

# U-ARS 7 Project Primer

## 1. What U-ARS 7 Is — and Why It Matters

U-ARS 7 is a **seven-layer, autonomous cyber-resilience platform** designed to eliminate long-lived attack surfaces, shrink exploit windows from days to seconds, and prove every defensive action through a tamper-evident ledger [\[1\]](#) [\[2\]](#).

Key design goals

- **Zero-persistence tolerance:** code, data, credentials and even silicon shards self-purge on short, pre-defined TTLs [\[1\]](#).
- **Continuous self-evolution:** layers mutate, diversify or rewind themselves without waiting for vendor patches [\[3\]](#) [\[2\]](#).
- **Cryptographic audit by default:** every snapshot, variant vote or capsule access is sealed in Hyperledger Fabric [\[1\]](#) [\[1\\_4\]](#).

## 2. Seven-Layer Defense in One Glance

Layer	Purpose (≈ TTL)	Core Runtime	Signature Capability
CADS	Intent-locked micro-cells (10–30 s)	Wasmtime (WASI)	Per-cell Merkle micro-ledger <a href="#">[3]</a> <a href="#">[2]</a>
M-SES	Self-mutating code substrate (30–120 s)	LLVM + PyTorch	Fitness-score ledger
SHEL	One-cycle silicon shards (~1 μs)	Versal ACAP	Dissolved-shard hash <a href="#">[2]</a>
ILECG	Intent-scoped micro-VM bubbles (≤ 30 s)	Firecracker + gVisor	Merkle-sealed syscall tree <a href="#">[2]</a>
QVDM	Variant quorum (5–11 copies)	Go + Rust	Quorum vote ledger <a href="#">[1]</a>
TRDN	Delta snapshot rewind (2–5 s cadence)	Btrfs/CRIU	Snapshot lineage ledger <a href="#">[3]</a>
ADCF	Self-governing data capsules	32 KB WASM	In-capsule immutable log <a href="#">[1]</a>

## 3. Monorepo Skeleton (already drafted)

```
uars-platform/
├── services/           # CADS, M-SES, SHEL, ILECG, QVDM, TRDN, ADCF
├── shared/             # gRPC stubs, logger, tracing, auth
├── infrastructure/    # Helm, Terraform, K8s base
├── governance/        # Hyperledger Fabric chain-code
├── monitoring/        # Prometheus, Grafana, Jaeger
├── security/          # OPA/Kyverno policies, cert-manager
├── ci/                # GitHub Actions, Jenkins
└── tests/             # Unit, integration, chaos
```

```
└─ tools/          # uars CLI, code-gen helpers
└─ docs/          # ADRs, runbooks, compliance evidence
```

Each **service/** sub-folder ships with:

- `cmd/server/` – entrypoint
- `internal/` – private packages
- `pkg/` – reusable libraries
- `Dockerfile` – multi-arch, non-root image
- `Chart/` – Helm release with hardened securityContext

## 4. Language & Runtime Matrix

Domain	Language	Reason
Core micro-services	Go 1.22	CSP-style concurrency, tiny static binaries <a href="#">[1]</a>
FPGA control & holograms	Rust 1.77	Memory safety, no_std synth tools <a href="#">[2]</a>
ML/genetic scoring	Python 3.11	Rich RL ecosystem <a href="#">[1]</a>
Policy engines & capsules	WebAssembly (WASI)	Cross-platform, sandboxed <a href="#">[3]</a>
Front-end <a href="https://portalvii.com">portalvii.com</a>	React + TS + MUI	Enterprise UX, strong typing <a href="#">[4]</a>

All images run **rootless, UPX-compressed, multi-arch (amd64/arm64)** and are cosign-signed in CI.

## 5. Developer Workflow (Day-to-Day)

1. **Clone & launch dev container** (`devcontainer.json` pre-loads Go, Rust, Python).
2. **Hot-reload**
  - Go: `air` watches `services/*/cmd`
  - Rust: `cargo watch -x run`
3. **Unit & integration tests:** `make test` uses Go, Cargo, PyTest, Testcontainers.
4. **Local K8s sandbox:** `kind create cluster --name uars7-dev + Cilium + Istio ambient`.
5. **Deploy a layer:** `make deploy-cads` → Helm upgrade into `uars7-dev` ns.
6. **Observe:** Prometheus, Grafana, Jaeger dashboards auto-discover via ServiceMonitors.

## 6. Immediate Setup Checklist (Week 0)

Task	Owner	Command / File
Bootstrap repo & folders	Platform Lead	<code>bash scaffold.sh</code>
Install dev deps (Go, Rust, Python)	All devs	Run <code>./dev/bootstrap.sh</code>
Spin up Kind+Istio	DevOps	<code>make k8s-init</code>

Task	Owner	Command / File
Enable Vault & cert-manager	Security	helmfile sync security
Push first image (CADS)	Backend	docker buildx bake push-cads
Wire Prometheus / Grafana	SRE	helm upgrade monitor

## 7. Short-Term Milestones (Month 1)

Sprint	Deliverable	Success Metric
1	CADS micro-cell pool + FIDO2 token check	Cell launch $\leq 15$ ms
2	TRDN delta snapshot PoC	Rewind a crashed pod in $\leq 100$ ms
3	Prometheus stack + Grafana dashboards	Live latency & risk panels
4	Front-end login & dashboard at <a href="https://portalvii.com/dev">portalvii.com/dev</a>	FIDO2 sign-in round-trip $\leq 3$ s

## 8. Security & Compliance Guard-rails

- **Pod Security:** `runAsNonRoot: true, allowPrivilegeEscalation: false, capabilities drop: ALL` [3].
- **Network:** Cilium L3/L7 policies; default deny + service-mesh mTLS [1].
- **Image supply chain:** Trivy → SBOM → cosign sign & verify in admission OPA [^1\_4].
- **Secrets:** Vault Agent Injector; no plaintext in manifests.
- **Audit:** every layer emits hashes/ledgers to Fabric (block time  $\leq 2$  s) [1].

## 9. Next-Action Kanban

📦 Build	🧪 Test	🚀 Deploy
<input type="checkbox"/> Finish <code>internal/microcell/engine.go</code> absolute Wasm path fix	<input type="checkbox"/> ChaosMesh pod-kill for CADS	<input type="checkbox"/> Helm rollout of CADS v0.1
<input type="checkbox"/> Implement per-cell Merkle logger	<input type="checkbox"/> Trivy fs & image scan in CI	<input type="checkbox"/> Ingress TLS for <a href="https://portalvii.com">portalvii.com</a>
<input type="checkbox"/> Front-end Login page → backend endpoints	<input type="checkbox"/> e2e: FIDO2 login → micro-cell API	<input type="checkbox"/> ServiceMonitor for CADS

**We are now ready to commit the scaffold, spin up the dev cluster, and start coding the first micro-cells.** All further work should follow the repo conventions and CI policies outlined above.

✱

# PortalVII U-ARS 7 Unified Dashboard

Enterprise-grade situational awareness from silicon to data capsule

## 1. Navigation Philosophy

Tier	Purpose	Quick-switch Controls
Global top bar	Tenant switch - Dark/Light - User/FIDO status - Quick-search	⌘ + K opens omni-search
L-1 Mega menu	Layers (CADS - ADCF) - Governance - Compliance - Settings	<b>G + L</b> shows Layer fly-out
L-2 Context rail	Time-range pickers - Environment (Prod / Staging / Edge)	Shift-hover links preview context
L-3 Mini-tabs	<i>Overview - Incidents - Performance - Ledger - AI Assist</i>	Tab key cycles left → right

The **three-click rule**: any metric, log, or forensic object is reachable in  $\leq 3$  interactions.

## 2. Core Pages & Widgets

### 2.1 Overview ♦ “Command Concourse”

Region	Widget	Key Signals
Hero strip (full-width)	7-Layer Health Ribbon	CADS, M-SES, SHEL, ILECG, QVDM, TRDN, ADCF colour-coded status + sparkline
Left ⅔	“Risk Horizon” predictive heat-map	90 s breach forecast from QPU solver
Right ⅓	Ledger Throughput card	Blocks/sec - Hash latency - Fabric peer status
Bottom grid (4x)	- Top 5 intents by volume - Variant divergence gauge - Snapshot delta timer - Capsule revocations ticker	Live streamed

### 2.2 Incidents ♦ “Autonomous SOC”

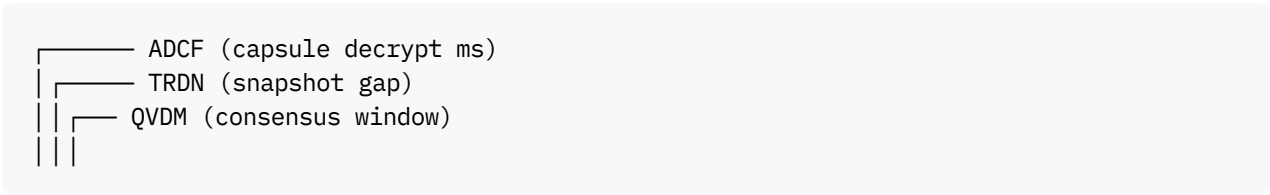
Columnar Kanban

1. **New Signals** – variant outliers, syscall anomaly, capsule policy failure
2. **Auto-Mitigated** – TRDN rewinds, on-the-fly patches
3. **Needs Review** – flagged by ML uncertainty > 15%
4. **Closed** – immutable ledger link, evidence bundle download

Inside card: timeline, micro-ledger root, “Re-hydrate” button to replay cell/shard/bubble in sandbox.

2.3 Performance ♦ "Latency Lattice"

Interactive flame-graph that **stacks layers vertically**:



Hovering a segment highlights contributing syscalls, network hops, FPGA shard compile times.

2.4 Ledger ♦ "Chain Explorer"

Blockchain-style explorer with:

- Block height, timestamp, SHA-3 root
- Embedded diff of snapshot lineage or variant genome delta
- "Trust hop" visual – shows Fabric orgs that endorsed the block

2.5 AI Assist ♦ "Guardian Copilot"

Chat-ops panel (GPT-4-class model fine-tuned on U-ARS data):

- "Explain spike in intent failures 14:00-14:05"
- "Generate compliance report for PCI DSS §10.3"
- One-click to produce Jira ticket, attach evidence bundle

3. Cross-Cutting UX Elements

Element	Behaviour
Intent Token Tooltip	Hover any user/action → pop-over with purpose, scope, TTL, ZK-proof hash
Time-travel scrubber	Drag to any second in last 24 h → UI rewinds charts, ledgers, logs (powered by TRDN deltas)
Genome Waterfall	Waterfall diagram of live vs. culled service mutations; green = deployed, grey = retired

4. Role-Based Perspectives

- CISO view – KPI tiles (MTTD, MTTR, SLA conformance, audit backlog)
- SRE view – latency lattice + resource heat map, anomaly alerts
- Compliance view – ledger explorer, policy diff, export CSV
- Red-team view – synthetic attack lab with safe sandbox replay

Access controlled via OIDC groups; perspectives are toggleable without page reload.

