**MCPX‑KendoBridge — UI/UX Specification (KendoReact Fluent v12 + ThemeBuilder)**

**Document:** docs/08\_ui\_ux.docx  
**Project Code:** MCPX‑KendoBridge  
**Version:** 2.0.0 (Migration to Kendo)  
**Last Updated:** 2025‑09‑27  
**Document Owner:** DoSE (Accountable) — DocFactory (Responsible) — T‑Arch & SecLead (Consulted)

**Purpose.** Define an **audit‑ready** UI/UX standard for the Admin Portal using **KendoReact (Fluent v12)** themed via **ThemeBuilder** with tokens exported from **Figma Make**. This doc governs information architecture, theming, accessibility, streaming behaviors (SSE), non‑secret config display, performance budgets, and CI/CD gates—aligned with Technijian SDLC guardrails: **GitHub‑first**, four environments (**Alpha → Beta → RTM → Prod**), **No‑Hard‑Coding**, **Stored‑procedure‑only** DB access, **add‑only** migrations, and **secrets only** in environment stores.

**DB COMPLIANCE (UI impact):** The UI must not embed dynamic values. All dynamic behavior (timeouts, keep‑alive cadence, allowed origins, child cmd/args) comes from the API which reads **non‑secret** config from SQL Server via **stored procedures** (sp\_Config\_\*, sp\_Feature\_IsEnabled, sp\_Lookup\_Get). **Secrets (SQL connections, Telerik license) are never shown or stored in UI or DB**—they live in **GitHub Environments**.

**1) Scope & Non‑Goals**

**In scope (Admin Portal, read‑only ops):**

* Dashboards (health/readiness); Config viewer (non‑secret values only); Sessions (SSE activity); Access Control visibility (Origin allow‑list).
* Theming pipeline (**Figma Make → ThemeBuilder → Kendo Fluent v12**).
* Accessibility (WCAG 2.2 AA), localization scaffold, streaming UX, observability hooks.

**Out of scope:** Editing of server config or feature flags; displaying any secrets; administrative write actions.

**2) Technology & Theming Pipeline**

**2.1 Stack**

* **React 18** + **KendoReact** (Fluent v12 theme).
* **ThemeBuilder overrides** generated from **Figma Make** tokens.
* **Routing:** React Router.
* **Types & contracts:** Types auto‑generated from **OpenAPI 3.1** (/api/openapi/mcp-proxy.yaml).
* **Streaming:** SSE via POST /mcp (when Accept: text/event-stream) and GET /mcp (session notifications).

**2.2 Figma → ThemeBuilder → Kendo: required steps**

1. **Design tokens in Figma Make.** Name tokens by semantic role (e.g., color.surface, color.brandPrimary, radius.sm).
2. **Export to ThemeBuilder.** Use ThemeBuilder’s Fluent base, import token JSON, tweak density/typography.
3. **Export theme assets.** Produce a CSS or SCSS overrides artifact (e.g., themebuilder-overrides.css).
4. **Import order (strict):**
5. @progress/kendo-theme-fluent/dist/all.css
6. themebuilder-overrides.css // overrides from ThemeBuilder
7. globals.css // project‑specific P0 fixes (avoid unless required)
8. **Token parity check.** Verify core tokens (brand, surface, text, focus ring, states) after import using the Kendo ThemeBuilder preview and UI screenshots.
9. **Evidence.** Attach ThemeBuilder export and a short parity checklist to the release Evidence Pack (retain ≥ 1 year).

**3) Information Architecture & Navigation**

**Primary layout.** AppBar (title + session indicators) + Drawer (left nav) + DrawerContent (page body).  
**Routes.**

| **Route** | **Purpose** | **Notes** |
| --- | --- | --- |
| / (Dashboard) | Uptime, child/session counts, readiness state | Read‑only; shows p50/p95 and TTFB tiles if available |
| /sessions | Per‑session activity; SSE stream status | Displays Mcp‑Session‑Id; supports live/paused view |
| /config | Non‑secret key/value config | Values from /config/effective; redaction badges |
| /access | Origin allow‑list visibility | Reads Security:AllowedOrigins via /config/effective |

