🤫

HUSSH TECHNOLOGIES CORPORATION

# Personal Supercomputing Infrastructure

Technical Specifications · Systems Design · Next Generation

A specification for the next generation of personal computing: a fully on-device, consent-first data agent and the infrastructure that lets a person's own machines — and the AI agents acting for them — hold, organize, and act on personal information at supercomputing scale, without ceding custody. This document pairs the company's story with verified hardware specifications and measured results from the implemented system.

Rio Grande — the original Puppy One.

Est. August 2021. The spec every Puppy is built to.

Agentic. Personal. Sovereign.

Revision 2026-06-12 · specifications are reference values, re-verified at quote · TARGET figures are budgeted design goals

01

# Origin: a break-in, and a puppy.

Hushh Technologies Corporation was founded in August 2021. Two arrivals marked that week. The first was the company — incorporated after the founder's family suffered identity theft, and after watching how casually the institutions entrusted with personal information actually handled it. The founding conviction compressed to five words: your data, your business.

The second arrival was a Bernedoodle puppy named Rio Grande. It took four years to recognize that the second explained the first. A dog is a system of remarkable capability bound to a single, legible loyalty: it fetches what you ask, guards the house, runs always-on, and never once sells the household's secrets. Nobody audits a privacy policy to trust a dog — its incentives are observable. That property — high capability with absolute, legible, single-party loyalty — is exactly what personal-data software lacks. It became the design brief for an entire class of system we call hushh One Puppy.

> **Capability without legible loyalty is surveillance. We refused to ship that.**

### The problem, stated as a systems problem

Model the individual as a database. Each person's records are sharded across dozens of institutions — banks, brokerages, insurers, carriers, clinics, retailers, mail providers — each keying its shard on one natural key: the phone number. The data describes the person, but the person has the weakest read path of any party in the system: they cannot query across it, cannot see it whole, cannot revoke a downstream copy. The cloud-AI pattern of the early 2020s made this worse by offering intelligence in exchange for custody. We wanted the inverse primitive: intelligence that runs where the data already lives, under the owner's key, creating no new custodian.

02

# The thesis: move compute to the data.

The classical supercomputing insight applies directly to personal data: when data is large, sensitive, or gravitationally heavy, you move computation to the data, not the data to computation. For seventy years personal information moved to the mainframe, then the server, then the cloud — each migration widening the gap between the owner and control. The hardware finally makes the inverse possible. The most capable computers ever manufactured now fit in a pocket, on a desk, beside a chair, with enough local compute and unified memory to run capable models entirely on-device.

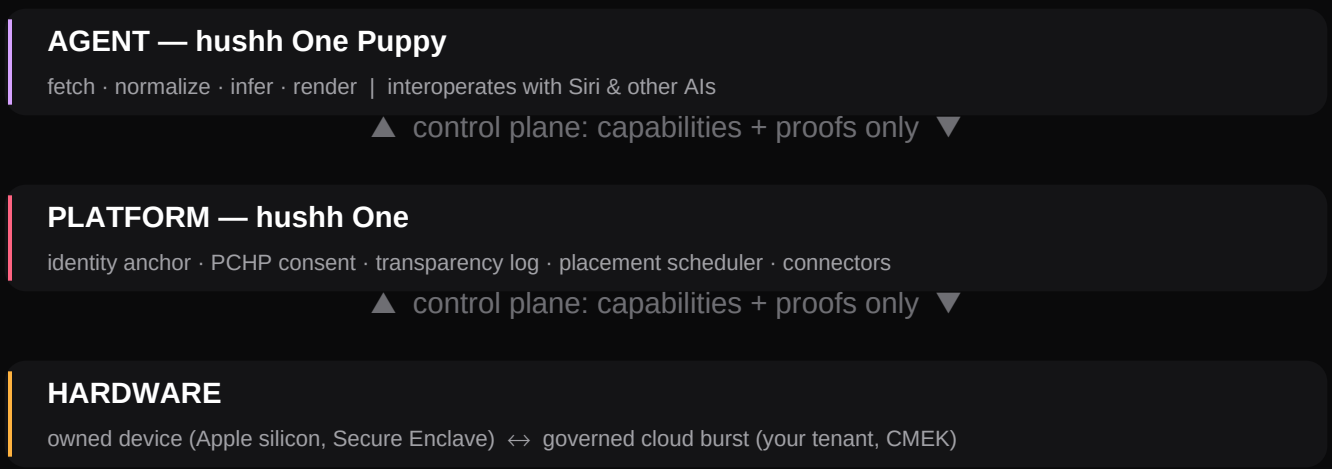
Next-generation personal supercomputing infrastructure is the layer that exploits this: it lets a person's own machines hold and process their own information, and lets trusted software and AI agents use that information efficiently — with consent, with a complete audit trail, and without taking custody. Three invariants fix the architecture.

