**Document: 04 – Functional Requirements**

**Project:** MCPX-KendoBridge Admin Portal  
**Document ID:** TJ-MCPX-DOC-04  
**Version:** 1.0.0  
**Status:** Draft  
**Date:** 2025-09-27  
**Owner:** Doc-Factory (Technijian)  
**Confidentiality:** Technijian Internal

**Document Control**

| **Version** | **Date** | **Author** | **Change Summary** | **Status** |
| --- | --- | --- | --- | --- |
| 0.1.0 | 2025-09-27 | Doc-Factory | Initial FR outline aligned to Vision & Use Cases | Draft |
| 1.0.0 | 2025-09-27 | Doc-Factory | Complete FR set with traceability & acceptance tests | Draft |

**Approvals**

| **Role/Team** | **Name** | **Signature/Date** | **Comment** |
| --- | --- | --- | --- |
| Director of Software Eng. |  |  |  |
| Systems Architect |  |  |  |
| Security/Compliance |  |  |  |
| DBA |  |  |  |
| QA Lead |  |  |  |

**1. Purpose**

Specify the **functional behavior** of the MCPX-KendoBridge Admin Portal and Admin API so that design, implementation, and testing can proceed consistently with the Technijian SDLC. This document is authoritative for *what* the system does (not *how fast* or *how secure*—see NFRs).

**2. Scope**

Covers the admin-only web application (React + **KendoReact Fluent 2**), the .NET 8 Admin API, Azure SSO integration, RBAC, configuration/flags/lookups (SP-only DAL), jobs with **SSE** streaming, audit trail, evidence viewing, access management (via Microsoft Graph, if enabled), CORS allow-list, and environment awareness (Alpha→Beta→RTM→Prod).

**3. References**

* TJ-MCPX-DOC-01 Vision Brief
* TJ-MCPX-DOC-02 Glossary
* TJ-MCPX-DOC-03 Actors & Use Cases
* OpenAPI 3.1 (Admin API)
* DB Stored Procedure Contracts (Config/Flags/Lookup/Audit)
* UI Component Map (Figma → **KendoReact Fluent 2**)
* DevSecOps & CI/CD Playbooks (CodeQL, Dependency Review, Secret Scanning, SBOM)

**4. Functional Requirements**

**Notation**: Requirements are uniquely identified as **FR-<Domain>-<ID>** and mapped to Use Cases (UC-xx) and verification checks. “Must” = mandatory; “Should” = recommended; “May” = optional.

**4.1 Authentication & Authorization (AUTH)**

* **FR-AUTH-01 (Must)** — The portal SHALL require **Azure AD (Entra ID) SSO** using **MSAL PKCE**; unauthenticated users are redirected to sign-in. *(UC-01)*
  + *Acceptance:* Given an unauthenticated user, when accessing /, then they are redirected to AAD; after sign-in, they land on /dashboard.
* **FR-AUTH-02 (Must)** — The API SHALL validate **JWT bearer** tokens (issuer, audience, signature, expiry) and extract **app role** / **group** claims for RBAC. *(UC-01)*
  + *Acceptance:* Requests without valid Authorization: Bearer receive 401 with standard error envelope.
* **FR-AUTH-03 (Must)** — Two roles SHALL exist: **Portal.Admin** (full admin) and **Portal.Viewer** (read-only). *(UC-01, UC-02, UC-03)*
  + *Acceptance:* Viewer cannot invoke any mutating endpoints; attempts receive 403.
* **FR-AUTH-04 (Should)** — Tokens SHOULD be stored in session (or secure cookie) and not persisted to localStorage.
  + *Acceptance:* Security test confirms no tokens are written to localStorage.

**4.2 Navigation & Shell (WEB)**

* **FR-WEB-01 (Must)** — The web app SHALL implement routes /login, /dashboard, /config, /flags, /lookups, /jobs, /audit, /evidence, /access, /security/cors, with an **environment badge** visible on all pages. *(UC-02..UC-12)*
* **FR-WEB-02 (Must)** — The UI SHALL use **KendoReact (Fluent 2 theme)** components mapped from the Figma Make prototype. *(Design SOP)*
* **FR-WEB-03 (Should)** — The shell SHOULD include a Drawer/Sidebar navigation, AppBar with user menu, and a global notifications area.