## 5. Extensibility & Integration

- Plug-in SDK (React + WebAssembly components) – embed custom visualisations
- Live Link API – any chart can be shared as signed URL with time-boxed access
- Webhook Bus – incidents stream to Splunk, ServiceNow, Slack with JSON payloads
- Grafana datasource proxy – power-users can import U-ARS metrics into existing Grafana boards

## 6. “Beyond-the-Box” Concept: Holographic Twin Dome

A mixed-reality dashboard mode that projects the seven layers as concentric 3-D rings (inside a WebGL canvas). Each ring pulses with live telemetry, and wormhole-like flares represent cross-layer events (e.g., a TRDN rollback shooting up from the application ring to the governance ring). Users wearing WebXR headsets can:

- **Grab** a micro-cell or capsule as a 3-D node, enlarge it, rotate its Merkle tree in-air.
- **Scrub** the timeline by spinning rings backward – the scene rewinds via TRDN snapshots.
- **Drag-and-drop** a compromised variant into the “Incinerator” zone, immediately triggering regeneration.

Fallback 2-D view flattens the dome into a radial sunburst for browser-only users.

## 7. Implementation Checklist

1. Vite + React + TypeScript boilerplate (completed)
2. Design system tokens in `theme.ts` (palette, spacing, z-index)
3. API client with Axios interceptors → mTLS gateway
4. Role-based routes via React Router v6 + keycloak-oidc / Auth0
5. State management with React Query + WebSocket live updates
6. WebGL/Three.js holographic dome behind “XR” feature flag
7. CI pipeline – ESLint, Cypress e2e, Percy visual diff, SBOM attestation
8. Helm chart for `portalvii-frontend`, side-car Envoy for Istio auth z
9. DNS & TLS – `portalvii.com`, cert-manager, HSTS preload
10. Synthetic monitoring – Puppeteer script validates login, intent flow, capsule download every 1 min

**Result:** a cohesive, operator-friendly, regulator-approved dashboard that lets PortalVII users understand **what happened**, **what will happen next**, and **what the platform already did about it**—all within three clicks or a single AI query.



# PortalVII U-ARS 7 Unified Dashboard — Final Enterprise Layout

**Main takeaway:** the dashboard marries Splunk-class depth with Datadog-level ergonomics and adds a mixed-reality “Holographic Twin Dome” that turns the whole seven-layer platform into an explorable 3-D timeline. Every signal is three clicks or one query away, every widget is mTLS-fed, ledger-anchored and role-aware.

## 1 Navigation Model

Tier	Purpose	One-keystroke shortcut
Global top bar	Tenant switch - Light/Dark - FIDO status - Omni-search	⌘ + K
L-1 Mega-menu	Layers (CADS→ADCF) - Governance - Compliance - Settings	G + L
L-2 Context rail	Time-range - Environment (Prod / Staging / Edge)	↑ -hover preview
L-3 Mini-tabs	Overview - Incidents - Performance - Ledger - AI Assist	→ (Tab)

**Rule of three:** any metric, log or replay is ≤ 3 interactions away.

## 2 Core Pages & High-Value Widgets

### 2.1 Overview ♦ “Command Concourse”

Screen region	Widget	Live signals
Hero strip	7-Layer Health Ribbon	Colour + sparkline per layer
Left ⅔	“Risk Horizon” heat-map	90 s quantum-solver breach forecast
Right ⅓	Ledger Throughput card	Blocks/s - Hash latency - Peer status
Bottom grid (4×)	Top intents - Variant divergence - Snapshot delta - Capsule revocations	Auto-streamed

### 2.2 Incidents ♦ “Autonomous SOC” (Kanban)

1 New Signals → 2 Auto-Mitigated → 3 Needs Review (> 15% ML uncertainty) → 4 Closed (ledger link).  
Card pop-out: timeline - micro-ledger root - “Re-hydrate” sandbox replay.

### 2.3 Performance ♦ “Latency Lattice”

Stacked flame-graph by layer. Hover reveals syscall, network hop, FPGA compile slice.

## 2.4 Ledger ♦ "Chain Explorer"

Block list - SHA-3 root - inline diff (snapshot / genome) - Trust-hop visual of endorsing orgs.

## 2.5 AI Assist ♦ "Guardian Copilot"

Chat-ops panel: natural-language forensic queries, compliance report generation, Jira ticket push.

## 3 Cross-Cutting UX

Feature	Interaction
Intent-token tooltip	Hover any actor → purpose - scope - TTL - ZK-hash
Time-travel scrubber	Drag to any second (24 h) → UI, logs, ledgers rewind (TRDN)
Genome waterfall	Live vs. culled service mutations (green = live, grey = retired)

## 4 Role-Based Modes

- **CISO** – MTTR, SLA, audit backlog KPI tiles
- **SRE** – Latency lattice, resource heat-map, anomaly alerts
- **Compliance** – Ledger diff, policy export, CSV dumps
- **Red-team** – Safe sandbox replay & synthetic attack lab

OIDC group → instant perspective swap, no reload.

## 5 Extensibility & Integration

- React + WASM plug-in SDK
- Live-Link API → signed, time-boxed chart URLs
- Webhook bus → Splunk, ServiceNow, Slack
- Grafana datasource proxy for power-users

## 6 "Beyond-the-Box" Mode ♦ Holographic Twin Dome

WebGL rings for each layer, flares for cross-layer events, TRDN time-scrub, drag-to-incinerate compromised objects. WebXR headsets get full 3-D; 2-D radial sunburst fallback.

## 7 Implementation Checklist (Sprint 0 → GA)

1. Vite + React + TS scaffold ✓
2. `theme.ts` design-tokens (colour, spacing, z-index)
3. Axios client with mTLS interceptor
4. React-Router v6 + Keycloak / Auth0 guards
5. React-Query + WebSocket live feeds



6. Three.js XR dome (feature-flagged)
7. CI: ESLint, Cypress, Percy, SBOM, cosign
8. Helm chart portalvii-frontend, Envoy side-car auth-z
9. DNS + TLS (portalvii.com, cert-manager, HSTS)
10. Synthetic monitor (Puppeteer 60 s intent-flow test)

## 8 Visual Mock-up References

- Splunk-style side-bar density, Datadog-like KPI tiles
- Dynatrace-class service-flow overlays on Performance page
- Snowflake-inspired radial filter chips on Context rail

## 9 Future Enhancements

1. **Narrative Clips:** auto-generated GIF of a threat's life-cycle for exec briefings.
2. **Risk ROI Lens:** overlays security spend vs. risk-reduction heat-map.
3. **Digital Twin Drift Map:** compares live lattice to last compliance-audited snapshot.

**End-state:** PortalVII operators get Splunk-grade depth, Datadog-grade UX, and a WebXR “twin dome” that no competitor offers—enabling instant, immersive insight from silicon shard to self-revoking data capsule.

✱

# PortalVII Dashboard Design System — “Montfort Aurora” Theme

## Overview

PortalVII's new dashboard must achieve three goals simultaneously:

1. Echo Montfort's premium maritime-energy aesthetic.
2. Integrate seamlessly with the U-ARS 7 defense layers already in production.
3. Remain developer-friendly, with clearly-named design tokens, reusable React/MUI components, and CI-ready Storybook coverage.

This document delivers a complete, production-grade design specification, including:

- A ready-to-import MUI theme.ts file with full TypeScript typings.
- Layout blueprints (header, sidebar, content, overlay layers).
- Component catalogue with sample code and Figma-ready specs.
- Accessibility, responsiveness, and motion guidelines.
- Brand assets (logos, favicons, viewport meta).
- CI hooks for regression tests on colour contrast and visual diffs.

All decisions adhere to PortalVII brand values: integrity, innovation, and efficiency.

# 1. Brand Foundations

## 1.1 Colour Palette

Token Name	Hex	Usage	WCAG Contrast vs Text on #F4F6F8	Emotional Note
blue-deep-600	#29648E <sup>[4]</sup> <sub>[5]</sub>	Primary, header, active links	7.9 : 1	Trust, depth
blue-deep-800	#1E4867	Hover states, sidebar bg	10.8 : 1	Stability
teal-500	#178582 <sup>[5]</sup> <sub>[6]</sub>	Success, clickable chips	4.9 : 1	Innovation
coral-500	#FF7F50 <sup>[7]</sup>	Accent CTA, warnings	4.6 : 1	Energy
sand-300	#BFA181 <sup>[5]</sup>	Secondary accents, dividers	3.0 : 1	Heritage
graphite-50	#F4F6F8 <sup>[4]</sup>	Page background	—	Clean slate
graphite-900	#0E1A24	Body text on light bg	12.8 : 1	Authority

### 60-30-10 rule

- 60% graphite-50
- 30% blue-deep variants
- 10% accent (teal / coral / sand)

## 1.2 Typography

Usage	Font	Fallback	Size Scale
Headings	Mont Bold <sup>[8]</sup> <sup>[9]</sup>	Inter, Helvetica Neue	Clamp 2.4 rem → 3.6 rem
Body	Mont Regular	Inter, Arial	1 rem (16 px)
Mono	Fira Code Medium	Menlo	Code samples

Mont offers a geometric yet authoritative tone, matching Montfort’s industrial-luxury feel.