- **Locality.** Personal data is processed on hardware the owner controls; models run on-device for the common case. Remote compute is the exception, provisioned in a tenant the owner governs — never a new custodian.
- **Legible consent.** Every access emits a cryptographically signed, content-addressed receipt and an entry in an append-only, hash-chained transparency log the owner can read in full. Consent is a protocol, not a checkbox.
- **Single-party sovereignty.** One principal. No second reader — not the vendor, advertiser, or model provider. Revocation is a first-class, low-latency operation.

03

# System architecture

The system separates a data plane (where personal data is read, transformed, stored — always on owned hardware) from a control plane (consent grants, receipts, the transparency log, device lifecycle — small, auditable, the only component requiring coordination). Personal data never traverses the control plane; the control plane carries capabilities and proofs, not content. This is what allows a complete audit trail to exist without the audit system ever seeing the data.




Each layer is independently substitutable. The agent is loyal to one principal and interoperates with platform assistants via intents. The platform is the durable IP: identity anchoring on the phone-number key, the consent engine, the transparency log, the connector adapters, and the placement scheduler that decides on-device versus burst. The hardware layer spans owned silicon and a governed cloud tenant.

04

# Hardware specifications

The infrastructure is hardware-plural by design: it runs on the owner's existing device and scales to personal supercomputers when the workload demands. All figures are reference values from public vendor materials (June 2026); re-verified at quote.

## On-device tier — Apple silicon (shipping; agent-native)

| Device | Compute / memory | Role |
|---|---|---|
| iPhone 17 Pro Max | A19 Pro · 12GB · Secure Enclave | Identity anchor · key root |
| MacBook Pro 16" M5 Max | 40-core GPU w/ per-core Neural Accel · ≤128GB | Extreme on-device inference |
| iPad Pro M5 | M5 · ≤2TB · Pencil Pro | Consent-ceremony / advisor-desk |

*Unified memory bandwidth removes the discrete-GPU memory wall for local models.*

## Personal supercomputer tier — NVIDIA (reservation; Linux runtime on roadmap)

| System | Peak / memory | Class |
|---|---|---|
| DGX Spark (GB10) | 1 PFLOP · 128GB unified · ≤200B-param models | Desk-side Ultra entry |
| DGX Station (GB300 Ultra) | ~20 PFLOPS · 748GB coherent · 1T-param models | Flagship personal |
| RTX PRO 6000 Blackwell | 96GB GDDR7 / card · 1–4× in tower | Local fine-tuning ceiling |

## Governed-cloud burst tier — Google Cloud (your tenant; CMEK)

| Resource | Spec | Burst use |
|---|---|---|
| Ironwood TPU v7 | 4.6 PFLOPS FP8 · 192GB HBM3e · 7.37 TB/s · ≤9,216/pod | Model-scale train/serve · ~44% |
| A4 / A4X (Blackwell) | B200 192GB / GB200 NVL72 >1 EFLOP | CUDA-native bursts · |
| Axion C4A (Arm) | ≤72 vCPU · 576GB DDR5 · ~65% better $/perf | Always-on orchestration |

*Cloud is rented capacity in a tenant the owner governs — owned core, governed surge. Pricing regional; quote live.*

05

# PCHP: consent as a protocol

PCHP (the permission & consent layer) treats every data access as a transaction that must present a valid, unexpired, sufficiently-scoped grant and must emit a receipt. Defaults are closed: a freshly provisioned device consents to nothing until the owner grants, item by item, during the in-person setup ceremony. Receipts are content-addressed and signed by an ed25519 device key held in the Secure Enclave — a receipt proves a specific action over specific fields occurred without disclosing the field values.

**CONTROL-PLANE TYPES (implemented)**

```
Grant   { grant_id, principal, source, scope[], expiry, revoked, sig }
Receipt { grant_id, action, fields[], purpose, content_hash, ts, sig }
LogEntry{ prev_hash, receipt, sig, hash = sha256(prev || canon(body)) }

# revocation writes a tombstone; the scheduler refuses any future action
# whose grant chain is tombstoned — and the refusal is itself logged.
```

The transparency log is append-only and hash-chained: tampering with any past entry invalidates every subsequent hash, giving the owner a verifiable O(n) audit of all access. “What does this agent know about me, and who authorized it” becomes a query the owner can actually run.

**MEASURED — implemented & tested**

ed25519 receipt sign/verify: tamper to any field breaks the signature (test: pass).
Hash-chained log: mutating one past entry fails verify_chain() (test: pass).
Field-scope enforcement: withheld fields never reach the normalized picture (test: pass).
Revoked / expired grant: access raises ConsentError before any fetch (tests: pass).

