**MCPX‑KendoBridge — Incident Runbook (SEV, Comms, Evidence & Post‑Mortem)**

**Document:** runbooks/incident.docx  
**Version:** 2.0.0 (Kendo Migration)  
**Last Updated:** 2025‑09‑27  
**Owner:** SRE Lead (Responsible) — DoSE (Accountable) — DocFactory (Author)  
**Applies to:** API (.NET 8, Streamable‑HTTP + SSE), Admin Portal (KendoReact Fluent v12, read‑only)

**Purpose.** Provide a **repeatable, auditable** process for detecting, triaging, communicating, mitigating, and learning from incidents affecting MCPX‑KendoBridge. This runbook integrates with Technijian’s **GitHub‑first** SDLC (required checks), four‑environment flow (**Alpha → Beta → RTM → Prod**), **SSE** transport requirements (TTFB/heartbeats), **session stickiness** by Mcp‑Session‑Id, and Evidence Pack retention (≥ 1 year).

**Compliance banner (always in effect):** **Add‑only** schema; **Stored‑procedure‑only** DAL (app has **EXECUTE‑only** on SPs); **No‑Hard‑Coding** of dynamic values (child cmd/args/cwd, request timeout, heartbeat cadence, Origin allow‑list, feature flags). All dynamic values are **DB‑sourced** (non‑secret) via sp\_Config\_\*, sp\_Feature\_IsEnabled, sp\_Lookup\_Get. **Secrets** (SQL connection strings, **Telerik license**) **never** appear in code/DB/logs; configure them **only** in **GitHub Environments**.

**1) Scope & Triggers**

**In scope:** Availability, correctness, security, performance (latency & **SSE TTFB**), configuration drifts, streaming quality (heartbeats), session/child stability, and UI read‑only portal issues.

**Primary triggers (open an incident if sustained):**

| **Trigger** | **Condition (sustained)** | **Source** |
| --- | --- | --- |
| Availability degradation | Availability < 99% (rolling 10 min) or spike in 5xx | SLI dashboard |
| **SSE TTFB** regression | p95 > 200 ms for 15 min | Streaming TTFB SLI |
| Readiness failures | /ready failing > 2 min | Readiness SLI |
| Heartbeat drift | Heartbeat gap p95 > 2× Network:SseKeepAliveSeconds | Streaming quality SLI |
| Policy regression | Spike in origin\_forbidden or feature\_disabled | Error‑code counters |
| Child instability | Rising child\_restart\_count or bad\_gateway\_child\_unavailable | Metrics/logs |
| Secret exposure (any) | Suspected leak (token/SQL/Telerik license) | Secret scanning / reports |

SLO & alert thresholds are defined in **Monitoring, SLOs & Post‑Release**.

**2) Severity, SLAs & Examples**

| **Sev** | **Definition** | **Examples** | **Target Response** |
| --- | --- | --- | --- |
| **P1** | Broad outage or severe degradation | /ready red; availability < 99%; **TTFB p95>200 ms**; persistent 5xx | Page on‑call immediately; mitigation ≤ 30 min |
| **P2** | Degraded but limited impact | Heartbeat drift; intermittent 40x/50x; minor config drift | Acknowledge ≤ 15 min; fix ≤ 24 h |
| **P3** | Minor/cosmetic or low risk | UI cosmetic, non‑blocking a11y | Acknowledge ≤ 1 business day |

**3) Roles & ICS (Incident Command System)**

| **Role** | **Primary** | **Responsibilities** |
| --- | --- | --- |
| **Incident Commander (IC)** | SRE Lead (backup: DoSE) | Owns decisions, clock, severity, resources, transitions (mitigate→resolve→close) |
| **Operations Lead** | SRE | Diagnostics, rollout/rollback, ingress/SSE checks, sticky routing, scale‑out |
| **Tech Lead (Service)** | Dev Lead | Root‑cause analysis, forward‑fix, config rollback vs code rollback call |
| **Comms Lead** | Program/Account | Stakeholder updates (internal/external), status pages, cadence |
| **Scribe** | QA or DocFactory | Live timeline, decisions, artifacts, evidence paths |
| **Security Lead** | SecLead | Secrets policy, exposure triage, license rotation invocation |