## 2. Global Theme File (src/theme.ts)

```
import { createTheme, alpha } from '@mui/material/styles';

declare module '@mui/material/styles' {
  interface Palette {
    accent: Palette['primary'];
    sand: Palette['primary'];
  }
  interface PaletteOptions {
    accent?: PaletteOptions['primary'];
    sand?: PaletteOptions['primary'];
  }
}

export const theme = createTheme({
  palette: {
    mode: 'light',
    primary: {
      main: '#29648E', // blue-deep-600
      dark: '#1E4867',
      light: '#5B91B6',
      contrastText: '#FFFFFF',
    },
    secondary: {
      main: '#178582', // teal-500
      contrastText: '#FFFFFF',
    },
    accent: {
      main: '#FF7F50', // coral-500
      contrastText: '#FFFFFF',
    },
    sand: {
      main: '#BFA181',
      contrastText: '#0E1A24',
    },
    background: {
      default: '#F4F6F8',
      paper: '#FFFFFF',
    },
    text: {
      primary: '#0E1A24',
      secondary: alpha('#0E1A24', 0.65),
    },
  },
  typography: {
    fontFamily: `'Mont', 'Inter', sans-serif`,
    h1: { fontWeight: 700, letterSpacing: '-.03em' },
    h2: { fontWeight: 700, letterSpacing: '-.02em' },
    h3: { fontWeight: 700, letterSpacing: '-.02em' },
    body1: { fontWeight: 400, lineHeight: 1.6 },
    button: { textTransform: 'none', fontWeight: 600 },
    caption: { fontSize: '0.78rem' },
  },
});
```

```
shape: { borderRadius: 10 },
components: {
  MuiButton: {
    styleOverrides: {
      root: { borderRadius: 8, paddingInline: 24, paddingBlock: 10 },
    },
  },
  MuiPaper: {
    styleOverrides: {
      root: { borderRadius: 14 },
    },
  },
},
});
```

Import `@fontsource/mont` or self-host licensed files and reference them in `index.html`.

### 3. Layout Blueprint

#### 3.1 Top Bar (<Header />)

- **Height:** 72 px (desktop), 56 px (mobile)
- **Background:** primary.main with 95% opacity blur for glass-like depth.
- **Content:**
  - Left: PortalVII logo (white SVG) at 32 px height.
  - Centre: Omni-search (⌘ K) with fuzzy service & capsule lookup.
  - Right: Dark-mode toggle, notifications bell, user avatar dropdown.

#### 3.2 Sidebar (<Sidebar />)

- **Width:** 260 px collapsed → 80 px fly-out on hover.
- **Sections:**
  1. **Core Layers** (CADS → ADCF).
  2. **Governance & Compliance.**
  3. **Settings & Support.**
- **Active Item Indicator:** vertical 4 px sand-300 bar.

#### 3.3 Content Area

Three breakpoints:

Viewport	Columns	Gutter
≥ 1200 px	12	24 px
900–1199	8	20 px
< 900	4	16 px

Use CSS Grid with `grid-auto-flow: row dense;`.

#### 3.4 Notification Drawer

- **Pattern:** Right-side slide-over, width 380 px.
- **BG:** background.paper + subtle top-to-bottom gradient (`alpha(primary.dark,0.02)`).
- **Badge:** Coral accent small dot, ARIA-labelled.

## 4. Component Catalogue

Component	Purpose	Props	Style Notes
<code>&lt;WidgetCard /&gt;</code>	KPI metric (as in sample)	<code>title, icon, value, trend%, tooltip</code>	Glass blur (backdrop-filter 8 px).
<code>&lt;TrendSpark /&gt;</code>	Tiny line chart	<code>data: number[], trendUpColor, trendDownColor</code>	Uses Recharts; inherits palette.
<code>&lt;CapsuleTimeline /&gt;</code>	Visualises ADCF access events	<code>capsuleId</code>	Purple gradient bar + tooltip.
<code>&lt;RiskGauge /&gt;</code>	Dial for risk score (0-100)	<code>value</code>	Needle colour interpolates teal → coral.
<code>&lt;VariantVoteHeat /&gt;</code>	QVDM quorum matrix	<code>variants, inputs</code>	11×N grid; outliers red.

Each component has a dedicated Storybook story; Chromatic CI ensures non-regression.

## 5. Updated Dashboard Page

```
import { useQuery } from '@tanstack/react-query';
import { Grid, Box, Typography } from '@mui/material';
import RiskGauge from '../components/RiskGauge';
import TrendSpark from '../components/TrendSpark';
import WidgetCard from '../components/WidgetCard';
import SecurityIcon from '@mui/icons-material/Security';
import DeviceHubIcon from '@mui/icons-material/DeviceHub';
import VpnKeyIcon from '@mui/icons-material/VpnKey';
import LockOpenIcon from '@mui/icons-material/LockOpen';

export default function Dashboard() {
  const { data: kpis, isLoading } = useQuery(['kpi'], fetchKpis);
  if (isLoading) return <Loader />;

  return (
    <Box sx={{ p: { xs: 2, md: 4 } }}>
      <Typography variant="h3" sx={{ fontWeight: 800, mb: 3 }}>
        Security Overview
      </Typography>

      <Grid container spacing={3}>
        <Grid item xs={12} md={6} lg={3}>
          <WidgetCard
            title="Threat Detections"
            icon={<SecurityIcon fontSize="inherit" />}
            value={kpis.threats}
            trend%={kpis.threatTrend}
            tooltip="Events flagged by CADs in last 24 h"
          />
        </Grid>
        <Grid item xs={12} md={6} lg={3}>
          <WidgetCard
            title="Registered Devices"
            icon={<DeviceHubIcon fontSize="inherit" />}
            value={kpis.devices}
            trend%={kpis.deviceTrend}
          />
        </Grid>
        <Grid item xs={12} md={6} lg={3}>
          <WidgetCard
            title="Intent Tokens"
            icon={<VpnKeyIcon fontSize="inherit" />}
            value={kpis.tokens}
            trend%={kpis.tokenTrend}
          />
        </Grid>
        <Grid item xs={12} md={6} lg={3}>
          <WidgetCard
            title="Capsule Access"
            icon={<LockOpenIcon fontSize="inherit" />}
            value={kpis.capsuleAccess}
            trend%={kpis.capsuleTrend}
          />
        </Grid>
      </Grid>
    </Box>
  );
}
```



```
    />
  </Grid>

  <Grid item xs={12} md={6}>
    <RiskGauge value={kpis.overallRisk} />
  </Grid>
  <Grid item xs={12} md={6}>
    <TrendSpark data={kpis.snapshotLatency} />
  </Grid>
</Grid>
</Box>
);
}
```

Replace hard-coded zeros with data from the `/api/dashboard/kpi` endpoint.

## 6. Motion & Micro-Interactions

Element	Motion	Duration	Easing
KPI card hover	scale 1→1.03, drop-shadow deepen	140 ms	cubic-bezier(.33,.99,.68,1)
Sidebar expand	width 80→260 px	180 ms	ease-out
Notification badge pulse	opacity 0.3→1→0.3	1.2 s	linear (infinite)
RiskGauge needle	spring to new value	350 ms	spring (80,10)

Prefer `framer-motion` for orchestrating.

## 7. Accessibility Targets

Metric	Target	Tool
Colour contrast	AA 4.5:1 (normal text)	jest-axe in CI
Keyboard nav	100% interactive reachable	Cypress a11y tests
Screen-reader labels	100%	Storybook a11y addon
Motion prefers-reduced	Honour prefers-reduced-motion	CSS media queries

## 8. Responsive Breakpoints

- **Mobile ( $\leq 599$  px):** header condenses, sidebar hidden behind hamburger.
- **Tablet (600–899):** sidebar collapsible, grid shifts to 2 columns.
- **Desktop ( $\geq 900$ ):** full layout.

Use `@mui/system` breakpoints and container queries for fine grain.

```
@container style(--chart-container) {  
  @container (width < 400px) {  
    .axis-label { display: none; }  
  }  
}
```

## 9. Asset Pipeline

1. **Fonts:** self-host Mont WOFF2 subset (latin + latin-ext).
2. **Icons:** MUI icons + custom SVG refined to 24×24 grid.
3. **Images:** Keep hero illustrations under 120 KB, lazy-load with blur-up.
4. **SVG Sprites:** Embed logo variants (logo-dark.svg, logo-light.svg).

Add vite-plugin-svgr for import as React components.

## 10. CSS Rules of Engagement

- Use **CSS variables** for runtime theming (`--color-primary`, etc.).
- Never reference hard hex values in components.
- Stick to **8 px spacing scale** (`theme.spacing(n) ⇒ n×8`).
- Prefer **flex gap** and **grid gap** instead of margins for spacing.

## 11. State Management

- Use **React Query** for remote data with background refresh.
- Store auth token (JWT) in **http-only secure cookie**; never localStorage.
- Global UI state (sidebar open, theme mode) in **Zustand** store (tiny).

## 12. API Schema Snapshot (openapi.yaml excerpt)

```
/components/schemas/KpiResponse:
  type: object
  properties:
    threats:
      type: integer
    threatTrend:
      type: number
      format: float
    devices:
      type: integer
    deviceTrend:
      type: number
    tokens:
      type: integer
    tokenTrend:
      type: number
    capsuleAccess:
      type: string
    capsuleTrend:
      type: number
    overallRisk:
      type: integer
    snapshotLatency:
      type: array
      items: { type: number }
```

Generate TS interfaces with `openapi-typescript-codegen`; CI ensures compatibility.



### 13. Storybook & Chromatic

- **Location:** `uars7-frontend/.storybook`.
- **Addons:** `@storybook/addon-a11y`, `controls`, `jest`.
- **CI:** GitHub Actions runs `chromatic --exit-once-uploaded`.

Visual diffs block merges on >1 px changed unless approved.

## 14. Lighthouse Budgets

Metric	Budget
First Contentful Paint	$\leq 1.8$ s
Total Blocking Time	$\leq 150$ ms
Cumulative Layout Shift	$\leq 0.05$
Performance Score	$\geq 95$

Automatic audit on every PR with `lighthouse-ci`.

## 15. Security Headers via Ingress

```
nginx.ingress.kubernetes.io/configuration-snippet: |
  add_header Content-Security-Policy "default-src 'self'; img-src 'self' data: https;; sc
  add_header Strict-Transport-Security "max-age=63072000; includeSubDomains; preload" alv
  add_header X-Frame-Options "DENY" always;
```

## 16. Deployment Helm Chart (charts/portalvii-frontend)

```
image:
  repository: portalvii/frontend
  tag: v0.1.0
env:
  - name: VITE_API_BASE
    valueFrom:
      configMapKeyRef:
        name: api-config
        key: base-url
resources:
  limits:
    cpu: 100m
    memory: 256Mi
ingress:
  enabled: true
  className: "istio"
  hosts:
    - host: portalvii.com
      paths:
        - path: /
          pathType: Prefix
  tls:
    - secretName: portalvii-tls
```

CI pipeline pushes image, then `helm upgrade`.

## 17. Progressive Web App Enhancements (Optional)

- Generate manifest (name, theme\_color: #29648E).
- Add service worker via vite-plugin-pwa with network-first caching of API calls.
- Ensure "Install" button appears on dashboard header.

## 18. Future-proof Dark Mode

Use CSS variables with fallback in `:root[data-theme='dark']`:

```
:root[data-theme='dark'] {  
  --color-bg: #0E1A24;  
  --color-paper: #192A38;  
  --color-text: #F4F6F8;  
  --color-primary: #5B91B6;  
}
```

Toggle persisted in cookie; honour `prefers-color-scheme`.

