

Run anyone's code. Trust no one's.

TeamHUD lets you drop in third-party and AI-authored widgets — then holds every one of them at arm's length. Isolation is the platform's job, not a promise the widget author makes.

- ✓ Its own sealed sandbox — like an app on your phone
- ✓ No disk or network access unless *you* allow it
- ✓ Only ever gets the permissions you grant
- ✓ Revoke or kill any widget, instantly

THE CONTAINMENT MODEL

Untrusted on the outside. Your data at a protected core.

Every widget sits inside a set of nested boundaries. Nothing it does reaches inward except through a gate you control — and the innermost ring, where your files and the operating system live, it can never touch at all.

● UNTRUSTED — ANYONE'S WIDGET

Third-party or AI-authored UI code. Powerful and useful — and treated as **guilty until granted**.

● ISOLATION — ITS OWN SEALED ROOM

Runs in a dedicated OS-level container. It **can't see the host**, can't read another widget's data, and composites into the overlay looking native — while staying caged.

● PERMISSION GATE — ONLY WHAT YOU GRANT

Every request for data, network, or storage is checked here. What a widget **can** do is exactly what it asked for, intersected with what you approved.

effective = **declared** $\cap$ **granted**

● TRUSTED CORE

Your data · your files · the operating system

Native code and the OS live here — reachable only by TeamHUD's own signed code. Untrusted widgets are **structurally cut off**: they can ask, never reach.

THE MOVING PARTS

From what you see, down to the machine

The same story as one layered stack. Each part does one job and hands off through a checked seam — and the trust level (color) doesn't follow the depth (position). Untrusted code sits both at the top *and* deep near the hardware; it's caged at every level.

 UNTRUSTED  SEMI-TRUSTED  TRUSTED  POSITION = CLOSENESS TO THE HARDWARE

▲ TOP · WHAT YOU SEE AND TOUCH

UNTRUSTED · THE UI

Widget

Any framework you drop in — Lit, React, Angular. The part you look at and click.

UNTRUSTED · ISOLATION

Isolation shell

Runs that UI in its own sealed browser context. Your machine and the host app are invisible to it.

 **SANITIZED BOUNDARY** ————— the only way out — every instruction scrubbed

SEMI-TRUSTED

Overlay surface

Paints the widget into your HUD and forwards your clicks back. It draws; it doesn't decide.

SEMI-TRUSTED

Bridge — the narrow doorway

One small, curated set of calls to the trusted side. Never an open passthrough.

 **PERMISSION CHECK + AUDIT** ————— matched to your grants · logged · denied by default

TRUSTED CORE

Main process

Holds your permission grants and gates network and devices — separately, per widget.

UNTRUSTED · BACKEND

Backend sandbox

When a widget runs logic, it runs in a locked-down process: no filesystem, no network unless you allowed it.

 **SIGNED-CODE ONLY** ————— native power is TeamHUD's code alone

TRUSTED CORE

Native & operating system

Your files, hardware, and the OS. The floor of the stack — reachable by no widget, only by us.