**4) First 5 Minutes (stabilize & get eyes)**

1. **Declare incident** (set SEV) and open the comms channel (#mcpy‑incident‑<date>).
2. **Assign roles** (IC, Ops, Tech, Comms, Scribe, Security).
3. **Freeze risky changes**: pause deployments/promotions; keep rollback runbook at hand.
4. **Top‑line checks** (paste outputs to channel; no secrets):
5. curl -fsS https://<env>.example.com/api/ready | jq .
6. curl -fsS https://<env>.example.com/api/healthz | jq .
7. curl -fsS https://<env>.example.com/api/config/effective | jq .
8. **Streaming probe (TTFB):**
9. curl -N -H 'Accept: text/event-stream' \
10. -H 'Mcp-Session-Id: ic-probe' \
11. -H 'Content-Type: application/json' \
12. -w '\nTTFB(ms)=%{time\_starttransfer}\n' \
13. -d '{"jsonrpc":"2.0","id":"1","method":"ping","params":{"stream":true}}' \
14. https://<env>.example.com/api/mcp
15. **Ingress sanity:** confirm **no buffering** of text/event-stream and appropriate read/idle timeouts.
16. **Sticky sessions:** verify hashing by Mcp‑Session‑Id still active (same pod handling repeat calls).

(Transport & SSE budgets are defined in Monitoring/NFR; use them as acceptance targets.)

**5) Streaming‑First Triage (decision tree)**

**A. Readiness is red?**  
→ Follow **Deploy/Rollback** runbooks to restore readiness; verify DB SP reachability and child spawn probe.

**B. TTFB p95 > 200 ms or heartbeat drift?**

1. Check ingress buffering/timeouts for text/event-stream.
2. Confirm Network:SseKeepAliveSeconds value via /config/effective (non‑secret).
3. Inspect CPU saturation and session\_count per pod; consider **scale‑out** (see **Scale‑out** runbook).
4. If regression tied to new image, perform **Code Rollback** (graceful SSE drain).

**C. Surge in origin\_forbidden?**

1. Check Security:AllowedOrigins in /config/effective.
2. If misconfigured: perform **Config Rollback** (DB‑sourced; non‑secret).
3. If attack traffic suspected: rate‑limit at edge (if available) and notify Security.

**D. Legacy endpoints causing issues?**  
→ Ensure EnableLegacyHttpSse=false unless explicitly required; verify 403 feature\_disabled.

**E. Suspected secret exposure?**  
→ Stop‑the‑line, rotate in **GitHub Environments** (never via DB), follow **Rotate Telerik License** runbook for Kendo license, and sanitize logs/evidence.

**6) Diagnostics Playbook**

**6.1 API/Child Process**

* **Child health:** check child\_up, child\_restart\_count metrics; examine structured logs for childPid.
* **Session binding:** confirm requests with same Mcp‑Session‑Id hit same replica (LB affinity).
* **Timeouts:** compare observed durations to Network:RequestTimeoutSeconds (DB‑sourced).
* **JSON vs SSE paths:** reproduce both (JSON fallback + streamed) to isolate transport.

**6.2 Database & Config**

* **SP availability:** transient failures in sp\_Config\_\*/sp\_Feature\_IsEnabled cause readiness to fail; confirm with readiness logs.
* **Config parity (RTM):** ensure RTM points to **Prod DB (read‑only)** and parity checks pass before Prod.

**6.3 UI (KendoReact)**

* **CSP:** confirm no third‑party egress; local assets only.
* **A11y:** axe smoke failures are P3 unless blocking; ensure no token display/logging.

**6.4 Security**

* **Error envelope:** ensure errors use canonical {code,message,requestId?}; no payload bodies in logs.
* **Spike taxonomy:** bucket by origin\_forbidden, missing\_session\_id, feature\_disabled, timeout, etc., to guide action.

**7) Mitigation Options (choose least‑disruptive first)**

1. **Config Rollback (DB‑sourced, non‑secret):** restore AllowedOrigins, cadence, or timeouts via idempotent seeds/SPs.
2. **Feature Disable:** set EnableLegacyHttpSse=false.
3. **Scale‑out:** add replicas; ensure **PDB** and sticky sessions; verify SSE drain on scale‑in later.
4. **Code Rollback (image flip, graceful SSE drain):** use **Rollback** runbook; confirm TTFB and heartbeats post‑flip.
5. **Forward‑fix:** hot‑patch only with IC approval; produce post‑mortem actions.

**8) Communications (templates)**

**Guidelines:** never include secrets; avoid payload bodies; include timestamp, severity, blast radius, current action, ETA for next update. Use requestIds/sessionIds when needed.