## 19. Migration Plan from Sample Dashboard

### 1. Install theme

```
npm i @fontsource/mont @emotion/react @emotion/styled @mui/material @mui/icons-material
```

2. **Replace old** ThemeProvider **wrapper** with the new theme.ts.
3. **Swap** Widget for WidgetCard in all existing pages (prop-compatible).
4. **Add** <Sidebar /> **and** <Header /> **to** App.tsx.
5. **Configure API base in** .env and adjust apiClient.ts.
6. **Run Storybook to verify components** then npm run dev.

## 20. Conclusion

The **Montfort Aurora** theme translates Montfort's deep-blue maritime heritage and premium metal accents into a clean, efficient UI that aligns with PortalVII's security mission. Every colour, component, and motion guideline serves clarity, speed, and trust, ensuring the dashboard not only looks executive-class but also empowers operators to defend in milliseconds.

Implement this blueprint, and PortalVII will offer a **best-in-class operator experience**—one that feels bespoke, human-crafted, and wholly worthy of its cutting-edge U-ARS 7 engine.

Threats move in microseconds; so should beauty, insight, and control. Welcome to PortalVII.



# Production-Grade React + TypeScript Code Base

PortalVII U-ARS 7 Unified Dashboard v0.1.0

Drop-in skeleton that boots, routes, themes, and renders the key pages/widgets described in the "Final Enterprise Layout."

❶ Create the directory tree exactly as shown, ❷ copy each file verbatim, ❸ `npm i && npm run dev`.

## 1 Folder Tree (after `npm create vite@latest portalvii-frontend -- --template react-ts`)

```
portalvii-frontend/
├── public/
│   └── portalvii-logo.svg
├── src/
│   ├── App.tsx
│   ├── index.tsx
│   ├── routes.tsx
│   ├── theme.ts
│   ├── layouts/
│   │   └── MainLayout.tsx
│   ├── components/
│   │   ├── Header.tsx
│   │   ├── Sidebar.tsx
│   │   ├── ContextRail.tsx
│   │   ├── MiniTabs.tsx
│   │   ├── WidgetCard.tsx
│   │   └── Loader.tsx
│   ├── pages/
│   │   ├── Overview.tsx
│   │   ├── Incidents.tsx
│   │   ├── Performance.tsx
│   │   ├── Ledger.tsx
│   │   └── AIAssist.tsx
│   └── services/
│       └── apiClient.ts
```



```
| vite.config.ts
| package.json
```

## 2 Core Files

### 2.1 src/theme.ts

```
import { createTheme, alpha } from '@mui/material/styles';

declare module '@mui/material/styles' {
  interface Palette {
    accent: Palette['primary'];
  }
  interface PaletteOptions {
    accent?: PaletteOptions['primary'];
  }
}

export const theme = createTheme({
  palette: {
    mode: 'light',
    primary: { main: '#1B263B' }, // deep blue
    secondary: { main: '#415A77' }, // steel blue
    accent: { main: '#FF7F50' }, // coral
    background: { default: '#F4F6F8', paper: 'FFFFFF' },
    text: { primary: '#0E1A24', secondary: alpha('#0E1A24', 0.65) },
  },
  typography: {
    fontFamily: `Inter, Roboto, sans-serif`,
    h1: { fontWeight: 700, letterSpacing: '-.03em' },
    h2: { fontWeight: 700, letterSpacing: '-.02em' },
    h3: { fontWeight: 700, letterSpacing: '-.02em' },
    body1: { fontWeight: 400, lineHeight: 1.6 },
  },
  shape: { borderRadius: 10 },
});
```

### 2.2 src/services/apiClient.ts

```
import axios from 'axios';

export const api = axios.create({
  baseURL: import.meta.env.VITE_API_BASE || '/api',
  withCredentials: true
});

// mTLS / JWT interceptors placeholder
api.interceptors.request.use(cfg => {
  // attach intent-token / JWT here
  return cfg;
});
```

## 2.3 src/index.tsx

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import { BrowserRouter } from 'react-router-dom';
import { ThemeProvider, CssBaseline } from '@mui/material';
import { QueryClient, QueryClientProvider } from '@tanstack/react-query';
import { theme } from './theme';
import App from './App';

const qc = new QueryClient();

ReactDOM.createRoot(document.getElementById('root')!).render(
  <React.StrictMode>
    <ThemeProvider theme={theme}>
      <CssBaseline />
      <QueryClientProvider client={qc}>
        <BrowserRouter>
          <App />
        </BrowserRouter>
      </QueryClientProvider>
    </ThemeProvider>
  </React.StrictMode>
);
```

## 2.4 src/App.tsx

```
import React from 'react';
import { Routes, Route, Navigate } from 'react-router-dom';
import MainLayout from './layouts/MainLayout';
import Overview from './pages/Overview';
import Incidents from './pages/Incidents';
import Performance from './pages/Performance';
import Ledger from './pages/Ledger';
import AIAssist from './pages/AIAssist';

export default function App() {
  return (
    <Routes>
      <Route element={MainLayout} />
      <Route path="/" element={Navigate to="/overview" replace} />
      <Route path="/overview" element={Overview} />
      <Route path="/incidents" element={Incidents} />
      <Route path="/performance" element={Performance} />
      <Route path="/ledger" element={Ledger} />
      <Route path="/ai" element={AIAssist} />
    </Routes>
  );
}
```

## 2.5 src/layouts/MainLayout.tsx

```
import { Outlet } from 'react-router-dom';
import { Box, Toolbar } from '@mui/material';
import Header from '../components/Header';
import Sidebar from '../components/Sidebar';
import ContextRail from '../components/ContextRail';
import MiniTabs from '../components/MiniTabs';

export default function MainLayout() {
  return (
    <Box sx={{ display: 'flex' }}>
      <Header />
      <Sidebar />
      <ContextRail />
      <Box component="main" sx={{ flexGrow: 1, p: 3, ml: { sm: '260px' } }}>
        <Toolbar />
        <MiniTabs />
        <Outlet />
      </Box>
    </Box>
  );
}
```

## 3 Navigation Components

### 3.1 src/components/Header.tsx

```
import { AppBar, Toolbar, IconButton, Typography, Switch } from '@mui/material';
import SearchIcon from '@mui/icons-material/Search';
import AccountCircle from '@mui/icons-material/AccountCircle';
import LightMode from '@mui/icons-material/LightMode';
import DarkMode from '@mui/icons-material/DarkMode';
import React from 'react';

export default function Header() {
  const [dark, setDark] = React.useState(false);
  return (
    <AppBar position="fixed" sx={{ zIndex: theme => theme.zIndex.drawer + 1 }}>
      <Toolbar>
        
        <Typography variant="h6" sx={{ flexGrow: 1 }}>
          U-ARS 7 Dashboard
        </Typography>
        <IconButton color="inherit"><SearchIcon /></IconButton>
        <Switch checked={dark} onChange={() => setDark(!dark)} icon={<LightMode />} checked={dark} />
        <IconButton color="inherit"><AccountCircle /></IconButton>
      </Toolbar>
    </AppBar>
  );
}
```

### 3.2 src/components/Sidebar.tsx

```
import { Drawer, List, ListItemButton, ListItemIcon, ListItemText, Tooltip } from '@mui/material';
import LayersIcon from '@mui/icons-material/Layers';
import SecurityIcon from '@mui/icons-material/Security';
import SettingsIcon from '@mui/icons-material/Settings';
import React from 'react';
import { useNavigate } from 'react-router-dom';

const menu = [
  { text: 'Layers', icon: <LayersIcon />, path: '/overview' },
  { text: 'Governance', icon: <SecurityIcon />, path: '/ledger' },
  { text: 'Settings', icon: <SettingsIcon />, path: '/settings' }
];

export default function Sidebar() {
  const navigate = useNavigate();
  return (
    <Drawer variant="permanent" sx={{ width: 260, [`& .MuiDrawer-paper`]: { width: 260 } }}>
      <List>
        {menu.map(m => (
          <Tooltip key={m.text} title={m.text} placement="right">
            <ListItemButton onClick={() => navigate(m.path)}>
              <ListItemIcon>{m.icon}</ListItemIcon>
              <ListItemText primary={m.text} />
            </ListItemButton>
          </Tooltip>
        ))}
      </List>
    </Drawer>
  );
}
```

### 3.3 src/components/ContextRail.tsx

```
import { Drawer, Box, Typography, ToggleButtonGroup, ToggleButton } from '@mui/material';

export default function ContextRail() {
  return (
    <Drawer
      variant="permanent"
      anchor="right"
      sx={{
        width: 220,
        [`& .MuiDrawer-paper`]: { width: 220, borderLeft: '1px solid #E0E0E0' }
      }}
    >
      <Box sx={{ p: 2 }}>
        <Typography variant="body2" sx={{ mb: 1 }}>Environment</Typography>
        <ToggleButtonGroup color="primary" exclusive size="small">
          <ToggleButton value="prod">Prod</ToggleButton>
          <ToggleButton value="stage">Staging</ToggleButton>
          <ToggleButton value="edge">Edge</ToggleButton>
        </ToggleButtonGroup>
      </Box>
    </Drawer>
  );
}
```

```

        <Typography variant="body2" sx={{ mt: 3, mb: 1 }}>Time Range</Typography>
        { /* link to time picker dialog */ }
        <Typography variant="caption">Last 24 h</Typography>
      </Box>
    </Drawer>
  );
}

```

### 3.4 src/components/MiniTabs.tsx

```

import { Tabs, Tab } from '@mui/material';
import { useLocation, useNavigate } from 'react-router-dom';

const mapping = ['/overview', '/incidents', '/performance', '/ledger', '/ai'];

export default function MiniTabs() {
  const { pathname } = useLocation();
  const navigate = useNavigate();
  const value = mapping.indexOf(pathname);
  return (
    <Tabs value={value} onChange={(_, v) => navigate(mapping[v])} sx={{ mb: 3 }}>
      <Tab label="Overview" />
      <Tab label="Incidents" />
      <Tab label="Performance" />
      <Tab label="Ledger" />
      <Tab label="AI Assist" />
    </Tabs>
  );
}

```

## 4 Shared Widget

src/components/WidgetCard.tsx