**4.3 Health & Readiness (HEALTH)**

* **FR-HEALTH-01 (Must)** — API SHALL expose GET /healthz and GET /ready returning component states and version info. *(UC-02)*
* **FR-HEALTH-02 (Must)** — The Dashboard SHALL render health, uptime, p50/p95, error rate, queue depth using Kendo Cards/Charts. *(UC-02)*

**4.4 Effective Configuration (CONFIG)**

* **FR-CONF-01 (Must)** — GET /config/effective SHALL return non-secret, merged configuration for the current environment, including requestId. *(UC-03)*
* **FR-CONF-02 (Must)** — The UI SHALL render a pageable/filterable **Kendo Grid** of effective config with export (CSV). *(UC-03)*
* **FR-CONF-03 (Must)** — All configuration consumed by the API/APP SHALL originate from **DB stored procedures** (SP-only DAL). *(UC-04)*
* **FR-CONF-04 (Must)** — Mutations SHALL be performed via endpoints (e.g., POST/PUT /config/keys) that call sp\_Config\_\*, and every change SHALL write an **audit** record. *(UC-04)*
* **FR-CONF-05 (Should)** — Requests MAY include Idempotency-Key to prevent duplicate mutations. *(UC-04)*

**4.5 Feature Flags (FLAGS)**

* **FR-FLAG-01 (Must)** — API SHALL provide CRUD endpoints for feature flags (/flags/\*) backed by sp\_Feature\_\*. *(UC-05)*
* **FR-FLAG-02 (Must)** — UI SHALL allow enable/disable and optional targeting by environment/role. *(UC-05)*
* **FR-FLAG-03 (Must)** — All flag changes SHALL be audited. *(UC-05)*

**4.6 Lookups (LOOKUP)**

* **FR-LOOK-01 (Must)** — API SHALL provide CRUD endpoints for lookup tables (/lookups/\*) backed by sp\_Lookup\_\*, following **add-only schema** (soft-delete/deprecate). *(UC-06)*
* **FR-LOOK-02 (Must)** — UI SHALL allow create/update/deprecate with validation and inline error surfacing. *(UC-06)*

**4.7 Jobs & Streaming (JOBS, SSE)**

* **FR-JOB-01 (Must)** — POST /jobs SHALL create a job and return a Job ID; **Idempotency-Key** supported. *(UC-07)*
* **FR-JOB-02 (Must)** — GET /jobs/{id}/events with Accept: text/event-stream SHALL emit **SSE** events: progress and **heartbeat**; terminal states completed|failed. *(UC-07)*
* **FR-JOB-03 (Should)** — Clients SHOULD be able to reconnect and resume (Last-Event-ID) when supported. *(UC-07)*
* **FR-JOB-04 (Must)** — All job state transitions SHALL be logged with correlation IDs; user-initiated actions SHALL be audited. *(UC-07)*

**4.8 Audit Trail (AUDIT)**

* **FR-AUD-01 (Must)** — API SHALL expose GET /audit with filters (actor, action, entity, date range) and export (CSV). *(UC-08)*
* **FR-AUD-02 (Must)** — The system SHALL capture **who/what/when/before→after** for config/flag/lookup mutations and admin actions. *(UC-04..UC-06, UC-09, UC-10)*

**4.9 CORS Allow-List (SECURITY/CORS)**

* **FR-CORS-01 (Must)** — API SHALL expose GET/PUT /security/cors to view/edit allowed Origins per environment. *(UC-09)*
* **FR-CORS-02 (Should)** — Edits SHOULD support an approval workflow (two-person rule) configurable per environment. *(UC-09)*
* **FR-CORS-03 (Must)** — All edits SHALL be audited. *(UC-09)*

**4.10 Access Management (ACCESS, Microsoft Graph)**

* **FR-ACCESS-01 (Must)** — If Graph write is enabled, API SHALL expose /access/\* to list users, view role/group membership, and **add/remove** users to **Portal.Admin** or **Portal.Viewer**. *(UC-10)*
* **FR-ACCESS-02 (Must)** — If Graph write is **not** enabled, UI SHALL generate a **Change Request** artifact with approver routing. *(UC-10)*
* **FR-ACCESS-03 (Should)** — UI SHOULD display propagation notes (user may need to re-login). *(UC-10)*

**4.11 Evidence Pack (EVIDENCE)**

* **FR-EVID-01 (Must)** — API SHALL expose GET /evidence/releases and GET /evidence/{releaseId} to list and download **Evidence Pack** artifacts (build/test reports, CodeQL, SBOM, OpenAPI diff, approvals). *(UC-11)*
* **FR-EVID-02 (Must)** — UI SHALL render releases in a Grid with artifact links and checksums/attestations. *(UC-11)*

**4.12 RTM⇄Prod Config Parity (PARITY)**

* **FR-PAR-01 (Must)** — API SHALL expose GET /config/parity?from=RTM&to=Prod returning a structured diff and severity tags. *(UC-12)*
* **FR-PAR-02 (Must)** — UI SHALL block promotion if **critical** deltas exist and surface remediation guidance. *(UC-12)*

**4.13 Error Handling & Rate Limiting (ERROR, RL)**

* **FR-ERR-01 (Must)** — All errors SHALL use a standard **error envelope** { code, message, details?, requestId }. *(UC-13)*
* **FR-ERR-02 (Must)** — The API SHALL implement **rate limiting** (per principal/IP/session) and return **429** with Retry-After on breach. *(UC-13)*
* **FR-ERR-03 (Should)** — The UI SHOULD render friendly messages and include requestId for support.

