**02 – Glossary (Technijian SDLC)**

This glossary standardizes terms across architecture, security, SDLC, UI, API, data, and operations for the **MCPX-KendoBridge Admin Portal**.

**A. SDLC & Governance**

* **Technijian SDLC** — Our delivery model: Sprint 0 discovery → iterative 2-week sprints → RTM gate → Prod. Requires evidence and quality gates at each promotion.
* **Sprint 0** — Foundation sprint to finalize scope, FR/NFR, context & data flows, OpenAPI shell, DB SP contracts, CI skeleton, and Figma→Kendo SOP.
* **Definition of Ready (DoR)** — User story has acceptance criteria, UX refs, data contracts, security notes, and test approach.
* **Definition of Done (DoD)** — Code, tests, security checks, docs, OpenAPI, and evidence artifacts complete and merged; a11y checks pass; monitoring updated.
* **Evidence Pack** — Release-bundled proof: build/test reports, CodeQL, Dependency Review, Secret Scanning summary, SBOM+attestation, OpenAPI diff/lint, monitoring snapshot, approvals. Retain ≥ 1 year.
* **RTM (Release-to-Manufacturing)** — Pre-Prod environment validating against Prod read-only data or equivalent parity to reduce “surprise” in Prod.
* **ADR (Architecture Decision Record)** — Small doc recording a decision, options considered, and trade-offs.
* **SoT (Source of Truth)** — Canonical owner of a datum (e.g., ThemeBuilder tokens for UI theming; DB for configuration).
* **Non-Goals** — Explicitly excluded items to keep scope focused.

**B. Identity, Security & Access**

* **Azure AD / Entra ID** — Identity provider for SSO.
* **MSAL (PKCE)** — Client-side library and authorization flow used by the web app to obtain tokens securely.
* **App Registration** — Azure AD application with redirect URIs, scopes, and app roles defined.
* **Access Token (JWT)** — Bearer token presented to the API; validated server-side.
* **ID Token** — Token with user identity claims; used by the web app for display/UX, not for API auth.
* **Refresh Token** — Long-lived token used (if enabled) to renew access tokens; often avoided in SPAs in favor of silent flows.
* **App Role / AAD Group** — Authorization constructs; we bind least-privilege admin roles to approved users via groups.
* **RBAC** — Role-based access control, enforced in the API and optionally echoed in the UI to hide unauthorized actions.
* **CORS Allow-List** — Per-environment list of allowed browser origins.
* **PoLP** — Principle of Least Privilege.
* **Error Envelope** — Standard API error shape: { code, message, details?, requestId }.
* **Rate Limiting** — Server policy limiting requests per principal/IP/session; violations return **429** with Retry-After.

**C. Frontend (Design → Code)**

* **Figma Make Prototype** — Design source; frames named by route (/login, /dashboard, /config, etc.).
* **ThemeBuilder Tokens** — Design tokens exported from Telerik ThemeBuilder; single source of truth for theming.
* **KendoReact (Fluent 2)** — Component library/theme used in code to replace prototype widgets 1:1 (Grid, Drawer, AppBar/Toolbar, Form, Inputs, Charts, Dialogs, Notifications).
* **Component Mapping** — Page-by-page mapping from Figma elements to KendoReact Fluent 2 components.
* **A11y / WCAG 2.2 AA** — Accessibility baseline enforced via automated axe checks and manual keyboard/focus review.
* **Client-Side Routing** — Route structure mirrors page frames; guards enforce auth and role checks.

**D. API & Integration**

* **Admin API** — .NET 8 service exposing JSON endpoints; supports **SSE** for streaming.
* **OpenAPI 3.1** — Machine-readable contract for endpoints, auth, errors, and schemas; validated in CI.
* **Bearer Auth** — API expects a valid JWT in Authorization: Bearer <token>.
* **SSE (Server-Sent Events)** — Unidirectional streaming over HTTP. Clients send Accept: text/event-stream; server emits event: / data: lines with heartbeats.
* **Heartbeat** — Periodic SSE event to keep connections alive and drive progress UI.
* **Idempotency-Key** — Optional header to safely retry operations without duplicating effects.
* **Health/Ready** — /healthz (process & dependencies) and /ready (service is ready to take traffic).

**E. Data & Persistence**

* **SQL Server 2022** — Primary relational store.
* **Add-Only Schema Rule** — Migrations never drop or modify columns in place; we add columns and deprecate at boundaries.
* **SP-Only DAL** — Application interacts with the DB exclusively via stored procedures (no ad-hoc SQL from code).
* **AppConfig / FeatureFlag / Lookup** — Canonical configuration tables accessed via SPs; no hard-coded config in code.
* **Audit Trail** — Immutable log of config mutations and significant actions (who/what/when).
* **Migration Naming** — VYYYYMMDDHHmm\_\_description.sql style (monotonic and auditable).
* **Effective Config** — The merged, environment-specific view of configuration the API serves to the UI for non-secret keys.