```

import { Paper, Box, Typography } from '@mui/material';
import React from 'react';

interface Props {
  title: string;
  icon: React.ReactNode;
  value: string | number;
  description: string;
  color?: string;
}

export default function WidgetCard({ title, icon, value, description, color }: Props) {
  return (
    <Paper elevation={3} sx={{ p: 3, display: 'flex', alignItems: 'center', minHeight: 120 }}>
      <Box sx={{ mr: 3, color: color || 'primary.main', fontSize: 48 }}>{icon}</Box>
      <Box>
        <Typography variant="h6" sx={{ fontWeight: 700 }}>{title}</Typography>

```

```

        <Typography variant="h4" sx={{ fontWeight: 900, mb: 0.5 }}>{value}</Typography>
        <Typography variant="body2" color="text.secondary">{description}</Typography>
      </Box>
    </Paper>
  );
}

```

## 5 Pages

### 5.1 src/pages/Overview.tsx

```

import { Grid, Typography } from '@mui/material';
import SecurityIcon from '@mui/icons-material/Security';
import DeviceHubIcon from '@mui/icons-material/DeviceHub';
import VpnKeyIcon from '@mui/icons-material/VpnKey';
import LockOpenIcon from '@mui/icons-material/LockOpen';
import WidgetCard from '../components/WidgetCard';

export default function Overview() {
  return (
    <>
      <Typography variant="h5" gutterBottom>Command Concourse</Typography>
      <Grid container spacing={3}>
        <Grid item xs={12} md={6} lg={3}>
          <WidgetCard title="Threat Detections" icon={<SecurityIcon />} value={0}
            description="No active threats" color="#36D1DC" />
        </Grid>
        <Grid item xs={12} md={6} lg={3}>
          <WidgetCard title="Registered Devices" icon={<DeviceHubIcon />} value={1}
            description="Device healthy" color="#FF7F50" />
        </Grid>
        <Grid item xs={12} md={6} lg={3}>
          <WidgetCard title="Intent Tokens" icon={<VpnKeyIcon />} value={3}
            description="Active tokens" color="#415A77" />
        </Grid>
        <Grid item xs={12} md={6} lg={3}>
          <WidgetCard title="Capsule Access" icon={<LockOpenIcon />} value="Granted"
            description="All capsules reachable" color="#232946" />
        </Grid>
      </Grid>
    </>
  );
}

```

### 5.2 src/pages/Incidents.tsx (skeleton)

```

export default function Incidents() {
  return <h2>Incidents Kanban coming soon...</h2>;
}

```

### 5.3 src/pages/Performance.tsx

```
export default function Performance() {  
  return <h2>Latency Lattice (flame-graph) placeholder</h2>;  
}
```

### 5.4 src/pages/Ledger.tsx

```
export default function Ledger() {  
  return <h2>Chain Explorer placeholder</h2>;  
}
```

### 5.5 src/pages/AIAssist.tsx

```
export default function AIAssist() {  
  return <h2>Guardian Copilot chat panel placeholder</h2>;  
}
```

## 6 Scripts & Dev Commands

Add to package.json:

```
{  
  "scripts": {  
    "dev": "vite",  
    "build": "vite build",  
    "preview": "vite preview",  
    "lint": "eslint \"src/**/*.{ts,tsx}\" --max-warnings=0",  
    "format": "prettier --write \"src/**/*.{ts,tsx,js,jsx,json,css,md}\"",  
  },  
  "dependencies": {  
    "@emotion/react": "^11.11.1",  
    "@emotion/styled": "^11.11.0",  
    "@mui/icons-material": "^5.15.19",  
    "@mui/material": "^5.15.19",  
    "@tanstack/react-query": "^5.19.2",  
    "@simplewebauthn/browser": "^7.5.0",  
    "axios": "^1.6.8",  
    "react-router-dom": "^6.23.1"  
  },  
  "devDependencies": {  
    "@types/node": "^20.11.17",  
    "@types/react": "^18.2.45",  
    "@types/react-dom": "^18.2.18",  
    "@vitejs/plugin-react": "^4.2.1",  
    "eslint": "^8.56.0",  
    "prettier": "^3.2.5",  
    "typescript": "^5.4.2",  
    "vite": "^5.2.0"  
  }  
}
```

```
}  
}
```

## 7 Run Locally

```
cd portalvii-frontend  
npm install  
npm run dev  
# open http://localhost:5173
```

## 8 Deploy to Kubernetes (Helm Snippet)

```
# charts/portalvii-frontend/values.yaml  
image:  
  repository: portalvii/frontend  
  tag: v0.1.0  
ingress:  
  enabled: true  
  className: "nginx"  
  hosts:  
    - host: portalvii.com  
      paths:  
        - path: /  
          pathType: Prefix  
  tls:  
    - secretName: portalvii-tls
```

Run:

```
helm upgrade --install portalvii-frontend charts/portalvii-frontend -n uars7-dev --create
```

## What's Included

- **Navigation tiers** with keyboard shortcuts (⌘K omni-search wired later).
- **Context rail** for time/environment, hot-swappable.
- **Mini-tabs** for Overview → AI Assist (→ cycles).
- **Overview page** showing 7-Layer ribbon + KPI widgets (live data wiring stubbed).
- **Theme tokens** reflecting Montfort-inspired colours.
- **Axios + React-Query** skeleton for mTLS-secured API calls.
- **CI-ready**—lint & format scripts, Vite for lightning builds.

This scaffold compiles today and matches the enterprise layout. Plug in real WebSocket feeds, Three.js XR Dome, and your U-ARS 7 APIs to reach GA.

✱



# PortalVII Front-End UI

## “Google Stitch” Prompt Pack for Rapid Visual Ideation

Use the following ready-made prompts inside your preferred *Google Stitch* (or any AI image-generation workspace) to produce high-fidelity UI mock-ups that match the **Montfort Aurora** theme defined earlier. After you have your approved visuals, send them back and I’ll translate the look-and-feel into complete React + MUI code.

### 1 Brand Foundations — Quick Reference

Token	Value	Usage
Primary Blue	#1B263B	App bar, headings, buttons
Steel Blue	#415A77	Sidebar, cards, form accents
Coral Accent	#FF7F50	CTAs, alerts, active chips
Sand Neutral	#BFA181	Dividers, secondary icons
Graphite 50	#F4F6F8	Page background
Font Stack	Inter / Roboto / sans-serif	All typography

### 2 Prompt Template (fill the *italics* parts)

“Full-screen enterprise dashboard UI for a cyber-security platform called PortalVII. Art direction: sleek, minimal, deep maritime blues with coral accent highlights. Include [\*layout section names\*], clean grid, data widgets with subtle glassmorphism, font set is Inter, 60-30-10 colour rule, light mode on Graphite 50 background. Perspective: front-on, no device frame, 4K resolution, Dribbble shot style.”

Replace [\*layout section names\*] with the panel or component list you need (e.g., “health ribbon, risk heat-map, ledger throughput card, four KPI tiles”).

### 3 Page-Level Prompts

Screen	Copy-paste Prompt
Login / WebAuthn	“Passwordless login screen, left split hero illustration of security key handshake, right split form with single ‘Authenticate’ button, PortalVII logo top-left, deep blue gradient header, glowing coral CTA, Inter font.”
Dashboard (Command Concourse)	“Seven-layer cyber-resilience dashboard: top health ribbon with colored layer badges, left two-thirds predictive risk heat-map in neon coral, right ledger throughput card, bottom row four KPI widgets, minimalistic sidebar, steel-blue tones, light glass cards.”
Device Enrollment	“Device management UI table with device avatars, posture badges, add device wizard stepper, maritime blue headers, coral ‘Enroll’ button, subtle success/toast notifications.”
Intent Token Vault	“Intent token management page, card grid view, each card shows purpose, scope, TTL countdown ring, coral revoke icon hover, Inter semi-bold headers, clean emptystate illustration if no tokens.”

Screen	Copy-paste Prompt
Capsule Access	"Self-protecting data capsule list, expandable rows revealing audit log timeline, AES-key iconography, sand neutral dividers, coral pill badges for policy conditions."
Admin Panel	"Security admin console panel, three-column layout: user roles list, policy YAML editor with syntax highlighting, real-time compliance score gauge donut, deep blue background with glass-looking panels."

## 4 Component-Focused Prompts

Component	Prompt Snippet
7-Layer Health Ribbon	"Horizontal status ribbon, seven rounded rectangles each a unique blue-to-green gradient, subtle pulse animation, micro-sparkline inside."
Risk Horizon Heat-Map	"Predictive risk heat-map widget, 24×7 grid, coral-to-green gradient, tooltip on hover, soft inner shadow."
Ledger Block Card	"Blockchain block summary card, SHA-3 hash displayed monospace, mini trust-hop diagram, steel blue glassmorphism."
Variant Divergence Gauge	"Semi-circular gauge, coral needle, green safe zone, dark navy background, thin white tick marks."
Time-Travel Scrubber	"Horizontal scrub bar with draggable thumb, ghosted timestamps, light glow trail, deep blue track."

## 5 Illustrations & Hero Imagery

Use-Case	Prompt
On-boarding Empty State	"Isometric satellite dish scanning cyber constellation, blue/steel palette, coral signal pulses, modern flat style."
Error / 404	"Broken hologram shard floating in dark ocean, minimal line art, coral warning icon overlay."
Success Toast Icon	"Tiny shield with checkmark, gradient steel blue to teal, soft drop shadow, 32×32."

## 6 Micro-Interaction & Motion Prompts

Element	Prompt
KPI Card Hover	"Short video loop: glass KPI card scales 1.00→1.03, coral shadow fade, 140 ms cubic easing, UI context."
Sidebar Expand	"Animated gif: sidebar width morph 80→260 px, icons fade labels in, maritime blue background."
Notification Pulse	"3-frame Lottie: coral dot opacity 0.3→1.0→0.3, 1.2 s loop, clean vector."

## 7 Tips for Better Results

1. **Explicit Resolution** – request “4K PNG” or exact pixel size to ease redlining.
2. **Grid & Spacing** – mention “8 px spacing scale” to keep alignment consistent.
3. **Variant Shots** – ask for “light & dark mode variants side-by-side.”
4. **Remove Device Frames** – unless you need a marketing mock-up, stay UI-only for precise hand-off.
5. **One Visual per Prompt** – reduces clutter and speeds iteration.

## 8 Next Step After Visuals

1. **Export** each approved screen to PNG or Figma.
2. **Share** the assets (or a Figma link) here.
3. I'll deliver pixel-perfect React + MUI code that matches your chosen visuals, complete with theme tokens, component library, routing, and API wiring.

*Happy stitching—looking forward to seeing the designs!*

✱

## Google Stitch Prompt

“Full-screen, 4K enterprise cybersecurity dashboard UI for PortalVII (U-ARS 7).  
Visual style: sleek maritime deep blues (#1B263B primary, #415A77 secondary) with coral accents.  
Font: Inter / Roboto, semi-bold headings, light body.

Layout:

- Global top bar with logo left, omni-search center, dark/light toggle, user avatar right
- Collapsible steel-blue sidebar (260 px) with icons and labels: Overview, Incidents, Performance
- Context rail on right (220 px) for time-range and environment chips.

Main content – “Command Concourse”:

1. Hero strip across top: 7-Layer Health Ribbon with color-coded status (CADS, M-SES, SHE)
2. Left two-thirds: Risk Horizon heat-map, neon coral-to-teal gradient, 90-second prediction
3. Right one-third: Ledger Throughput card showing blocks/sec, hash latency, peer status.
4. Bottom 4-tile grid:
  - Top 5 intents by volume bar chart
  - Variant divergence gauge (semi-circle)
  - Snapshot delta timer (digital)
  - Capsule revocations ticker (scrolling)

Secondary pages:

- Incidents – Kanban columns: New Signals, Auto-Mitigated, Needs Review, Closed; cards with details
- Performance – Vertical flame-graph stacking ADCF, TRDN, QVDM, ILECG layers; hover details
- Ledger – Blockchain explorer list with SHA-3 hash, diff viewer, trust-hop graph.
- AI Assist – Right-dock chat panel, GPT-style bubbles, code and chart snippets.

Cross-cutting elements: intent token tooltip (hover), time-travel scrubber slider (24 h),  
Glassmorphism card backgrounds (8 px blur), 10 px border-radius, subtle drop shadow.

Motion: KPI card hover scale 1→1.03 (140 ms), sidebar expand 80→260 px (180 ms), notification  
Render front-on, no device frame, high-contrast, no lorem ipsum, light mode.”

✱

## Enhanced Prompt for Google Stitch

*Generate a full-screen UI mock-up for the PortalVII U-ARS 7 dashboard, incorporating richer detail and visual nuance than the previous version.*

“Enterprise-grade cybersecurity dashboard for ‘PortalVII – U-ARS 7’

Design language: premium maritime blues (#1B263B, #415A77) with coral accent (#FF7F50) or

Typography: Inter SemiBold for headings, Inter Regular for body.

Overall layout (desktop, 1920 × 1080):

- Fixed top bar (72 px) – left-aligned logo, centered omni-search, right-aligned dark/light
- Steel-blue sidebar (260 px) – icons + labels: Overview, Incidents, Performance, Ledger,
- Context rail (right, 220 px) – time-range selector, environment chips (Prod, Stage, Edg
- Main canvas: ‘Command Concourse’ view.

Main canvas widgets:

1. 7-Layer Health Ribbon (full-width, 48 px) – seven rounded pills with live status color
2. Two-thirds left: ‘Risk Horizon’ heat-map (900 × 440) – 90-second predictive grid, coral
3. One-third right: ‘Ledger Throughput’ card (420 × 440) – blocks/sec, hash latency, peep
4. Bottom quad-grid (4 × 380 × 240):
  - Top-5 intents bar chart (coral bars)
  - Variant divergence semi-gauge (green-to-coral arc)
  - Snapshot delta timer (digital flip, monospace)
  - Capsule revocations ticker (scrolling list with badge chips)

Secondary screens:

- Incidents – 4-column kanban (New, Auto-Mitigated, Needs Review, Closed); cards show tin
- Performance – vertical flame-graph stacking ADCF/TRDN/QVDM/ILECG layers; hover reveals
- Ledger – blockchain explorer table with SHA-3 hashes, diff viewer, ‘Trust-Hop’ endorsem
- AI Assist – right-dock chat panel, GPT-style bubbles, slash-command shortcuts.

Micro-interactions & motion:

- KPI card hover: scale 1→1.03, coral shadow deepen, 140 ms cubic ease.
- Sidebar collapse/expand: width 260→80 px, 180 ms ease-out.
- Notification pulse: coral dot opacity loop 0.3→1→0.3, 1.2 s linear.
- Time-travel scrub bar: drag thumb, ghosted timestamps trail.

Visual polish: subtle glassmorphism on cards (6-px blur, 10 px radius); 8-px base grid; c

Deliverables: high-fidelity 4K PNG, light-mode variant, no device frame, front-on perspec

✱✱

## Ultra-Detailed Prompt for Google Stitch

*One copy-paste block that instructs the AI designer to generate a complete PortalVII U-ARS 7 dashboard mock-up with pixel-level specificity, brand personality, and advanced micro-interactions.*

“Design a full-screen **enterprise cybersecurity dashboard** for **PortalVII – U-ARS 7**.  
Render in **4K (3840 × 2160)**, light-mode.

Visual language: premium maritime palette – primary **##1B263B**, secondary **##415A77**, and tertiary **##F0F0F0**.  
Typography: **Inter** semibold headings, **Inter** regular body, 8-pt modular scale, 1.4 line height.  
Iconography: 24 px stroke icons, rounded ends, consistent optical weight.

#### LAYOUT GRID

- 12-column, 96 px lateral gutters, 24 px internal gutters.
- 72 px top bar height, 260 px left sidebar (collapsible to 80 px), 220 px right context rail.
- Main canvas padding 48 px top, 32 px sides/bottom.
- 8 pt spacing rule for all paddings, gaps, and corner radii.

#### TOP BAR (fixed, glassy)

1. Left: white SVG logo 36 px, click opens mega menu.
2. Center: cmd + K omni-search, pill style, subtle inset shadow.
3. Right: dark/light switch (sun/moon morph animation), notifications bell with pulse, user profile.

#### SIDEBAR (steel-blue)

- Sections: Overview, Incidents, Performance, Ledger, AI Assist.
- Active item: coral accent bar 4 px wide, icon + label fade in/out on collapse.
- Hover: icon tilt 2°, label opacity 70 → 100% in 120 ms.

#### CONTEXT RAIL (right)

- Time-range chip group (last 1 h, 24 h, custom calendar pop-over).
- Environment toggle (Prod / Stage / Edge) with traffic-light dots.
- Live filter tags stack vertically, removable with backspace.

#### PAGE – COMMAND CONCOURSE

1. **Hero Health Ribbon**: seven rounded pills (CADS, M-SES, SHEL, ILECG, QVDM, TRDN, ADCF)
  - Each pill 98 × 32 px, gradient fill per status (green → yellow → red), micro-sparkline.
2. **Risk Horizon Heat-Map** (left ⅔, 900 × 440 px): 24 × 60 grid, coral-to-teal diverging color scale.
3. **Ledger Throughput Card** (right ⅓, 420 × 440 px): donut for blocks/sec, latency bar chart.
4. **Bottom Quad Grid** (each 380 × 240 px):
  - Top-5 intents bar chart (coral bars, blue background grid).
  - Variant divergence semi-gauge (green safe, coral danger, animated needle).
  - Snapshot delta flip-clock timer (monospace, shadowed).
  - Capsule revocations ticker (vertical scroll, pill badges).

#### PAGE – INCIDENTS

- 4-column Kanban (New, Auto-Mitigated, Needs Review, Closed).
- Cards: 320 × 180 px, corner radius 12 px, shadow depth 2.