**No external egress.** All images/icons are local or Kendo assets; block external CDNs in CSP.

**4) Component Map (KendoReact, Fluent v12)**

| **Page** | **Kendo components** | **Data Source** | **Key Interactions** |
| --- | --- | --- | --- |
| Dashboard | AppBar, Card, Grid, Badge | /ready, /healthz | Auto‑refresh (30s), status color coding |
| Sessions | Grid, Toolbar, Badge, Notification | POST /mcp stream, GET /mcp SSE | Start/stop stream view; show heartbeat age |
| Config | Grid, Dialog | /config/effective | Copy key/value; redact secrets; filter/search |
| Access | Grid | /config/effective | Render allow‑list; read‑only |

**5) Accessibility (WCAG 2.2 AA)**

* Use native Kendo semantics (labels, roles, focus indicators, key bindings).
* Maintain **4.5:1** text contrast minimum; ensure focus rings meet brand tokens.
* **Axe smoke tests** on /, /sessions, /config, /access. Fail CI on critical violations.
* Provide ARIA live‑regions for streaming status (“Connected”, “Disconnected”, “Reconnecting…”).
* Keyboard navigation: drawer toggles, grid pagination and row focus.

**6) Streaming UX (SSE) — Interaction Model**

**Transport rules.**

* **Streamed tool call:** POST /mcp with headers: Accept: text/event-stream, Mcp-Session-Id, and bearer.
* **Background notifications:** GET /mcp with Mcp-Session-Id.
* **Heartbeats:** : comment lines every Network:SseKeepAliveSeconds seconds (from DB).
* **Error envelope:** Any HTTP error uses {code, message, requestId?} (never leak payloads).

**UI behavior.**

* Show **connection chip** per stream: Connected / Reconnecting / Disconnected.
* Display **heartbeat age**; warn if gap > 2× configured interval.
* Preserve scroll position on new event: message frames; provide “Pause live” toggle.
* On 401/403/feature‑disabled: stop and surface actionable help; never retry blindly.

**7) Data & Configuration (No‑Hard‑Coding)**

* **Relative API base.** The UI calls '/api' on the same origin. No hostnames checked into code.
* **Readiness & health:** GET /ready, GET /healthz → typed models from OpenAPI.
* **Config:** GET /config/effective → display **non‑secret** keys only; annotate sensitive categories as “redacted by design”.
* **Origin allow‑list:** Render Security:AllowedOrigins from the config endpoint; do not edit in UI.
* **Secrets:** Do not read or display; do not echo tokens in logs. **Telerik license** is injected at build time only (CI secret), never shipped.

**8) Performance & Reliability Budgets (UI slice)**

* **SSE TTFB:** ≤ 200 ms p95 (intra‑VPC).
* **Non‑streaming JSON:** p50 ≤ 300 ms; p95 ≤ 800 ms.
* **Restart‑to‑ready:** ≤ 30 s published by backend and shown on Dashboard.
* **Initial UI bundle:** code‑split major routes; no external CDNs.
* **Observability:** Report frontend errors to console only (scrubbed); never log secrets. Align with backend SLOs and attach snapshots to Evidence Pack after releases.

**9) Security & Privacy Controls (UI)**

* **CSP:** default‑deny; allow connect-src only to same origin/API; img-src to self (no Unsplash/CDNs).
* **Tokens:** Bearer lives in memory/session; never log or persist.
* **Redaction:** Config view must clearly mark items as **non‑secret**; anything sensitive remains server‑side.
* **Origin policy visibility:** UI reflects allow‑list; enforcement happens server‑side using DB‑sourced values.

**10) CI/CD & Licensing (UI Workstream)**

* **Build:** If @progress/kendo-\* dependencies are detected, CI requires **TELERIK\_LICENSE or TELERIK\_LICENSE\_PATH** injected at build time; the license file is created in a temp path and **never logged, committed, or baked into artifacts**.
* **A11y:** Run npm run test:a11y (axe) per route; upload results to Evidence Pack.
* **Evidence Pack:** Include ThemeBuilder export, axe report, and a UI screenshot set per env; retain **≥ 1 year**.

