```
import type { NodeId, PanelDocs } from "./types";

export const zoneOpsDocs: Record<NodeId, PanelDocs> = {
  N_CTRL: {
    purpose: "Select scan domain & behavior; initiate Foundation run per Execution Order
Map.",
    inputs: ["Domain selection", "Behavior selection", "Run type (Foundation/Full)"],
    outputs: ["run_id context", "orchestrator start"],
    influence: ["Drives downstream adapters", "Controls gating for Stage II"],
    dependencies: ["(root)"]
  },
  N_10CAT: {
    purpose: "10 CAT Assessment (Stage I foundation engine) — category breakdown +
composite score.",
    inputs: ["Contract metadata (mocked)", "Adapter outputs"],
    outputs: ["Category scores", "Total score", "Severity estimate"],
    influence: ["Feeds label assignment", "Contributes to Foundation progress"],
    dependencies: ["N_CTRL"]
  },
  N_GSS: {
    purpose: "Global Spy Glass (baseline) — global risk + IBP posture snapshot.",
    inputs: ["Macro proxies (mocked)", "Underchain posture (mocked)"],
    outputs: ["GlobalRisk", "IBP Sync", "Whale region posture"],
    influence: ["Feeds label assignment", "Completes Stage I unlock gate"],
    dependencies: ["N_CTRL"]
  },
  N_LABEL: {
    purpose: "Foundation assemble + label assignment gate. Unlocks Stage II if complete.",
    inputs: ["10 CAT result", "GSS result"],
    outputs: ["Foundation snapshot", "Label (OK/WATCH/RISK)", "Unlock state"],
    influence: ["Stage II unlock", "Progress bars fill"],
    dependencies: ["N_10CAT", "N_GSS"]
  },
  N_FOUNDATION_PROGRESS: {
    purpose: "Displays Foundation Built progress state for the current run.",
    inputs: ["Phase status", "Completion flags"],
    outputs: ["Progress fill", "Unlock visual cue"],
    influence: ["Unlock line into Stage II"],
    dependencies: ["N_LABEL"]
  },

  N_MICRO: {
    purpose: "Selects Micro Tide / Zoom target context for Stage II tools.",
    inputs: ["Foundation snapshot", "User selected section"],
    outputs: ["Zoom target"],
    influence: ["Sets context for Zoom-In & downstream tools"],
    dependencies: ["N_FOUNDATION_PROGRESS"]
  },
  N_ZOOM: {
    purpose: "Zoom-In generator — deeper breakdown + MV ledger stamp output.",
    inputs: ["Zoom target", "Foundation snapshot"],
    outputs: ["S1–S7 sections", "MV id", "Top-2 outcomes"],
    influence: ["Triggers Mindwake/Toll hooks if severity ≥ 4"],
    dependencies: ["N_MICRO"]
  },
  N_HIST: {
    purpose: "Historical Comparison Mode — compares current MV to prior MV entries.",
    inputs: ["MV id", "Registry store"],
    outputs: ["Similarity tier", "Closest matches"],
    influence: ["Confidence shading"],
    dependencies: ["N_ZOOM"]
  },
  N_TH2: {
    purpose: "Twin Horizon II — Macro Ocean deepening pass.",
    inputs: ["Foundation snapshot", "Macro adapters"],
    outputs: ["Macro tides", "Transmission hints"],
    influence: ["Convergence context"],
    dependencies: ["N_ZOOM"]
  },
  N_TOLL: {
    purpose: "Toll Archive hook — auto-draft internal alert when severity ≥ 4.",
    inputs: ["Zoom output", "Severity", "Tripwire flags"],
    outputs: ["Draft parchment entry (text)"],
    influence: ["Internal recordkeeping + hooks"],
    dependencies: ["N_ZOOM"]
  },
  N_MINDWAKE: {
    purpose: "Mindwake prompt hook — generates image prompt when severity ≥ 4 and
records registry entry.",
    inputs: ["Zoom output", "Severity"],
    outputs: ["Prompt draft", "Registry entry"],
    influence: ["Visual intelligence layer"],
    dependencies: ["N_ZOOM"]
  },
  N_REPLAY: {
    purpose: "Replay Dashboard Launcher — internal-only; never auto-runs.",
    inputs: ["Run logs", "Engine version snapshots"],
    outputs: ["Scorecards", "Accuracy timelines"],
    influence: ["Institutional validation"],
    dependencies: ["(manual)"]
  },

  N_STATUS: {
    purpose: "Header status + version badges + toggles (Neon/Audio/Zone Ops).",
    inputs: ["Store state"],
    outputs: ["UI controls"],
    influence: ["Global UI mode behavior"],
    dependencies: ["(root)"]
  },
  N_LIVE: {
    purpose: "Live Feed surface stream; informational; does not gate runs.",
    inputs: ["Mock feed rotation"],
    outputs: ["UI feed display"],
    influence: ["Context display only"],
    dependencies: ["(none)"]
  }
};
```