**MCPX‑KendoBridge — Vision & Objectives**

**Document:** docs/01\_vision.docx  
**Project Code:** MCPX‑KendoBridge  
**Version:** 1.0.0 (Draft)  
**Last Updated:** 2025‑09‑23  
**Document Owner:** DoSE (Accountable) — DocFactory (Responsible)

**One‑liner:** A secure, observable **HTTP/Streamable‑HTTP** proxy that spawns and bridges the **Telerik KendoReact MCP** (STDIO) so modern assistants (e.g., ChatGPT/MyGPT connectors) can consume Kendo MCP remotely—without violating Technijian’s **GitHub‑first**, **No‑Hard‑Coding**, and **SP‑only** rules.

**Document Control**

**Revision History**

| **Version** | **Date** | **Author/Role** | **Summary of Changes** |
| --- | --- | --- | --- |
| 1.0.0‑D | 2025‑09‑23 | DocFactory (R) | Initial draft of Vision & Objectives |

**Approvals**

| **Name / Role** | **Responsibility** | **Signature / Date** |
| --- | --- | --- |
| Director of Software Engineering (DoSE) | Accountable (A) | \_\_\_\_ / \_\_\_\_ |
| Systems Architect (T‑Arch) | Consulted (C) | \_\_\_\_ / \_\_\_\_ |
| QA Lead | Consulted (C) | \_\_\_\_ / \_\_\_\_ |

**Distribution List**

* Engineering, QA, Architecture, DevOps, Client Services

**Table of Contents**

1. Executive Summary
2. Problem & Context
3. Goals (Business & Technical)
4. Non‑Goals
5. Scope (MVP vs Optional)
6. Users & Stakeholders
7. Product Principles
8. Environment Strategy & Promotion (GitHub‑first)
9. Success Metrics & SLOs
10. High‑Level Requirements (Summary)
11. Risks & Mitigations
12. Deliverables
13. Release Phases & Milestones
14. Compliance & Guardrails
15. References
16. Appendices
17. Assumptions
18. Next Steps

**1. Executive Summary**

**Vision.** Make KendoReact’s local **STDIO MCP** usable in cloud and enterprise environments via a **stateless, session‑aware** web API that speaks MCP over HTTP with **SSE** streaming, enforces **Origin allow‑lists**, and exposes health/metrics for reliable operations across **Alpha → Beta → RTM → Prod**.

**Value.**

* **Interoperability:** HTTP‑native MCP access for ChatGPT/MyGPT connectors and peers.
* **Security & Compliance by default:** CORS Origin allow‑list; **no secrets in code/docs**; **SP‑only** config; CI gates (CodeQL, Dependency Review, Secret Scanning); evidence retention ≥ 1 year.
* **Operability:** Health/readiness endpoints, structured JSON logs, and minimal metrics for sessions and child processes.
* **Scalability:** Horizontally scalable with **one child process per session** to isolate workloads and bound resource use.

**2. Problem & Context**

**Today:** The KendoReact MCP server runs as a **local STDIO** tool. This limits usage to desktop/co‑located processes and blocks HTTP‑first assistants and network‑segmented deployments.

**Pain Points**

* No remote transport or session isolation via HTTP/SSE.
* Limited health/metrics; opaque child lifecycle.
* Browser/assistant security posture unclear (CORS/origin).
* Configuration drift across environments.

**Opportunity**

* Provide a **web‑standard** MCP surface with **streaming** semantics and robust environment controls that drop into Technijian’s GitHub‑first SDLC and four‑stage promotion model.

**3. Goals (What success looks like)**

**3.1 Business Goals**

1. **Enable adoption** of Kendo MCP by HTTP‑only assistants and connectors.
2. **Reduce integration time** from weeks to days with an OpenAPI 3.1 contract and runbooks.
3. **Lower operational risk** via standardized health/metrics/logging and CI gates.