**11) Reference Implementation Notes (non‑secret)**

**Theme import (order matters):**

// src/main.tsx

import '@progress/kendo-theme-fluent/dist/all.css';

import './themebuilder-overrides.css';

**OpenAPI‑typed client (example):**

// api/client.ts

export async function getReady(): Promise<{status:'ok'|'fail'; uptimeSeconds:number; sessionCount:number; childProcesses:number;}> {

const r = await fetch('/api/ready', { credentials: 'include' });

if (!r.ok) throw await r.json(); // {code,message,requestId?}

return r.json();

}

**SSE streamed call (simplified):**

export async function postMcpStream(sessionId: string, body: unknown, onMessage: (json: any)=>void) {

const res = await fetch('/api/mcp', {

method: 'POST',

headers: {

'Accept': 'text/event-stream',

'Content-Type': 'application/json',

'Mcp-Session-Id': sessionId

},

body: JSON.stringify(body)

});

if (!res.ok || !res.body) throw await res.json();

const reader = res.body.getReader();

const decoder = new TextDecoder();

let buf = '';

for (;;) {

const { done, value } = await reader.read();

if (done) break;

buf += decoder.decode(value, { stream: true });

for (const line of buf.split('\n')) {

if (line.startsWith('data: ')) {

const raw = line.substring(6).trim();

try { onMessage(JSON.parse(raw)); } catch {}

}

}

const lastNL = buf.lastIndexOf('\n');

if (lastNL >= 0) buf = buf.slice(lastNL + 1);

}

}

**12) Page‑Level Acceptance Criteria**

**Dashboard**

* Loads /ready within 2 s; renders uptimeSeconds, sessionCount, childProcesses.
* Status tiles reflect **ok/fail** with accessible colors; axe smoke passes.

**Sessions**

* Shows active stream connection with **heartbeat age**; indicates reconnect state on transient network errors; no secret leakage.
* p95 **TTFB ≤ 200 ms** during smoke tests.

**Config**

* Lists only **non‑secret** keys from /config/effective; “redacted by design” note present; copy‑to‑clipboard for values.

**Access**

* Renders Security:AllowedOrigins; indicates server‑side enforcement; no edits allowed.

**13) Risks & Mitigations**

* **Ingress buffering SSE** → ensure text/event-stream **not buffered**; settings documented in Deploy/Scale‑out runbooks.
* **Theme divergence** → keep ThemeBuilder overrides authoritative; avoid ad‑hoc CSS.
* **Dependency drift** → pin versions; run Dependency Review in CI (fail on **high**).

**14) Cross‑References**

* OpenAPI contract: /api/openapi/mcp-proxy.yaml (servers for Alpha/Beta/RTM/Prod; bearer; error envelope; SSE examples).
* Test Strategy: docs/09\_test\_strategy.docx (axe smoke, contract tests, SSE harness).
* CI/CD: docs/10\_ci\_cd.docx (UI build + license injection; required checks).
* Compliance: docs/13\_compliance.docx (CSP/egress; secrets policy).
* Monitoring & SLOs: docs/11\_monitoring.docx (TTFB, p50/p95, 24‑h post‑release checks).
* Runbooks: runbooks/deploy.docx, runbooks/rollback.docx (SSE pass‑through, drain).

**15) Assumptions**

1. The Admin Portal is **read‑only** and hosted behind the same origin as the API (/api), avoiding CORS complexities.
2. **RTM** validates against **Prod DB (read‑only)**; UI reflects parity via /config/effective.
3. **Kendo license** is available and configured in **GitHub Environments** only (never in code/DB).

**16) Next Steps**

1. Import ThemeBuilder overrides; pin Kendo packages; remove prototype UI deps.
2. Wire /ready, /healthz, /config/effective, and **SSE** flows; add axe smoke (npm run test:a11y).
3. Enable the **Telerik license** build guard in CI; attach ThemeBuilder export + axe report to Evidence Pack.
4. Promote through **Alpha → Beta → RTM → Prod** with perf smoke and 24‑h checks.

**Footer (optional for Word header/footer):**  
*MCPX‑KendoBridge • UI/UX Specification • v2.0.0 (Kendo Migration) • 2025‑09‑27 • Confidential — Technijian Internal*