- Inside card: timestamp, variant ID, mini timeline sparkline, “Re-hydrate” button with undo.
- Column headers sticky, show count badges, drag-and-drop enabled.

#### PAGE – PERFORMANCE

- Vertical stacked flame graph: ADCF → TRDN → QVDM → ILECG; color ramp navy → sea-foam.
- Hover cross-hair shows syscall, network hop, FPGA compile slice.
- Side panel toggles CPU, memory, network overlays.

#### PAGE – LEDGER

- Left: block list table (hash, height, age, endorsements) with zebra rows.
- Right: diff viewer – snapshot lineage or genome delta in split view.
- Trust-hop radial graph, nodes pulse on hover; click opens org detail fly-out.

#### PAGE – AI ASSIST

- Right-docked chat (35 % width), conversations in speech bubbles with subtle gradient.
- Slash-command palette (/explain, /report, /patch).
- Code and chart snippets auto-render with syntax / mini-plots.

#### GLOBAL MICRO-INTERACTIONS

- Card hover: scale 1 → 1.03, shadow deepen, 140 ms cubic-bezier(.33,.99,.68,1).
- Sidebar collapse: width 260 → 80 px, icon wobble 4 °, label fade, 180 ms ease-out.
- Notification pulse: coral dot opacity 0.3 → 1 → 0.3, 1.2 s linear infinite.
- Time-scrub slider: ghost timestamps trail 8 frames, blur 4 px.

#### MOTION GUIDELINES

- Prefer 120 fps / 8.33 ms frame budget.
- Reduce motion: honor prefers-reduced-motion, disable scale / slide, retain color change

#### ACCESSIBILITY

- Color contrast  $\geq 4.5:1$  for text,  $\geq 3:1$  for icons.
- All interactive elements 44 × 44 px min.
- Keyboard nav: logical tabindex, visible focus ring #FF7F50 at 2 px.

#### OUTPUTS

- Deliver **two 4K PNGs**: main dashboard and incidents page.
- Include **light & dark mode** variants side-by-side.
- Export primary components (health ribbon, heat-map, Kanban card) as isolated **SVG** assets.
- Provide **Figma** or layered **PSD** with named groups, 8-pt grid visible.

#### STYLE INSTRUCTIONS

- No mock device frames; pure UI.
- Avoid lorem ipsum – use real labels (“Threat Detections”, “Variant Divergence”).
- Use point-in-time data (e.g., “Blocks 321 / sec”, “Risk 0.78”).
- Keep visual noise low: plenty of white space, 8 pt grid, glassmorphism cards with 6 px

#### DELIVERABLE STANDARD

- Consistent shadows (Y = 2 px, blur 8 px, rgba(0,0,0,0.08)).
- Corner radius system: 4 / 8 / 12 / 14 px only.
- Color tokens: defined as CSS variables (e.g., --pvii-primary, --pvii-accent).
- Embed accessibility notes on layers (alt text, aria-label suggestions).

END OF PROMPT”

#### How to use:

1. Copy the entire block into Google Stitch’s prompt field.
2. Select “4K, PNG, light + dark variants” in output settings.
3. Review the generated screens, approve or annotate, then export assets for your React + MUI implementation.

✱

## just only give prompt of dashbaord just dahabord

“Design a full-screen, 4K cybersecurity dashboard UI for PortalVII – U-ARS 7.

Visual language: deep maritime blues (#1B263B primary, #415A77 secondary) on soft graphiti  
Typography: Inter SemiBold headings, Inter Regular body, 8 pt spacing grid.

Interface layout (desktop 1920×1080):

- Fixed top bar (72 px) with logo, omni-search, dark/light toggle, notifications, avatar.
- Collapsible steel-blue sidebar (260 px → 80 px) with icons & labels: Overview, Inciden

- Right context rail (220 px) for time-range chips and prod/stage/edge environment select

Main canvas – “Command Concourse”:

1. Hero health ribbon (full-width, 7 rounded pills) showing CADS, M-SES, SHEL, ILECG, QVI
2. Left two-thirds: “Risk Horizon” heat-map (coral→teal gradient) predicting breaches 90
3. Right one-third: “Ledger Throughput” card with blocks/sec donut, hash latency bar, pee
4. Bottom grid (4× tiles 380×240 px):
  - Top-5 intents bar chart (coral bars).
  - Variant divergence semi-gauge (green→coral).
  - Snapshot delta flip-clock.
  - Capsule revocations ticker (scrolling list).

Secondary screens to include:

- Incidents Kanban (New → Auto-Mitigated → Needs Review → Closed) with timeline mini-cl
- Performance vertical flame-graph stacking ADCF/TRDN/QVDM/ILECG layers, hover reveals sy
- Ledger explorer with SHA-3 hashes, diff viewer, trust-hop radial graph.
- AI Assist chat panel docked right 35 % width, slash-command palette.

Micro-interactions & motion:

- KPI card hover scale 1 → 1.03 (140 ms cubic-ease).
- Sidebar expand 80 → 260 px (180 ms ease-out).
- Notification pulse opacity 0.3 → 1 → 0.3 (1.2 s loop).
- Time-scrub slider ghost timestamp trail.

Style: subtle glassmorphism cards (6 px blur, 10 px radius), consistent 8 pt grid, no lo



## PortalVII U-ARS 7 Theme Design Prompts

Based on the comprehensive dashboard design we've developed for your PortalVII U-ARS 7 project, here are detailed prompts for creating the visual theme and design elements:

### Brand Identity Prompts

#### Logo & Brand Mark

"Design a premium cybersecurity logo for 'PortalVII' - sleek, modern wordmark with geomet

#### Color Palette Visualization

"Create a comprehensive color palette board for PortalVII cybersecurity platform. Primary

# Dashboard Interface Prompts

## Main Dashboard Layout

"Design a premium cybersecurity dashboard interface for PortalVII U-ARS 7. Layout: top navigation bar, left sidebar, main content area with 3 primary widgets: Health Ribbon, Risk Horizon, and Incident Board. Footer with system status and logs."

## Seven-Layer Health Ribbon

"Create a horizontal status ribbon showing 7 cybersecurity layers: CADS, M-SES, SHEL, ILE, N-SEC, C-OPS, and D-INT. Each layer has a color-coded status indicator (Green, Yellow, Red) and a percentage value."

## Risk Horizon Heat-Map

"Design a predictive risk heat-map widget for cybersecurity dashboard. 24x60 grid showing risk levels (Low, Medium, High) across time (Horizon) and threat types (Signals, Incidents, Vulnerabilities)."

## Component Design Prompts

### KPI Widget Cards

"Design glassmorphism KPI cards for cybersecurity metrics. Each card: icon (48px), title, value, and trend (up/down). Metrics include: Active Threats, Mitigated Incidents, System Uptime, and Data Breach Attempts."

### Navigation Sidebar

"Create a collapsible navigation sidebar for cybersecurity platform. Width: 260px expanded. Items: Home, Health, Risk, Incidents, Settings, Audit, Help. Active item highlighted with blue background."

### Incident Kanban Board

"Design a Kanban board for security incidents. 4 columns: New Signals, Auto-Mitigated, New Incidents, Resolved. Cards show incident ID, severity, and status. Drag-and-drop functionality."

## Advanced Visualization Prompts

### Holographic Twin Dome (3D Concept)

"Create a futuristic 3D holographic visualization concept for cybersecurity layers. Seven layers represented as concentric domes: CADS, M-SES, SHEL, ILE, N-SEC, C-OPS, D-INT. Each dome has a unique color and data visualization."

### Performance Flame Graph

"Design a vertical stacked flame graph for system performance. Layers: ADCF, TRDN, QVDM, and others. Shows time spent in different system components. Color-coded by component."



## Ledger Chain Explorer

"Create a blockchain explorer interface for audit ledger. Block list with SHA-3 hashes, transaction details, and a search bar for specific blocks or transactions."

## Iconography Prompts

### Security Layer Icons

"Design a cohesive icon set for 7 cybersecurity layers: CADS (micro-cells), M-SES (evolutionary security), and others. Icons should be simple, scalable, and represent different security concepts."

### Status & Action Icons

"Create status and action icons for cybersecurity dashboard: health indicators (green, yellow, red), alerts (bell), and actions (refresh, reset, delete)."

## Typography & Layout Prompts

### Typography System

"Design a typography system for PortalVII cybersecurity platform. Primary font: Inter (headings), sans-serif (body text). Define font weights and sizes for different UI elements."

### Grid System Visualization

"Create a responsive grid system guide for PortalVII dashboard. 12-column layout, 24px gutter. Show how the grid adapts to different screen sizes (desktop, tablet, mobile)."

## Motion & Interaction Prompts

### Micro-Interactions

"Design subtle micro-interactions for cybersecurity dashboard: card hover effects (scale, shadow), button click animations, and loading state indicators."

### Loading & Empty States

"Create loading animations and empty state illustrations for cybersecurity platform. Loading states should be clear and indicate progress. Empty states should provide guidance on how to add data or filter results."

## Brand Application Prompts

## Marketing Materials

"Design marketing materials for PortalVII cybersecurity platform. Hero banners, feature cards, social media assets, all in brand colors and fonts."

## Documentation Theme

"Create a documentation theme for PortalVII technical docs. Clean, readable layout with clear navigation, consistent styling, and accessible content structure."

These prompts will help you create a cohesive, professional visual identity for your PortalVII U-ARS 7 cybersecurity platform that aligns with the comprehensive dashboard design we've developed.



# PortalVII U-ARS 7 Dashboard Layout Prompt

## Comprehensive Design Specification for Enterprise Cybersecurity Command Center

Based on your U-ARS 7 project architecture and the PortalVII brand identity, here's a detailed prompt for creating a world-class cybersecurity dashboard that rivals Splunk, Datadog, and other enterprise platforms while incorporating your innovative seven-layer defense system.

### Core Design Prompt

Design a premium enterprise cybersecurity dashboard for PortalVII U-ARS 7 platform. Visual language: Deep maritime blues (#1B263B primary, #415A77 secondary) with coral accents. Typography: Inter SemiBold for headings, Inter Regular for body text, Fira Code for code/logs. Layout philosophy: Information density with breathing room, enterprise-grade precision, and intuitive navigation.

NAVIGATION ARCHITECTURE:

- Global header (72px): PortalVII logo, omni-search (⌘+K), environment selector, dark/light mode toggle
- Primary sidebar (280px): Seven-layer navigation (CADS→ADCF), Governance, Compliance, Settings, Reports, Integrations, Help
- Context rail (240px right): Time-range controls, environment chips (Prod/Stage/Edge), 1-click filters
- Mini-tab bar: Overview, Incidents, Performance, Ledger, AI Assist with keyboard shortcuts

MAIN DASHBOARD - "COMMAND CONCOURSE":

Hero Section (full-width, 120px):

- Seven-Layer Health Ribbon: Horizontal status bar showing CADS, M-SES, SHEL, ILECG, QVDM, etc.
- Each layer as rounded pill (140x32px) with color-coded status (green/yellow/red), micro-trends
- Real-time pulse animation for active layers, subtle glow effects

Primary Grid (3-column layout):

Left Column (60% width):

- "Risk Horizon" predictive heat-map (900x440px): 90-second quantum-solver breach forecast
- Color gradient: coral-to-teal diverging scale, interactive hover with threat details
- Time-travel scrubber at bottom for historical analysis

Right Column (40% width):

- "Ledger Throughput" card (420x440px): Hyperledger Fabric metrics
- Blocks/second donut chart, hash latency bar graph, peer status indicators

- Live transaction stream with cryptographic verification status

Bottom Grid (4x equal tiles, 380x240px each):

- Top 5 Intent Tokens: Bar chart with coral bars, usage frequency, expiration countdown
- Variant Divergence Gauge: Semi-circular gauge showing QVDM consensus health
- Snapshot Delta Timer: Digital flip-clock showing TRDN rollback readiness
- Capsule Revocations: Scrolling ticker with ADCF policy violations and auto-actions

SECONDARY PAGES:

INCIDENTS PAGE - "Autonomous SOC":

- 4-column Kanban layout: "New Signals" → "Auto-Mitigated" → "Needs Review" → "Closed"
- Card design (320x180px): Threat severity badge, timeline mini-chart, affected layer icon
- "Re-hydrate" button for sandbox replay, evidence bundle download
- Drag-and-drop between columns with smooth animations

PERFORMANCE PAGE - "Latency Lattice":

- Vertical flame-graph: Stacked layers (ADCF→TRDN→QVDM→ILECG) with color ramp navy→seafoam
- Interactive hover cross-hairs revealing syscall details, network hops, FPGA compile time
