

```

"use client";

import SureSailPanel from "./SureSailPanel";
import FlowDiagram from "./FlowDiagram";
import { useSureSailStore } from "../store/sureSailStore";
import { zoneOpsDocs } from "../store/docsCatalog";
import { useMemo, useState } from "react";
import type { Nodeld } from "../store/types";

function DocsBlock({ nodeld }: { nodeld: Nodeld }) {
  const d = zoneOpsDocs[nodeld];
  return (
    <div>
      <div><b>Purpose:</b> {d.purpose}</div>
      <div style={{ marginTop: 6 }}><b>Inputs:</b> {d.inputs.join(" • ")})</div>
      <div style={{ marginTop: 6 }}><b>Outputs:</b> {d.outputs.join(" • ")})</div>
      <div style={{ marginTop: 6 }}><b>Influence:</b> {d.influence.join(" • ")})</div>
      <div style={{ marginTop: 6 }}><b>Dependencies:</b> {d.dependencies.join(" • ")})</div>
      {d.notes ? <div style={{ marginTop: 6 }}><b>Notes:</b> {d.notes}</div> : null}
    </div>
  );
}

export default function StageIColumn({
  orchestrator
}: {
  orchestrator: ReturnType<typeof import("../orchestrator/useRunOrchestrator")>.useRunOrchestrator;
}) {
  const uiMode = useSureSailStore((s) => s.ui.dashboardMode);
  const panelStates = useSureSailStore((s) => s.panelStates);
  const tenCat = useSureSailStore((s) => s.tenCat);
  const gss = useSureSailStore((s) => s.gss);
  const foundation = useSureSailStore((s) => s.foundation);

  const [domain, setDomain] = useState<"SOL" | "BTC" | "ETH" | "MULTI">("MULTI");
  const [behavior, setBehavior] = useState<"WHALE" | "UNDERCHAIN" | "RETAIL">("WHALE");

  const canRun = uiMode === "execution";

  const foundationPct = useMemo(() => {
    if (!foundation) return 0;
    return 100;
  }, [foundation]);

  return (
    <div className="stageWrap">
      <div className="columnHeader" style={{ background: "rgba(14,31,54,0.12)" }}>
        <div>
          <b>██ Stage I — Foundation Scan</b>
          <div className="sub">Build the core field map</div>
        </div>
        <div className="smallNote">Unlocks Stage II</div>
      </div>
    </div>
  );
}

```

```

</div>

<SureSailPanel
  nodeld="N_10CAT"
  title="10 CAT Assessment"
  stage="I"
  mode={uiMode}
  state={panelStates["N_10CAT"]}
  statusChip={{{
    label: tenCat ? `Score ${tenCat.totalScore}` : "10 CAT",
    tone: tenCat ? (tenCat.severity >= 6 ? "red" : tenCat.severity >= 4 ? "yellow" : "green") :
    "gray"
  }}}
  docs={<DocsBlock nodeld="N_10CAT" />}
  footer={
    <div style={{ width: "100%" }}>
      <span>Severity: {tenCat ? tenCat.severity.toFixed(1) : "-"}</span>
    </div>
  }
>
{tenCat ? (
  <div>
    <div><b>Total:</b> {tenCat.totalScore}/100</div>
    <div style={{ marginTop: 6 }}>
      {tenCat.categories.map((c) => (
        <div key={c.name}>
          {c.name}: {c.score}
        </div>
      )))
    </div>
  </div>
) : (
  <div>Run to populate 10-category results (mocked).</div>
)
</SureSailPanel>

<SureSailPanel
  nodeld="N_GSS"
  title="Global Spy Glass (Baseline)"
  stage="I"
  mode={uiMode}
  state={panelStates["N_GSS"]}
  statusChip={{{
    label: gss ? `Risk ${gss.globalRisk.toFixed(1)}` : "GSS",
    tone: gss ? (gss.globalRisk >= 6.5 ? "red" : gss.globalRisk >= 4 ? "yellow" : "green") :
    "gray"
  }}}
  docs={<DocsBlock nodeld="N_GSS" />}
>
{gss ? (
  <div>
    <div><b>IBP:</b> {gss.ibpSync}</div>
    <div><b>Whale Region:</b> {gss.whaleRegion}</div>
    <div style={{ marginTop: 6 }}>{gss.note}</div>