**3.2 Technical Goals**

1. **Primary transport:** Streamable‑HTTP with **SSE** for streaming responses and background notifications.
2. **Session isolation:** One child STDIO process per **Mcp‑Session‑Id**; request‑scoped streaming, session‑scoped notifications.
3. **Security controls:** Enforce **Origin** allow‑list from DB (Security:AllowedOrigins); redact secrets; standard error envelope.
4. **Operability:** /healthz, /ready, /config/effective (redacted); metrics (session count, child up/down).
5. **Compliance alignment:** Add‑only DB migrations; **SP‑only** DAL; **No‑Hard‑Coding**; GitHub‑first (branch protections, merge queue, required checks).

**4. Non‑Goals (Explicitly out of scope)**

* Persisting or transforming MCP payloads (we broker only).
* Storing license keys or other secrets in the DB.
* Building a full admin console beyond **read‑only** health/metrics (optional minimal UI only).
* Changing Kendo MCP behavior (it remains a **black box**).

**5. Scope (MVP vs Optional)**

**5.1 MVP Scope**

**Endpoints**

* POST /mcp — Accepts a single JSON‑RPC 2.0 message. Streams when Accept: text/event-stream; otherwise returns JSON.
* GET /mcp — Opens SSE channel for server‑initiated messages (background notifications).
* GET /healthz, GET /ready — Liveness/readiness checks.
* GET /config/effective — **Non‑secret** effective config (read‑only; values sourced from DB).

**Sessioning**

* Use/return **Mcp‑Session‑Id**. Spawn child on first request (or explicit initialize). **One child per session**.

**Security**

* Enforce **Origin** allow‑list from Security:AllowedOrigins (DB). No secrets logged.

**Observability**

* JSON logs with requestId, sessionId, childPid; metrics: session\_count, child\_up, child\_restart\_count.

**DB & Config**

* Add‑only tables: AppConfig, FeatureFlag.
* SPs: sp\_Config\_GetValue, sp\_Config\_GetAll, sp\_Feature\_IsEnabled (+ sp\_Lookup\_Get reserved).
* Seed keys (non‑secret):
  + Mcp:ChildCommand = npx
  + Mcp:ChildArgs = -y @progress/kendo-react-mcp@latest
  + Mcp:ChildCwd = ""
  + Security:AllowedOrigins = https://chat.openai.com,https://platform.openai.com
  + Network:SseKeepAliveSeconds = 15
  + Network:RequestTimeoutSeconds = 120

**Error Envelope**  
{ code: string; message: string; requestId?: string }

**5.2 Optional (feature‑flagged or later)**

* **Legacy transport:** POST /messages + GET /sse (HTTP+SSE) behind EnableLegacyHttpSse.
* **Minimal Ops UI:** React + **KendoReact (Fluent v12 + ThemeBuilder overrides)**, **read‑only** dashboard for health/sessions/config.

**6. Users & Stakeholders**

* **Remote MCP Client** (e.g., ChatGPT/MyGPT Connector): invokes POST /mcp and GET /mcp SSE.
* **Legacy MCP Client** (feature‑flagged): uses /messages + /sse.
* **Kendo MCP Child Process:** spawned via npx -y @progress/kendo-react-mcp@latest using STDIO.
* **Ops Admin:** monitors health/metrics and promotes releases (Alpha → Prod).

**7. Product Principles**

1. **Black‑box fidelity:** Never alter Kendo MCP semantics; only bridge transport.
2. **No‑Hard‑Coding:** All dynamic values from DB (AppConfig, FeatureFlag, future Lookup) via SPs. **No ad‑hoc SQL.**
3. **SP‑only DAL:** SqlCommand(CommandType.StoredProcedure); **add‑only** migrations (no destructive DDL).
4. **Security first:** CORS Origin allow‑list from DB; redact secrets; stable error envelope.
5. **GitHub‑first SDLC:** Branch protections, merge queue, required checks (Build/Tests, CodeQL, Dependency Review, Secret Scanning); SBOM artifact; ≥ 1‑year evidence retention.
6. **Observability:** Health/readiness, minimal metrics, structured logs with correlation.
7. **4‑Env discipline:** **RTM validates on Prod DB** before Prod promotion.