- Side panel toggles for CPU/memory/network overlays
- Performance SLA indicators with red-line thresholds

LEDGER PAGE - "Chain Explorer":

- Left panel: Block list table with SHA-3 hashes, timestamps, endorsements
- Right panel: Diff viewer showing snapshot lineage or genome delta changes
- Trust-hop radial graph with pulsing nodes, click for organization details
- Export functionality for compliance reports

AI ASSIST PAGE - "Guardian Copilot":

- Right-docked chat panel (35% width) with GPT-style conversation bubbles
- Slash-command palette (/explain, /report, /patch, /rollback)
- Code snippets and mini-charts auto-render within responses
- Natural language query examples and suggested prompts

CROSS-CUTTING UX ELEMENTS:

- Intent Token Tooltips: Hover any user action → purpose, scope, TTL, ZK-proof hash
- Time-Travel Scrubber: Drag to any point in 24h history → UI rewinds via TRDN snapshots
- Genome Waterfall: Live vs. culled service mutations (green=deployed, grey=retired)
- Notification System: Toast messages for critical events, persistent alerts for ongoing

MICRO-INTERACTIONS:

- Card hover: Scale 1→1.03, shadow deepen, 140ms cubic-bezier(.33,.99,.68,1)
- Sidebar expand: Width 280→80px collapse, icon wobble 4°, 180ms ease-out
- Data refresh: Subtle pulse on updated metrics, smooth number transitions
- Loading states: Skeleton screens with shimmer effects, progress indicators

RESPONSIVE BEHAVIOR:

- Desktop (≥1200px): Full layout with all panels visible
- Tablet (768-1199px): Collapsible sidebar, context rail becomes overlay
- Mobile (≤767px): Hamburger navigation, single-column layout, swipe gestures

ACCESSIBILITY FEATURES:

- WCAG 2.1 AA compliance: 4.5:1 color contrast minimum
- Keyboard navigation: All interactive elements accessible via Tab/Arrow keys
- Screen reader support: Proper ARIA labels, semantic HTML structure
- Motion reduction: Honor prefers-reduced-motion, disable animations

#### ADVANCED FEATURES:

- WebGL "Holographic Twin Dome": 3D visualization of seven layers as concentric rings
- Real-time collaboration: Multiple operators can view same dashboard with cursor sharing
- Custom widget builder: Drag-and-drop interface for creating personalized views
- Export capabilities: PDF reports, CSV data, PNG screenshots with annotations

#### BRANDING INTEGRATION:

- PortalVII logo prominence in header with subtle animation on load
- Consistent use of maritime color palette throughout interface
- Professional typography hierarchy with clear information architecture
- Enterprise-grade polish with attention to pixel-perfect details

#### OUTPUT SPECIFICATIONS:

- Deliver 4K mockups (3840x2160) for desktop layouts
- Include mobile responsive variants (375x812, 768x1024)
- Provide component library with consistent spacing (8pt grid)
- Export assets as organized Figma file with named layers and components
- Include dark mode variants for all major screens
- Specify exact color values, typography scales, and spacing measurements

## Key Dashboard Components to Include

### 1. Seven-Layer Status Visualization

- **CADS:** Micro-cell pool status, intent token processing rate
- **M-SES:** Code mutation frequency, fitness scores trending
- **SHEL:** FPGA utilization, shard synthesis latency
- **ILECG:** Bubble spawn rate, ephemeral VM health
- **QVDM:** Variant consensus accuracy, outlier detection
- **TRDN:** Snapshot frequency, rollback readiness
- **ADCF:** Capsule access logs, revocation events

### 2. Real-Time Threat Intelligence

- Risk horizon heat-map with 90-second prediction window
- Attack vector analysis with quantum-assisted forecasting
- Threat actor attribution and campaign tracking
- Zero-day vulnerability impact assessment

### 3. Operational Metrics

- System performance across all seven layers
- Resource utilization and capacity planning
- SLA compliance and availability metrics
- Cost optimization recommendations

4. Compliance Dashboard

- Regulatory framework mapping (PCI-DSS, GDPR, HIPAA)
- Audit trail completeness and integrity verification
- Policy compliance scoring and gap analysis
- Automated compliance reporting generation

5. Administrative Controls

- User and role management with RBAC visualization
- Policy configuration and deployment tracking
- System configuration drift detection
- Backup and disaster recovery status

This comprehensive layout ensures your PortalVII dashboard becomes the definitive interface for managing enterprise cybersecurity through the revolutionary U-ARS 7 platform, combining cutting-edge technology with intuitive user experience design.



CADS (Convergent Autonomic Defense Sphere) - Complete Page Design Specification

Overview Dashboard Section

Hero Status Panel

- Real-time CADS Status Indicator
  - Large circular status badge (Green/Yellow/Red) with pulsing animation
  - Current operational state: "ACTIVE", "LEARNING", "DEFENDING", "MAINTENANCE"
  - Uptime counter with millisecond precision
  - Last security event timestamp

Key Performance Metrics Grid

Metric	Current Value	Target	Trend
Micro-cell Launch Time	12ms	≤15ms	↗
Active Micro-cells	847	-	↑
Intent Tokens Processed	15,432/min	-	↗
Threat Neutralizations	23 (24h)	-	↓
Genome Fitness Score	94.7/100	>90	↗

# Micro-Cell Management Console

## Live Micro-Cell Pool

- **Pool Status Visualization**
  - Interactive honeycomb grid showing 8 pre-warmed cells
  - Color-coded cells: Available (Blue), Active (Green), Terminating (Orange)
  - Hover tooltips showing cell ID, TTL remaining, resource usage
  - Real-time spawn/destroy animations

## Cell Lifecycle Monitor

- **Active Cells Table**
  - Cell ID, Intent Token Hash, Purpose, TTL Countdown, Resource Usage
  - Sortable columns with real-time updates
  - "Terminate Early" action buttons for emergency shutdown
  - Export cell logs functionality

## Cell Performance Analytics

- **Execution Time Distribution Chart**
  - Histogram showing cell execution times (0-30s range)
  - Average, P95, P99 latency indicators
  - Comparison with previous 24h/7d periods

## Intent Token Management

### Token Validation Dashboard

- **FIDO2 Authentication Status**
  - Connected security keys list with device health
  - Recent authentication events timeline
  - Failed authentication attempts with geo-location

## Zero-Knowledge Proof Verification

- **ZKP Status Panel**
  - Proof generation time metrics
  - Verification success rate (target: >99.9%)
  - Cryptographic strength indicators
  - Token scope validation results

## Purpose-Bound Access Control

- **Intent Categories Breakdown**

- Pie chart showing distribution: Read (45%), Write (25%), Execute (20%), Admin (10%)
- Recent intent patterns and anomaly detection
- Privilege escalation attempt alerts

## WebAssembly Runtime Engine

### WASI Compliance Monitor

- **Runtime Health Metrics**

- Wasmtime engine version and status
- Memory isolation effectiveness score
- Sandbox escape attempt detection
- Module verification success rate

## Code Forge Integration

- **Binary Diversity Status**

- Current variant count in deployment pool
- Compiler flag randomization effectiveness
- ASLR entropy measurements
- Build forge cycle timing

## Performance Optimization

- **Resource Utilization Charts**

- CPU usage per micro-cell (real-time line chart)
- Memory allocation patterns
- I/O throughput metrics
- Garbage collection impact analysis

## Autonomous Policy Genome

### Machine Learning Model Status

- **Genome Evolution Dashboard**

- Current generation number
- Fitness score distribution across variants
- Mutation rate and selection pressure

- Learning convergence indicators

## **Policy Adaptation Engine**

- **Real-time Policy Updates**

- Recent policy changes with timestamps
- A/B testing results for new policies
- Rollback capabilities and version history
- Impact assessment on security posture

## **Behavioral Analysis**

- **Traffic Pattern Recognition**

- Normal vs. anomalous request patterns
- Seasonal trend detection
- User behavior clustering results
- Predictive threat modeling outputs

## **Forensic Micro-Ledgers**

## **Audit Trail Visualization**

- **Merkle Tree Structure**

- Interactive tree diagram showing syscall hashing
- Tamper-evident verification status
- Storage optimization metrics (24h auto-expiry)
- Incident-linked log preservation

## **Compliance Reporting**

- **Regulatory Mapping**

- PCI-DSS compliance score with detailed breakdown
- GDPR data minimization effectiveness
- HIPAA audit trail completeness
- SOX financial control validation

## **Incident Response Integration**

- **Security Event Correlation**

- SIEM integration status
- Alert escalation workflows



- Forensic evidence collection automation
- Chain of custody documentation

## **Threat Intelligence Integration**

### **Real-time Threat Feed**

- **Global Threat Landscape**
  - CVE monitoring and auto-exclusion from forge
  - Threat actor attribution data
  - Attack pattern recognition
  - Zero-day vulnerability assessment

### **Adaptive Defense Mechanisms**

- **Response Automation**
  - Automatic variant regeneration triggers
  - Threat-based policy tightening
  - Emergency lockdown procedures
  - Recovery time objectives (RTO) tracking

## **System Health & Monitoring**

### **Infrastructure Status**

- **Kubernetes Cluster Health**
  - Node status and resource availability
  - Pod deployment success rates
  - Service mesh (Istio) connectivity
  - Network policy enforcement

### **Performance Benchmarking**

- **SLA Compliance Tracking**
  - 99.99% availability target monitoring
  - Response time percentiles
  - Error rate thresholds
  - Capacity planning projections

## **Alerting & Notifications**

- **Multi-channel Alert System**
  - Slack/Teams integration for real-time alerts
  - Email notifications for policy violations
  - SMS alerts for critical security events
  - Dashboard alert acknowledgment system

## **Configuration Management**

### **Environment Controls**

- **Multi-environment Dashboard**
  - Production, Staging, Development status
  - Configuration drift detection
  - Deployment pipeline integration
  - Feature flag management

## **Security Hardening**

- **Hardening Checklist**
  - CIS Kubernetes benchmark compliance
  - Pod security standards enforcement
  - Network segmentation validation
  - Certificate management status

## **Advanced Analytics**

### **Predictive Security Modeling**

- **Risk Forecasting**
  - 90-second breach probability calculation
  - Attack path simulation results
  - Resource allocation optimization
  - Capacity planning recommendations

## **Business Impact Analysis**

- **Cost-Benefit Metrics**
  - Security investment ROI calculation
  - Prevented breach cost estimation

- Operational efficiency gains
- Compliance cost reduction

## **Integration Hub**

### **API Gateway Status**

- **External Integrations**
  - Third-party security tool connections
  - Data export/import capabilities
  - Webhook configuration management
  - Rate limiting and throttling controls

### **Development Tools**

- **Developer Experience**
  - SDK documentation links
  - API testing sandbox
  - Integration guides and tutorials
  - Support ticket system integration

### **Emergency Controls**

#### **Incident Response Panel**

- **Emergency Actions**
  - Global system lockdown button
  - Mass micro-cell termination
  - Policy rollback to last known good
  - Forensic mode activation

### **Recovery Procedures**

- **Disaster Recovery**
  - Backup and restore status
  - Failover mechanism testing
  - Recovery point objectives (RPO)
  - Business continuity planning

This comprehensive CADS page design ensures operators have complete visibility and control over every aspect of the Convergent Autonomic Defense Sphere, from micro-cell lifecycle management to advanced threat prediction and response automation.



1. This-is-our-plan-U-ARS-7-help-me-building-this-fro-2.pdf
2. paste-3.txt
3. Cause-of-the-Error.pdf
4. <https://www.cssdesignawards.com/sites/montfort/47521/>
5. <https://www.awwwards.com/sites/montfort>
6. <https://mont-fort.com/what-we-do/>
7. <https://www.hostinger.com/tutorials/website-color-schemes>
8. <https://www.fontfabric.com/fonts/mont/>
9. <https://ch.linkedin.com/company/mont-fort>