**A. Initial internal page (P1)**

* *Subject:* [P1][MCPX-KendoBridge] SSE TTFB regression in Prod — rollback in progress
* *Body (example)*
* When: 2025-09-27 14:20Z
* Severity: P1
* Impact: SSE streamed calls exceed TTFB p95>200ms; availability 98.8% last 10m
* Actions: Freezing deploys; checking ingress buffering; rolling back to LKG if no improvement in 10m
* Next update: 14:30Z
* IC: @sre-lead | Channel: #mcpy-incident-20250927

**B. External status update (if applicable)**

* *Title:* “Degraded streaming performance”
* *Message:* “We’re investigating increased latency on streamed responses. Non‑streaming requests operate normally. A mitigation is in progress.”

**C. Resolution notice**

* “Issue mitigated via code rollback to LKG. We are monitoring SSE TTFB and heartbeats; all metrics within targets.”

**9) Evidence Capture (for the Release pack)**

Collect and attach to the current Release; retain **≥ 1 year**:

* Timeline (timestamps, decisions, actions).
* /ready, /healthz, /config/effective snapshots (non‑secret).
* Monitoring screenshots: Availability, Latency p50/p95, **SSE TTFB**, Readiness timeline.
* Perf smoke outputs post‑mitigation.
* If rollback: LKG image digest, rollout logs.
* If configuration change: before/after seed or SP call records (no secrets).
* If security‑related: redacted log samples, Secret‑scan confirmation, license rotation report.

**10) Resolution & Close**

**Done** when:

* SLOs within target: availability ≥ 99.9%, JSON p50/p95 ≤ 300/800 ms, **SSE TTFB p95 ≤ 200 ms**, heartbeat cadence nominal.
* Root cause identified or narrowed (with hypotheses).
* Evidence uploaded and **24‑hour** post‑release checks scheduled (if Prod).

**11) Post‑Mortem (blameless) — Template**

**Title:** YYYY‑MM‑DD — <summary>  
**Summary:** What happened; who/what was impacted; duration.  
**Timeline:** (UTC) events/actions/decisions; include requestIds/sessionIds where helpful.  
**Root cause:** Technical + contributing factors (design/process).  
**Detection:** Which alerts fired; signals that should have fired.  
**Mitigation & fix:** Immediate actions; permanent fixes; regressions tests.  
**What went well / can improve:** People/process/tooling.  
**Action items (owners, due dates):** Include monitoring, tests, documentation updates, runbook refinements.  
**Evidence links:** Release assets, dashboards, PRs, commits.

Attach the post‑mortem PDF to the Release Evidence.

**12) RACI (Incident)**

| **Activity** | **A** | **R** | **C** | **I** |
| --- | --- | --- | --- | --- |
| Declare severity & assemble team | DoSE | SRE Lead | Dev Lead, SecLead | QA, Stakeholders |
| Technical triage & mitigation | DoSE | SRE + Dev Lead | QA, SecLead | Stakeholders |
| Comms (internal/external) | DoSE | Comms Lead | SRE, Dev Lead | All |
| Evidence capture & release upload | DoSE | Scribe + SRE | QA | All |
| Post‑mortem & actions | DoSE | Dev Lead | SRE, QA, SecLead | All |

**13) Cross‑References**

* **Deploy Runbook** (promotion, SSE ingress settings) & **Rollback Runbook** (graceful SSE drain).
* **Monitoring & SLOs** (TTFB, heartbeat, availability, post‑release checks).
* **Compliance** (secrets policy, CSP/egress, DB rules); **Test Strategy** (SSE harness, axe smoke).
* **Evidence Pack** (what to capture & retain, ≥ 1 year).

**14) Assumptions**

1. Ingress supports **SSE** with **no buffering**; read/idle timeouts allow configured heartbeat cadence.
2. Sticky routing targets the same replica for a given Mcp‑Session‑Id.
3. **RTM** uses **Prod DB (read‑only)**; parity checks precede Prod promotion.

**15) Next Steps**

* Schedule a quarterly **game day**: simulate SSE regression → practice triage, rollback, and evidence capture.
* Automate evidence snapshotting (OpenAPI, tests, SBOM, monitoring images) into the Release process.
* Keep runbooks aligned with evolving **NFRs** and **OpenAPI** error examples.

**Footer (optional for Word header/footer):**  
*MCPX‑KendoBridge • Incident Runbook • v2.0.0 • 2025‑09‑27 • Confidential — Technijian Internal*