**8. Environment Strategy & Promotion (GitHub‑first)**

* **Environments:** Alpha → **Beta** → **RTM (validates on Prod DB)** → **Prod**.
* **OpenAPI servers** enumerate all four (update hostnames per deployment).
* **Promotion gates** (merge‑queue aware): Build/Tests, CodeQL, Dependency Review, Secret Scanning, OpenAPI lint/diff, SBOM.
* **Secrets** (SQL connection string, Telerik license via TELERIK\_LICENSE\_PATH / TELERIK\_LICENSE) exist only in **GitHub Environments** or vendor portals—**never** in code/docs/DB.

**9. Success Metrics & SLOs**

**Service KPIs**

* **Latency (non‑streaming, intra‑VPC):** P50 ≤ **300 ms**, P95 ≤ **800 ms**.
* **Streaming TTFB:** ≤ **200 ms**.
* **Availability:** **99.9%** monthly.
* **Scalability:** ≥ **200** concurrent sessions per replica; **CPU‑bound before memory**.
* **Quality:** 0 secret leaks in logs (scanned); CI gates green on main.
* **Operations:** Single‑instance restart recovery ≤ **30 s**; incident MTTR < **30 m**.
* **A11y (if UI):** Pass axe smoke; WCAG 2.2 AA for core screens.

**Evidence of Success**

* OpenAPI 3.1 published and versioned at /api/openapi/mcp-proxy.yaml; used by connector integration tests.
* Evidence pack per release (tests, SARIF, SBOM, secret‑scan summary, OpenAPI diff, monitoring snapshot) retained ≥ **1 year**.

**10. High‑Level Requirements (Summary)**

* **Transport:** POST /mcp (JSON vs SSE), GET /mcp (SSE). Legacy /messages, /sse behind feature flag.
* **Sessioning:** One child process per Mcp‑Session‑Id; background notifications over SSE.
* **Security:** Origin allow‑list from DB; stable error envelope; no secrets in logs.
* **Health & Config:** /healthz, /ready, /config/effective (redacted).
* **Observability:** JSON logs + correlation; metrics (session count, child up/down).
* **DB & DAL:** Add‑only schema; **SP‑only** access; seeds for child command/args/origins/timeouts.
* **CI/CD:** GitHub‑first with branch protections, merge queue, CodeQL, Dependency Review, Secret Scanning, SBOM.

**11. Risks & Mitigations**

| **Risk** | **Impact** | **Mitigation** |
| --- | --- | --- |
| Ingress proxies buffering SSE | Breaks streaming semantics | Configure ingress for text/event-stream; keep‑alive per Network:SseKeepAliveSeconds |
| Child process churn or zombie PIDs | Resource leaks / instability | Supervise via session registry; graceful shutdown; track child\_restart\_count |
| Misconfigured Origins | Valid clients blocked or unsafe exposure | Manage Security:AllowedOrigins via DB; change controlled via migration/runbook |
| License handling errors | Build/runtime failures | Keep license in vendor portal + GitHub Environments; rotation runbook; **never in DB** |
| DB latency/outage | Config fetch slows/blocks readiness | Cache config snapshot in memory with TTL; fail‑fast readiness |
| Feature flag drift | Legacy endpoints unintentionally enabled | Control EnableLegacyHttpSse via FeatureFlag; audited changes |

**12. Deliverables**

* **Architecture docs:** Vision (this file), Context/Container/Component diagrams, FR/NFR, Data & DB Contracts (+ SPs), **OpenAPI 3.1**, Gherkin tests, CI/CD plan, Runbooks, ADRs.
* **Operational assets:** Health/ready endpoints, JSON logging, minimal metrics, evidence pack index and retention policy (≥ 1 year).
* **UI (optional):** Minimal **read‑only** Ops UI using **KendoReact (Fluent v12 + ThemeBuilder overrides)** with axe smoke tests.