```

```

        </div>
    ) : (
      <div>Baseline global posture + IBP sync snapshot (mocked).</div>
    )}
</SureSailPanel>

<SureSailPanel
  nodeld="N_CTRL"
  title="Controls / Run"
  stage="I"
  mode={uiMode}
  state={panelStates["N_CTRL"]}
  statusChip={{ label: "Run", tone: "gray" }}
  docs={<DocsBlock nodeld="N_CTRL" />}
>
<div style={{ display: "flex", gap: 10, flexWrap: "wrap", alignItems: "center" }}>
  <label>
    Domain ;
    <select value={domain} onChange={(e) => setDomain(e.target.value as any)}
disabled={!canRun}>
      <option value="MULTI">MULTI</option>
      <option value="SOL">SOL</option>
      <option value="BTC">BTC</option>
      <option value="ETH">ETH</option>
    </select>
  </label>

  <label>
    Behavior ;
    <select value={behavior} onChange={(e) => setBehavior(e.target.value as any)}
disabled={!canRun}>
      <option value="WHALE">WHALE</option>
      <option value="UNDERCHAIN">UNDERCHAIN</option>
      <option value="RETAIL">RETAIL</option>
    </select>
  </label>

  <button
    onClick={() => orchestrator.startFoundationRun({ domain, behavior })}
    disabled={!canRun}
    title="Runs Phase 0-2 Foundation flow"
  >
    Run Foundation Scan
  </button>
</div>

<div style={{ marginTop: 10, opacity: 0.85 }}>
  Stage II remains locked until the Foundation gate completes.
</div>
</SureSailPanel>

<SureSailPanel
  nodeld="N_LABEL"
  title="Label Assignment Gate"

```

```

stage="I"
mode={uiMode}
state={panelStates["N_LABEL"]}
statusChip={}
  label: foundation ? `Label ${foundation.label}` : "Gate",
  tone: foundation ? (foundation.label === "RISK" ? "red" : foundation.label ===
"WATCH" ? "yellow" : "green") : "gray"
}
docs=<DocsBlock nodeld="N_LABEL" />
>
{foundation ? (
<div>
  <div><b>Foundation:</b> COMPLETE</div>
  <div><b>Unlock:</b> {foundation.stageIIUnlocked ? "YES" : "NO"}</div>
  <div><b>Completed:</b> {new Date(foundation.completedAt).toLocaleString()}</div>
</div>
) : (
<div>Assembles Stage I outputs into a single snapshot + unlock decision.</div>
)
</SureSailPanel>

<SureSailPanel
nodeld="N_FOUNDATION_PROGRESS"
title="Foundation Progress"
stage="I"
mode={uiMode}
state={panelStates["N_FOUNDATION_PROGRESS"]}
statusChip={{ label: "Progress", tone: "gray" }}
docs=<DocsBlock nodeld="N_FOUNDATION_PROGRESS" />
>
<div className="progressWrap">
  <div className="progressLabel">[■■■■□□□] Foundation Built</div>
  <div className="progressBar">
    <div className="progressFill" style={{ width: `${foundationPct}%` }} />
  </div>
  <div className="smallNote" style={{ marginTop: 6 }}>
    {foundation ? "Stage I complete. Stage II is available." : "Run Foundation to build the
base."}
  </div>
</div>
</div>
</SureSailPanel>

<SureSailPanel
nodeld="N_LIVE"
title="Live Surface Feed (Mock)"
stage="I"
mode={uiMode}
state={panelStates["N_LIVE"]}
statusChip={{ label: "Stream", tone: "gray" }}
docs=<DocsBlock nodeld="N_LIVE" />
>
<div>
  <div>Rotating: Macro • Crypto • Tech • Liquidity</div>

```

```
<div style={{ marginTop: 8, opacity: 0.85 }}>
  (Prototype feed) — this panel never gates scans.
</div>
</div>
</SureSailPanel>

<FlowDiagram />
</div>
);
}
```