06

# The fetch pipeline & tail-tolerant executor

The operation that defines the product is first-fetch: from one connected account to a complete, normalized picture with a quantified insight, on the owner's device, in about a minute. The fetch stage dominates the budget and talks to sources we do not control, so it runs through a deadline-aware, fault-tolerant executor applying the tail-tolerance primitives of large-scale systems.

- **Deadline propagation.** A shared end-time; every attempt receives only the remaining budget, so the stage cannot overrun.
- **Bounded concurrency.** A semaphore caps in-flight fetches; connectors run in parallel with progressive assembly (no blocking on the slowest source).
- **Retry + circuit breaking.** Full-jitter exponential backoff for transient failures; a per-source breaker opens after N consecutive faults and half-opens after cooldown.
- **Graceful partial results.** On deadline, return completed sources and mark the rest degraded; the insight reports coverage. Bounded latency with degradation beats unbounded waiting for completeness.

**MEASURED — graceful degradation under a hung source**

Setup: 3 sources; one deliberately hung for 5.000s; stage deadline 1.000s.
Result: pipeline returned in 1.004s with a correct picture from 2 healthy sources.
Hung source cleanly marked degraded (reason: timeout); insight: 'partial: 2/3 sources'.
Transparency log held exactly 2 entries — no receipt for data never accessed.
Suite: 16 tests (consent, log, pipeline, executor) — all passing.

SLO

## Latency budget for first-fetch (TARGET)

The defining operation is budgeted stage-by-stage with headroom and instrumented so the target is measured, not asserted. p50 is materially below the tail target.

| Stage | Budget | Engineering note |
|---|---|---|
| OAuth consent + token | ≤ 8 s | human-paced; one tap, scoped read grant |
| Source enumeration | ≤ 6 s | identify financial senders/domains, 12 mo |
| Fetch + parse | ≤ 25 s | parallel connectors; incremental parse (dominant) |
| Normalize → picture | ≤ 8 s | accounts, balances, recurring, renewals |
| Local inference | ≤ 10 s | on-device model; dollar-denominated insight |
| Render + receipts | ≤ 3 s | picture view + transparency-log writes |
| TOTAL | ≤ 60 s | tail-latency target; bursts a step only if over budget |

07

# Placement, security & threat model

### Placement: owned core, governed surge

Placement is a scheduling decision, not dogma. Default site of computation is the owned device, because that is where the data already is and moving data is the expensive, risky operation. Compute bursts to the governed cloud tenant only when a workload provably exceeds the device — a large fine-tune, a batch beyond local memory, a model larger than unified memory can hold — carrying confidential-computing isolation, customer-managed keys, and the same receipt/log discipline. Personal data crossing that boundary is itself a logged consent event.

### Threat model

- **Central breach — mitigated by construction.** There is no central corpus; compromising the platform yields capabilities and hashes, not anyone's personal data.
- **Silent access — detectable.** Every read is grant-gated and produces a hash-chained log entry the owner can audit.
- **Custody creep — prevented.** No second reader; partner/model agents act only through scoped, revocable grants and never receive a durable copy.
- **Device loss — contained.** Keys in the Secure Enclave; data encrypted at rest; lost device revoked from the control plane and its grants tombstoned.

### Not yet claimed

FedRAMP High authorization is in progress, never stated as held. An attacker with persistent code execution on an unlocked device is outside the model; we reduce blast radius (Enclave keys, least-scope grants) but claim no immunity. Honesty about the boundary is part of the specification.

08

# Implementation status

A design is judged by the honesty of its status section.

| Component | Status |
|---|---|
| PCHP consent: ed25519 receipts + hash-chained log | Implemented · tested |
| Fetch pipeline (fetch/normalize/infer/render) + budget instrument | Implemented · tested |
| Tail-tolerant fetch executor (deadline/concurrency/breaker/partial) | Implemented · tested |
| Reservation backend + Order MCP server (agent-native commerce) | Implemented · tested |
| 60-second on-device first-fetch vs REAL sources (live OAuth) | In development — critical path |
| On-device model selection + inference budget | Prototyping |
| Governed-cloud burst boundary | Designed; not yet wired |
| Agent runtimes: Android / Windows / Linux | Roadmap, demand-gated |
| FedRAMP High authorization | In progress |

**The critical path is the live first-fetch. Everything else exists to make that minute trustworthy.**

INVITATION

## A computer that is finally yours

The personal computer promised the machine would work for the person; the cloud era quietly inverted that. Next-generation personal supercomputing infrastructure is the correction — capability that is local, loyal, legible, and owned, built to the standard a good dog sets: useful every day, and never once betraying the household.

We are recruiting systems engineers, on-device ML engineers, and security engineers who find the above under-specified in the right places. The hard, open, interesting problems are the on-device inference budget, the connector reliability surface, and the placement scheduler.

## Build with us · partner with us · reserve a Puppy

engineering@hushh.ai · partners@hushh.ai · hushh.ai/one