthropic's Claude Sonnet 4.6.

Version 1.4 · 11 October 2026

Web research now runs on the model of the resident who asks for it, so it reaches a cloud model only for a resident who picked one. Version 1.3 listed it among the flows Hybrid sends to the cloud once they are turned on.

Version 1.3 · 10 October 2026

Corrected six statements of version 1.2. It reported one path by which a Sovereign hub could reach a cloud model, where development found several; it described local models up to 30 billion parameters, where one model is verified on NVIDIA hardware; it placed camera perception in active integration, where it is still in development; it said no audio crosses in Hybrid, where an owner can send spoken requests to a cloud recognizer; it described consent to each category of data, where the owner accepts the Hybrid list as a whole, in the dashboard; and it reported rollback exercised on a development unit, where rollback has been tested only against a simulated container runtime. Also corrected the GPU memory figures and the appliance status, described the placement resolver, everything Hybrid sends, the egress Sovereign allows and what the self-test reports, and redrew both figures to match the text. Added in-text citations and complete references; the page also gained a report number, citation formats, a PDF, a table caption and references linked to their records.

Version 1.2 · 26 July 2026

Named the author and labeled the report as self-published and not peer reviewed.

Version 1.1 · 12 July 2026

Restated the GPU memory figures as designed allocations, explained how the language model's share splits into weights and cache, removed the words "by default" from the raw-video invariant, and described the on-hub perception stack as in active integration.

Version 1.0 · 11 July 2026

First published.