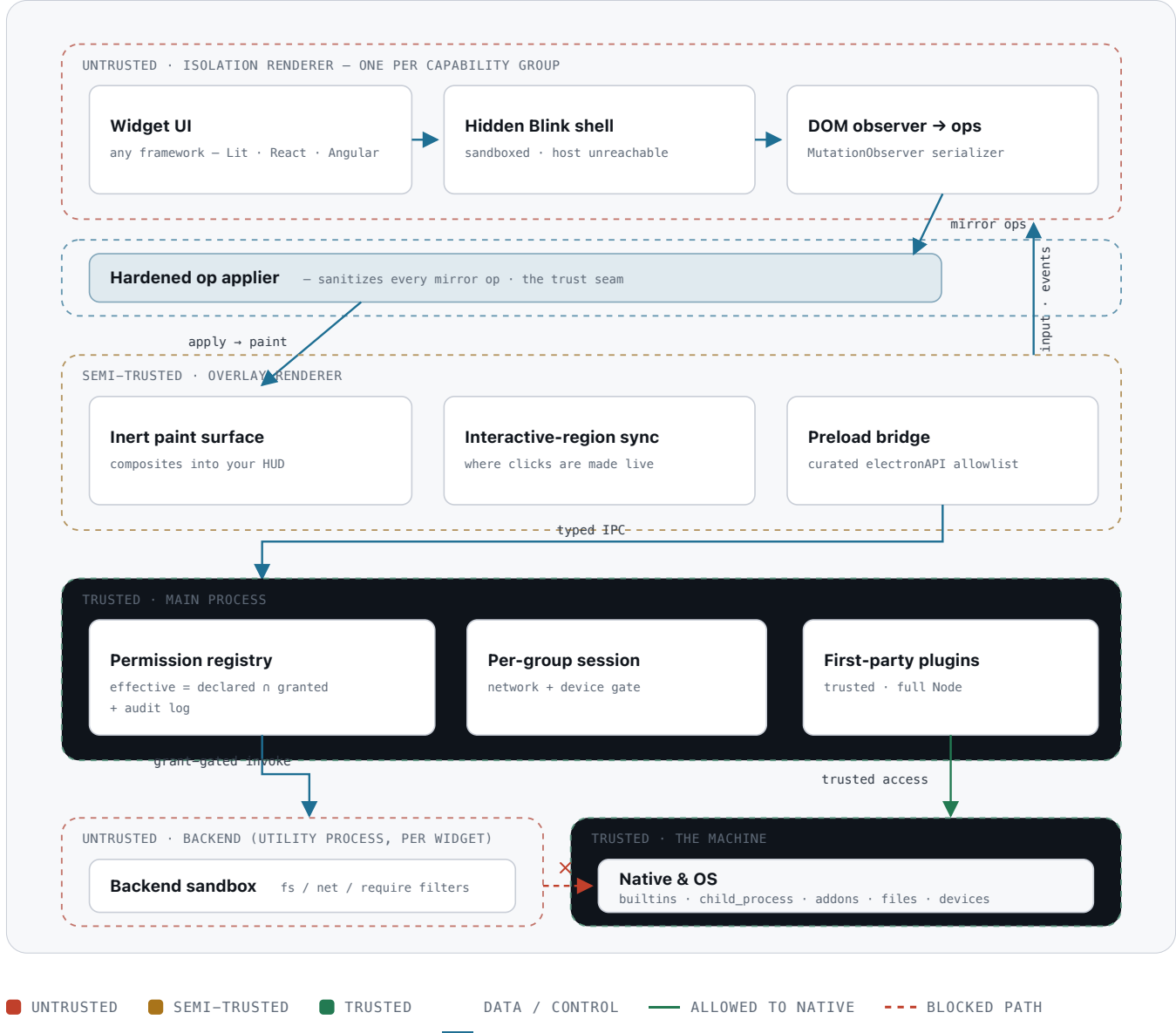
▼ BOTTOM · THE HARDWARE

THE ARCHITECTURE

The same system, drawn for engineers

Boxes are processes and components; arrows are the only channels between them.

Untrusted zones (red) never call the trusted core directly — every line crossing a boundary is a checked hand-off, and the one path from untrusted code to the native machine is *blocked by design*.



FROM TAP TO HARDWARE

What happens when a widget asks to do something

A single action travels through every boundary before anything real happens — and stops at the first one it isn't cleared for.

01

The widget asks

Fetch a feed, save a note, run some logic — the widget requests it, it doesn't just do it.

02

Sanitized at the seam

The boundary scrubs the request. Forged or malicious instructions can't cross the wire.

03

Checked against your grants

Allowed? It proceeds. Not allowed? It's denied and logged — no silent access.

04

Runs in a locked room

Any code execution happens in an isolated process: no filesystem, no off-list network.

05

The core stays sealed

Native code and the OS are only ever touched by TeamHUD's signed code — never the widget's.

WHY IT HOLDS

Four guarantees, built into the platform

Sandboxed by default

Isolation comes from the operating system, not from trusting the author. Every widget is contained the moment it loads — the same way apps are on your phone.

You hold the keys

Permissions are declared up front, granted by you, shown in a plain-language list, and revocable at any time. A kill-switch stops any widget cold.

Least privilege, everywhere

A widget gets exactly what you approved and nothing more — the intersection of what it asked for and what you allowed, enforced on your side of the line.

The core stays cold

Your data, your files, and native code sit at the protected center. Untrusted code is cut off by design — it can send a request, never reach in.

The trust model behind TeamHUD's plugin platform — from third-party UI down to native code.
Sandbox · least-privilege · auditable · revocable.