

```

"use client";

import { useEffect, useState } from "react";
import type { PanelMode, PanelState, NodId } from "../store/types";
import { useSureSailStore } from "../store/sureSailStore";

export default function SureSailPanel(props: {
  nodId: NodId;
  title: string;
  stage: "I" | "II";
  mode: PanelMode;
  state: PanelState;
  statusChip?: { label: string; tone: "green" | "yellow" | "red" | "gray" };
  docs: React.ReactNode;
  children: React.ReactNode;
  footer?: React.ReactNode;
}) {
  const ui = useSureSailStore((s) => s.ui);
  const [wipe, setWipe] = useState(false);

  // Header wipe when mode changes
  useEffect(() => {
    setWipe(true);
    const t = setTimeout(() => setWipe(false), 380);
    return () => clearTimeout(t);
  }, [props.mode]);

  const showDocs = props.mode === "docs";
  const icon = showDocs ? "📖" : "⚙️";

  const tone = props.statusChip?.tone ?? "gray";
  const chipBg =
    tone === "green"
      ? "rgba(80,220,160,0.25)"
      : tone === "yellow"
      ? "rgba(255,200,80,0.25)"
      : tone === "red"
      ? "rgba(255,90,90,0.22)"
      : "rgba(255,255,255,0.45)";

  const neonClass = () => {
    if (props.state === "locked") return "";
    if (ui.dashboardMode === "docs") return ""; // no neon in docs mode
    if (!ui.neon) {
      if (props.state === "loading") return "officeLoading";
      if (props.state === "executing") return "officeExecuting";
      return "";
    }
    if (props.state === "loading") return "neonLoading";
    if (props.state === "executing") return "neonExecuting";
    return "neonIdle";
  }();
}

```

```

return (
  <div
    className={panel ${props.state === "locked" ? "locked" : ""} ${neonClass}}
    data-nodeid={props.nodeId}
  >
    <div className="panelHeader">
      <div className={wipe ${wipe ? "animate" : ""}} />
      <div className="panelHeaderTitle">
        <span className="panelModelcon">{icon}</span>
        <b>{props.title}</b>
      </div>
      <span className="statusChip" style={{ background: chipBg }}>
        {props.statusChip?.label ?? props.state.toUpperCase()}
      </span>
    </div>

    <div className="panelBody">{showDocs ? props.docs : props.children}</div>

    {props.footer ? <div className="panelFooter">{props.footer}</div> : null}
  </div>
);
}

```