**F. Observability, Reliability & Performance**

* **SLI/SLO** — Indicators and objectives (e.g., p95 latency, error rate, TTFB for SSE).
* **TTFB (Time to First Byte)** — Target for first SSE event; baseline ≤ 200 ms for healthy paths.
* **p50/p95** — Latency percentiles; tracked per endpoint and surfaced on the dashboard.
* **Error Budget** — Allowable failure time before we pause feature work to stabilize.
* **Structured Logging** — JSON logs with requestId/correlation IDs and key fields (user, role, route, result).
* **Metrics** — Counters/gauges/histograms for requests, errors, queue depth, job durations, and resource usage.

**G. CI/CD & DevSecOps**

* **GitHub Actions** — CI for build/test/lint/a11y; CD for environment promotions with approvals and protections.
* **CodeQL** — Static analysis for C# and JS/TS code.
* **Dependency Review** — Checks for vulnerable or risky dependencies on PR.
* **Secret Scanning** — Ensures no secrets are committed; fails builds on detection.
* **SBOM (CycloneDX)** — Software bill of materials generated per build; signed/attested.
* **OpenAPI Lint/Diff** — Validates API changes and flags breaking deltas.
* **Axe CI** — Automated accessibility checks against critical flows.
* **CODEOWNERS / PR Template** — Guardrails that route reviews and enforce checklist items.

**H. Environments & Operations**

* **Alpha → Beta → RTM → Prod** — Promotion order; each stage increases confidence via traffic, data parity, and evidence.
* **Config Parity** — Evidence that RTM mirrors Prod-intended settings; differences are deliberate and documented.
* **Feature Flags** — Controlled rollout of capabilities per environment/tenant/user role.
* **Operational Runbook** — Playbooks for deploy, rollback, incident response, and common tasks.
* **Allow-List Change Procedure** — Controlled update process for CORS origins and API clients.

**I. UX Patterns & Components (KendoReact Fluent 2)**

* **Grid** — Tabular data with sorting/filtering/paging; tied to server queries.
* **Drawer / Sidebar** — Primary navigation.
* **AppBar / Toolbar** — Context actions, user menu, environment indicator.
* **Form & Inputs** — Validated forms with async submit and error summaries.
* **Dialog / Modal** — Confirmations, edit forms, error details.
* **Charts** — Health/perf visualizations (latency percentiles, error rate).
* **Notifications/Toasts** — Non-blocking status messages.
* **Skeletons/Placeholders** — Loading states while fetching data.

**J. Jobs & Streaming**

* **Job** — Long-running server action that emits progress events over SSE.
* **Job Lifecycle** — queued → started → zero-or-more progress → completed|failed.
* **Heartbeat Interval** — Minimum cadence for liveness updates when no progress events occur.
* **Job ID / Correlation ID** — Identifiers used to attach logs and metrics to a job instance.

**K. Compliance & Risk**

* **HIPAA / PHI** — Health data protections; ensure access control, audit, and minimum necessary principles when applicable.
* **PCI / FFIEC / ASVS** — Applicable security and financial controls referenced in risk assessments.
* **Data Classification** — Labels defining handling requirements (Public, Internal, Confidential, Regulated).
* **Retention** — Evidence and logs retained per policy (e.g., Evidence Pack ≥ 1 year).

**L. Frequently Used Abbreviations**

* **AAD** — Azure Active Directory (Entra ID)
* **ADR** — Architecture Decision Record
* **API** — Application Programming Interface
* **CI/CD** — Continuous Integration / Continuous Delivery
* **CORS** — Cross-Origin Resource Sharing
* **DAL** — Data Access Layer
* **DoD/DoR** — Definition of Done / Definition of Ready
* **FR/NFR** — Functional / Non-Functional Requirements
* **JWT** — JSON Web Token
* **MSAL** — Microsoft Authentication Library
* **RBAC** — Role-Based Access Control
* **RTM** — Release-to-Manufacturing
* **SBOM** — Software Bill of Materials
* **SLA/SLI/SLO** — Service Level Agreement/Indicator/Objective
* **SSE** — Server-Sent Events
* **TTFB** — Time to First Byte
* **WCAG** — Web Content Accessibility Guidelines

**Notes**

* Where this glossary conflicts with vendor documentation, the **Technijian SDLC** and our ADRs are authoritative.
* Terms marked as procedures (e.g., “Allow-List Change Procedure”) must be reflected in the **Operations Runbook** and referenced in the Evidence Pack for each release.