**MCPX‑KendoBridge — Glossary**

**Document:** docs/02\_glossary.docx  
**Project Code:** MCPX‑KendoBridge  
**Version:** 1.0.0 (Draft)  
**Last Updated:** 2025‑09‑23  
**Document Owner:** DoSE (Accountable) — DocFactory (Responsible)

**Purpose.** This glossary defines all terms, acronyms, headers, roles, artifacts, and conventions used in the MCPX‑KendoBridge solution. It aligns with Technijian’s GitHub‑first SDLC, **No‑Hard‑Coding**, and **SP‑only** database policy, and references DocFactory defaults for evidence, CI/CD, and UI SOPs.

**Document Control**

**Revision History**

| **Version** | **Date** | **Author/Role** | **Summary of Changes** |
| --- | --- | --- | --- |
| 1.0.0‑D | 2025‑09‑23 | DocFactory (R) | Initial end‑to‑end glossary |

**Approvals**

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

**How to Use This Glossary**

1. **Start with Acronyms** to decode shorthand used across specs.
2. **Consult the Alphabetical Terms** for precise definitions.
3. **Check Header & Naming Conventions** before adding new endpoints or DB objects.
4. **Reference Policies** marked as **DB COMPLIANCE** and **No‑Hard‑Coding** wherever DB/API/UI interact with configuration.

**Acronyms & Abbreviations**

| **Term** | **Expansion** | **In this project** |
| --- | --- | --- |
| AC | Acceptance Criteria | Pass/fail conditions in Gherkin scenarios. |
| ADR | Architecture Decision Record | Permanent decision log (e.g., transport, session model). |
| A11y | Accessibility | WCAG 2.2 AA baseline; axe smoke tests in UI (if present). |
| ASVS | Application Security Verification Standard | Mapping reference used in Security & Compliance doc. |
| CI/CD | Continuous Integration / Continuous Delivery | GitHub Actions pipelines; multi‑env promotion. |
| CORS | Cross‑Origin Resource Sharing | Origin allow‑list enforced from DB config. |
| DAL | Data Access Layer | **SP‑only** access via SqlCommand with StoredProcedure. |
| DB | Database | SQL Server with **add‑only** migrations and SP‑only policy. |
| DoR | Definition of Ready | Preconditions to start work. |
| DoD | Definition of Done | Evidence and gate requirements to complete work. |
| FR | Functional Requirements | System behavior (“what it must do”). |
| HPA | Horizontal Pod Autoscaler | Used when running in K8s (scale‑out runbook). |
| JSON‑RPC | JSON‑Remote Procedure Call | Envelope for MCP requests/responses. |
| Kendo MCP | Telerik KendoReact MCP server | The child STDIO process we spawn/bridge. |
| MCP | Model Context Protocol | Protocol class used by assistants; proxied via HTTP/SSE here. |
| p50/p95 | Percentile Latencies | SLO metrics (median and 95th percentile). |
| PR | Pull Request | Requires checks (Build/Tests, CodeQL, Dependency Review, Secret Scanning). |
| RTM | Release to Manufacture | Environment validating on **Prod DB** prior to Prod. |
| SBOM | Software Bill of Materials | Produced and retained with releases (≥ 1 year). |
| SSE | Server‑Sent Events | Streaming transport (text/event-stream). |
| STDIO | Standard I/O | Child transport: stdin/stdout for Kendo MCP. |
| SLO | Service Level Objective | Availability & latency targets. |
| TTFB | Time to First Byte | Streaming first byte; budget ≤ 200 ms. |
| UI | User Interface | Optional read‑only Ops interface using KendoReact Fluent v12. |
| WCAG | Web Content Accessibility Guidelines | UI meets level 2.2 AA if present. |

**Conventions & Style**

* **Case and capitalization**
  + Header names written as specified: Mcp-Session-Id, MCP-Protocol-Version, Accept, Origin, Content-Type.
  + Stored procedures: sp\_<Area>\_<Verb>\_<Noun> (e.g., sp\_Config\_GetValue).
  + Migrations: VYYYYMMDDHHMM\_\_<slug>.sql (e.g., V202509230900\_\_init\_schema.sql).
* **DB COMPLIANCE (always applicable)**
  + **Add‑only schema changes**, **Stored‑procedure‑only** access, **No‑Hard‑Coding**.
  + Dynamic values come from DB via SPs (sp\_Config\_\*, sp\_Feature\_IsEnabled, sp\_Lookup\_Get) or platform vaults for secrets.
* **Secrets**
  + **Never** in code, documents, or DB. Configure only in **GitHub Environments** or vendor portals (e.g., TELERIK\_LICENSE\_PATH / TELERIK\_LICENSE).
* **Environments**
  + **Alpha → Beta → RTM → Prod**, with **RTM validating on Prod DB**. Promotions occur via GitHub Environments and required checks.