**13. Release Phases & Milestones**

* **Sprint 0 — Discovery:** Draft Vision, Actors, Context, FR/NFR; initial ADRs; repo scaffolding.
* **Sprint 1 — Transport & Sessioning:** /mcp POST/GET; SSE streaming & keep‑alives; session registry; error envelope; unit/integration tests.
* **Sprint 2 — Security & Observability:** Origin allow‑list from DB; JSON logs; metrics; /healthz, /ready, /config/effective; deploy **Alpha**.
* **Sprint 3 — CI/CD & Evidence:** CodeQL, Dependency Review, Secret Scanning, SBOM; deploy **Beta**; perf tests; evidence wiring.
* **Sprint 4 — RTM Validation:** **RTM** on Prod DB (read‑only); OpenAPI finalized; acceptance tests pass; rollback validated.
* **Sprint 5 — Prod & Hardening:** Prod cut; 24‑hour post‑release checks; optional Ops UI.

**14. Compliance & Guardrails (Summary)**

* **DB:** Add‑only migrations; **Stored‑procedure‑only** access; **No‑Hard‑Coding** (config/flags/lookups via SPs).
* **Secrets:** Only in **GitHub Environments** or vendor portals (e.g., TELERIK\_LICENSE\_PATH, SQL connection string).
* **Pipelines:** Branch protections, merge queue, required checks, SBOM publication, ≥ 1‑year artifact retention.
* **A11y (if UI):** WCAG 2.2 AA baseline; axe smoke.
* **Auditability:** Evidence pack per release (test results, SARIF, SBOM, secret‑scan summary, OpenAPI diff, monitoring snapshot).

**15. References**

* Technijian DocFactory SDLC defaults and quality gates (GitHub‑first, four environments, evidence retention, UI SOPs).

**16. Appendices**

**16.1 Key Defaults (seeded in DB; non‑secret)**

* Mcp:ChildCommand = npx
* Mcp:ChildArgs = -y @progress/kendo-react-mcp@latest
* Mcp:ChildCwd = ""
* Security:AllowedOrigins = https://chat.openai.com,https://platform.openai.com
* Network:SseKeepAliveSeconds = 15
* Network:RequestTimeoutSeconds = 120

**16.2 Error Envelope (canonical)**

{ "code": "string", "message": "string", "requestId": "optional string" }

**16.3 OpenAPI Servers (replace TBDs)**

* https://alpha.<tbd>/api
* https://beta.<tbd>/api
* https://rtm.<tbd>/api *(validates on Prod DB)*
* https://app.<tbd>/api

**17. Assumptions**

1. Deployment is containerized with ingress that supports **SSE** and streaming responses.
2. SQL Server is reachable from all environments; migrations run in CI/CD.
3. Authentication uses platform‑provided bearer tokens (enforced at gateway or app; out of scope here).
4. The Kendo MCP child is launched via **npx** and requires no code‑level changes in this project.
5. Optional Ops UI is **read‑only** and can be deferred without blocking backend release.

**18. Next Steps**

1. Fill environment URLs in **OpenAPI 3.1** (/api/openapi/mcp-proxy.yaml) and commit.
2. Apply the provided **DB migrations and SPs**; seed non‑secret keys in **Alpha**.
3. Implement transport/session/bridge and origin checks; wire structured logs/metrics; surface health+ready+config endpoints.
4. Configure **GitHub Environments** with required secrets (SQL, Telerik license); enable **branch protections** and **merge queue**; turn on CodeQL, Dependency Review, Secret Scanning; publish **SBOM**.
5. Run Gherkin acceptance in **Alpha**, then **Beta**; validate **RTM** against Prod DB; finalize evidence pack; tag **Prod** release.

**Document Footer (insert in Word header/footer as needed):**