**Alphabetical Glossary of Terms**

**Accept (HTTP header)**  
Indicates desired media type for the response. When set to text/event-stream on POST /mcp, the server streams SSE events instead of a single JSON payload.

**Access Token / Bearer Auth**  
Authentication mechanism for protected endpoints. Modeled as bearerAuth in OpenAPI. Tokens are managed by the platform, not stored in DB or code.

**Add‑Only Migration**  
A DB change that only creates new objects or extends existing ones (columns/tables/SPs) without destructive alterations. Ensures safe forward evolution across Alpha → Prod.

**Admin (Ops Admin)**  
User persona with read‑only access to health, metrics, and effective configuration; responsible for environment promotions and operational checks.

**AppConfig (table)**  
Key/value store for non‑secret configuration. Primary keys are strings ([Key]), values are NVARCHAR(MAX). Accessed strictly via sp\_Config\_\* SPs. Used to drive dynamic behavior such as child command/args, SSE keep‑alive, request timeouts.

**Application Logs (JSON)**  
Structured logs including requestId, sessionId, and childPid. Used for correlation and incident response. No secrets or PII permitted.

**ASVS**  
Security verification reference to map controls (authn, access control, config, transport security) in Security & Compliance documentation.

**Backpressure**  
Flow control where streamed output is paced by the receiver. With SSE, the proxy flushes each event and avoids buffering large responses unnecessarily.

**BearerAuth (OpenAPI)**  
Security scheme definition for endpoints that require authentication; applied globally in openapi/mcp-proxy.yaml.

**C4 Model**  
A set of architecture views: Context, Container, and Component diagrams used across this project.

**Child Process (Kendo MCP)**  
Process spawned per session using command/args from DB (e.g., npx -y @progress/kendo-react-mcp@latest). Communicates via STDIO; lifecycle is supervised by the session manager.

**CodeQL**  
Static analysis for code security. A required GitHub check; results published as SARIF. Evidence retained with releases.

**Config Effective (endpoint)**  
GET /config/effective returns a **redacted**, read‑only snapshot of non‑secret config sourced from DB. Used for debugging drifts and audits.

**Content Negotiation**  
Server behavior that changes response format based on Accept header (e.g., JSON vs SSE stream).

**CORS (Origin Allow‑List)**  
Cross‑origin policy enforcing allowed Origin values provided by the DB config key Security:AllowedOrigins. Disallowed origins receive 403 with a standard error envelope.

**Dependency Review**  
GitHub check that blocks PRs introducing vulnerable or high‑risk dependencies.

**DoR / DoD**  
Definitions ensuring features are ready to start (DoR) and completed with evidence (DoD), including OpenAPI updates, tests, and runbooks.

**Effective Configuration**  
The resolved set of non‑secret runtime settings from DB (and environment/platform defaults). Exposed via API for ops visibility; secrets always redacted.

**Error Envelope**  
Canonical error shape: { code: string; message: string; requestId?: string }. Applied consistently across endpoints.

**EventSource (Client)**  
Browser/assistant API for consuming SSE streams. In our context, used by clients listening to GET /mcp or streamed POST /mcp.

**Evidence Pack**  
Bundle of artifacts—test reports, CodeQL SARIF, secret‑scan summary, SBOM, OpenAPI diff/lint, and monitoring snapshot—retained ≥ 1 year per release.

**Feature Flag**  
Boolean toggle stored in FeatureFlag table and accessed via sp\_Feature\_IsEnabled. Used to gate legacy endpoints (/messages, /sse).

**Figma Make → ThemeBuilder (UI SOP)**  
Design‑to‑implementation flow for the optional Ops UI: wireframes & tokens in Figma Make; export to ThemeBuilder; import base **Fluent v12** theme and apply overrides to KendoReact components.

**Gherkin**  
Plain‑text BDD syntax for acceptance tests. See /tests/gherkin/\* for scenarios covering sessions, streaming, notifications, and origin checks.

**GitHub Environments**  
Protected deployment contexts (alpha, beta, rtm, prod) holding secrets and approvals for promotion workflows.

**GitHub‑First SDLC**  
Operational model prioritizing GitHub Issues/Projects/Actions, branch protections, merge queue, and mandatory security checks.

**Health / Ready (endpoints)**  
/healthz for liveness; /ready for readiness (DB reachability, optional child spawn). Used by probes/monitors.

**Heartbeat (SSE)**  
A colon‑prefixed SSE comment (: <timestamp>) sent at intervals specified by Network:SseKeepAliveSeconds to keep connections alive through intermediaries.

**HPA (Horizontal Pod Autoscaler)**  
Container orchestration feature to scale replicas based on CPU/memory/requests. Our design expects **CPU‑bound before memory** at saturation.

**JSON‑RPC 2.0**  
Message envelope used between client and MCP server. Opaque to the proxy, which forwards payloads and streams STDIO output as SSE events.

**KendoReact Fluent v12**  
Base UI theme for KendoReact components; augmented by ThemeBuilder overrides in the optional Ops UI.

**Latency (p50/p95)**  
Service performance metrics measured for non‑streaming requests. Targets: **p50 ≤ 300 ms**, **p95 ≤ 800 ms** (intra‑VPC).

**Legacy Endpoints**  
POST /messages and GET /sse for HTTP+SSE compatibility. Disabled by default; enabled by EnableLegacyHttpSse feature flag.

**License Secret (Telerik)**  
Configured only via environment (TELERIK\_LICENSE\_PATH or TELERIK\_LICENSE). Never stored in DB or code; rotation documented in runbooks.

**Logs (Structured JSON)**  
Event records including correlation fields. Logging excludes secrets/PII; used for diagnostics and audits.

**Merge Queue**  
GitHub feature to batch and validate PRs before landing on main. Required for this repo.

**Metrics (Service)**  
Counters/gauges like session\_count, child\_up, child\_restart\_count, and latency summaries. Emitted for dashboards and alerts.

**Mcp‑Session‑Id (Header)**  
Client‑provided or server‑issued session identifier. The proxy maintains **one child process per session** and echoes the header on responses.

**MCP‑Protocol‑Version (Header)**  
Optional header for future protocol negotiation/versioning in the transport layer.

**Migrations (DB)**  
Script files in /db/migrations named VYYYYMMDDHHMM\_\_<slug>.sql. Always additive; run via CI/CD tooling.

**No‑Hard‑Coding**  
Strict rule forbidding literals for dynamic behavior. All runtime‑varying values must come from DB config/feature flags or the platform vault.

**OpenAPI 3.1**  
Formal API contract in /api/openapi/mcp-proxy.yaml; declares servers for Alpha/Beta/RTM/Prod, bearer auth, and shared error envelope.

**Origin (Header)**  
Indicates the source of a cross‑origin request. The proxy validates it against Security:AllowedOrigins from DB; violations return 403.

**Perf Budget**  
Quantified latency and throughput expectations (e.g., TTFB ≤ 200 ms). Used in k6 test plans.

**PID (Child PID)**  
Process identifier for the spawned Kendo MCP child; logged for correlation and lifecycle management.

**Prod DB (Validation in RTM)**  
The production database visited during RTM for validation read‑only checks; ensures configuration parity before Prod promotion.

**Rate Limiting (Deferred)**  
Optional control to throttle client requests; captured as a future ADR and not part of MVP.

**RACI**  
Responsibility matrix: DocFactory (R), DoSE (A), T‑Arch (C), Dev/QA/Client Services (I).

**Readiness**  
System condition indicating the service can accept traffic (e.g., DB reachable, config loaded, child spawn succeeds when required).

**SARIF**  
Static analysis results format (e.g., CodeQL). Part of the Evidence Pack for releases.

**SBOM**  
Inventory of dependencies (SPDX/CycloneDX) generated in CI and retained with the release.

**Secret Scanning**  
Automated check in GitHub to detect credential leakage. Required check on PRs.

**Security:AllowedOrigins (Config Key)**  
CSV of allowed origins (e.g., https://chat.openai.com,https://platform.openai.com) stored in AppConfig; used for CORS checks.

**Session (Concept)**  
Isolated context keyed by Mcp-Session-Id. The proxy spawns and owns a dedicated child process per session.

**SSE (Server‑Sent Events)**  
Unidirectional event stream (text/event-stream) for delivering incremental messages, notifications, and heartbeats. Used by both streamed POST and GET channels.

**STDIO Bridge**  
Code that forwards JSON‑RPC messages to the child process via stdin and relays stdout lines back to HTTP clients (as JSON or SSE events).

**Streamable‑HTTP**  
HTTP pattern enabling incremental delivery of response data (here using SSE) and responsive first byte times.

**Testing Strategy (k6, axe, RTL)**  
k6 for perf budgets; axe smoke for A11y; React Testing Library for UI components (if UI present). Evidence attached to releases.

**ThemeBuilder (Kendo)**  
Tool for generating theme overrides from design tokens exported by Figma Make; applied to Fluent v12 base in the Ops UI.

**TTFB (Streaming First Byte)**  
Time to first streaming byte after request acceptance; target ≤ **200 ms**.

**/config/effective (Endpoint)**  
Read‑only view of current non‑secret config; used to diagnose environment drift. Values sourced from DB SPs and redacted where sensitive.

**/healthz / /ready (Endpoints)**  
Health: liveness (process up). Ready: dependencies OK (DB reachable, config loaded). Used for probes and dashboards.

**/mcp (Endpoint)**  
Primary transport. POST /mcp accepts a single JSON‑RPC message; responds with JSON or streams SSE when requested. GET /mcp opens an SSE channel for background notifications.

**/messages / /sse (Endpoints)**  
Legacy HTTP+SSE compatibility endpoints, gated by EnableLegacyHttpSse.

**HTTP Headers & Fields (Project Canonical)**

| **Header** | **Direction** | **Meaning** |
| --- | --- | --- |
| Accept | Request | Set to text/event-stream to request streamed SSE response. |
| Content-Type | Request/Response | application/json for JSON bodies; text/event-stream for SSE. |
| Mcp-Session-Id | Request/Response | Session correlation header; one child per session; echoed by server. |
| MCP-Protocol-Version | Request | Optional for future protocol negotiation. |
| Origin | Request | Validated against Security:AllowedOrigins; enforced CORS allow‑list. |

**Database Objects & Stored Procedures (Canonical)**

* **Tables**
  + AppConfig([Key] PK, [Value] NVARCHAR(MAX), [UpdatedAt] DATETIME2)
  + FeatureFlag([Name] PK, [Enabled] BIT, [UpdatedAt] DATETIME2)
* **Seed (non‑secret) Keys**  
  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
* **Stored Procedures (SP‑only DAL)**  
  sp\_Config\_GetValue(@Key) → NVARCHAR(MAX)  
  sp\_Config\_GetAll() → [Key],[Value]  
  sp\_Feature\_IsEnabled(@Name) → BIT  
  sp\_Lookup\_Get(@Type,@Key) → NVARCHAR(MAX) (reserved)

**DB COMPLIANCE:** Add‑only schema; **Stored‑procedure‑only** access; **No‑Hard‑Coding**. All dynamic behavior is driven by AppConfig/FeatureFlag/Lookup SPs; secrets live only in GitHub Environments or vendor portals.

**Roles & Responsibilities (Selected)**

* **DocFactory (Responsible)** — Generates/updates all SDLC documents and scaffolding.
* **DoSE (Accountable)** — Owns quality gates, approvals, and evidence.
* **Systems Architect (Consulted)** — Validates architecture and decisions.
* **Dev / QA / Client Services (Informed)** — Consume artifacts and implement features/tests.
* **Ops Admin** — Oversees health/metrics, promotions, and incident procedures.

**Artifacts & Evidence**

* **OpenAPI 3.1** (/api/openapi/mcp-proxy.yaml) — servers for Alpha/Beta/RTM/Prod; bearer; error envelope.
* **CI Evidence:** Build/test logs, **CodeQL SARIF**, **Dependency Review**, **Secret Scanning**, **SBOM**, OpenAPI lint/diff.
* **Monitoring Evidence:** p50/p95 latency, error rate, session metrics; **24‑hour post‑release checks**.
* **Retention:** All release evidence kept ≥ **1 year**.

**Related Documents**

* docs/01\_vision.docx — Vision & Objectives
* docs/05\_fr.md — Functional Requirements
* docs/06\_nfr.md — Non‑Functional Requirements
* docs/07\_data\_contracts.md — Data & DB Contracts
* docs/10\_ci\_cd.md — CI/CD Plan
* docs/11\_monitoring.md — Monitoring & SLOs
* docs/12\_evidence\_pack.md — Evidence Pack
* docs/13\_compliance.md — Security & Compliance
* runbooks/\*.md — Deploy, Rollback, Incident, Rotate License, Scale‑out
* adr/\*.md — Architecture Decision Records

**Appendix A — Naming & Formatting Quick Reference**

* **Stored Procedures:** sp\_<Area>\_<Verb>\_<Noun>  
  Examples: sp\_Config\_GetValue, sp\_Feature\_IsEnabled, sp\_Lookup\_Get.
* **Migrations:** VYYYYMMDDHHMM\_\_<slug>.sql  
  Examples: V202509230900\_\_init\_schema.sql, V202509230905\_\_seed\_appconfig\_featureflag.sql.
* **Headers (exact case):** Mcp-Session-Id, MCP-Protocol-Version, Accept, Origin, Content-Type.
* **File Layout (high‑level):**  
  /api/openapi, /db/migrations, /db/stored\_procedures, /docs, /runbooks, /tests/gherkin, /.github/workflows.

**Appendix B — Examples (Non‑secret)**

**Error Envelope (JSON)**

{ "code": "origin\_forbidden", "message": "Origin not allowed", "requestId": "req-123" }

**SSE Heartbeat Line**

: 2025-09-23T12:00:00Z

**SSE Message Event**

event: message

id: 7

data: {"jsonrpc":"2.0","id":"abc","result":{"status":"ok"}}

**Footer (optional for Word header/footer):**  
*MCPX‑KendoBridge • Glossary • Version 1.0.0 (Draft) • 2025‑09‑23 • Confidential — Technijian Internal*