**4.14 Logging & Observability (OBS)**

* **FR-OBS-01 (Must)** — Every request SHALL include/emit a **correlation ID** (requestId) across UI/API/logs. *(Global)*
* **FR-OBS-02 (Must)** — The Dashboard SHALL present key SLIs (requests, errors, p50/p95, queue depth, uptime, version). *(UC-02)*

**4.15 Export & Reporting (EXPORT)**

* **FR-EXP-01 (Should)** — Grids (Effective Config, Audit, Evidence) SHOULD support CSV export with column selection. *(UC-03, UC-08, UC-11)*

**4.16 Accessibility (A11y) — *Functional UX behavior***

* **FR-A11Y-01 (Must)** — All interactive controls SHALL be keyboard operable and announce role/state to assistive tech. (NFR performance/contrast lives elsewhere.)
* **FR-A11Y-02 (Should)** — Pages SHOULD provide skip-to-content and visible focus indicators.

**5. UI Page Requirements (Figma → KendoReact Fluent 2)**

**Note:** All pages use ThemeBuilder tokens and the **Kendo Fluent 2** theme.

**5.1 Login**

* MSAL sign-in; environment badge; error surfaces for blocked access.

**5.2 Dashboard**

* Cards: Health/Ready, Version/Uptime, Queue Depth, p50/p95, Error Rate.
* Charts: Request rate & error % (time series).
* Actions: Quick links to Config, Flags, Jobs.

**5.3 Effective Config**

* Grid: key, description, current value, source, last changed (who/when), tags.
* Actions: Export CSV; open detail drawer.

**5.4 App Config Keys**

* Grid + “Add/Update” dialog with validation; audit trail panel.

**5.5 Feature Flags**

* Grid with toggle; targeting (env/role) control; audit panel.

**5.6 Lookups**

* Grid per lookup domain; add/update/deprecate; referential hints.

**5.7 Jobs**

* Create job form; job list; **Live stream** panel (SSE) with progress and heartbeats; retry guidance.

**5.8 Audit**

* Filter bar (actor, action, entity, date range); Grid; export; details drawer (before→after).

**5.9 Evidence**

* Releases list; artifacts with checksum/attestation; download.

**5.10 Access (Graph)**

* User search; role/group assignment controls (or Change Request flow).

**5.11 Security / CORS**

* Environment selector; Origins Grid; add/edit; approval workflow (if enabled); audit.

**6. API Contract Summary (Endpoints & Semantics)**

| **Area** | **Method & Path** | **Summary / Notes** |
| --- | --- | --- |
| Health | GET /healthz, GET /ready | Component states, version, readiness |
| Config | GET /config/effective | Non-secret merged config + requestId |
| Config | POST/PUT /config/keys | Create/Update keys (SP-backed, audited) |
| Flags | GET/POST/PUT/DELETE /flags/\* | CRUD on feature flags (SP-backed, audited) |
| Lookups | GET/POST/PUT/DELETE /lookups/\* | CRUD on lookups (SP-backed, add-only, audited) |
| Jobs | POST /jobs | Create job; supports Idempotency-Key |
| Jobs | GET /jobs/{id}/events (SSE) | Stream events (progress, heartbeat, final state) |
| Audit | GET /audit | Filterable audit log; export |
| Evidence | GET /evidence/releases | List releases |
| Evidence | GET /evidence/{releaseId} | Artifact listing / download |
| Access | GET/POST /access/\* | List/assign roles via Graph (if enabled) |
| CORS | GET/PUT /security/cors | View/edit allowed Origins (per env) |

**Common:** Bearer JWT required; standard error envelope; pagination (page, pageSize), sorting (sort), filtering via query.

**7. Audit Events (Canonical)**

| **Code** | **When** | **Attributes (min)** |
| --- | --- | --- |
| CONFIG\_UPDATED | Config key add/update | key, oldValue?, newValue, actor, requestId |
| FLAG\_TOGGLED | Feature flag enable/disable | flag, scope, oldValue, newValue, actor |
| LOOKUP\_CHANGED | Lookup add/update/deprecate | table, key, oldValue?, newValue, actor |
| JOB\_STARTED | Job created | jobId, type, paramsHash, actor |
| JOB\_COMPLETED | Job finished successfully | jobId, duration, resultHash |
| JOB\_FAILED | Job failed | jobId, errorCode, message |
| CORS\_UPDATED | Allowed origins edited | env, oldList, newList, actor |
| ACCESS\_CHANGED | Role/group assignment changed (if enabled) | user, oldRoles, newRoles, actor |
| LOGIN\_SUCCESS | User signed in | user, roles, correlationId |
| AUTHZ\_DENIED | Forbidden action attempted | user, action, reason, correlationId |

**8. Traceability Matrix**

| **FR ID** | **Use Case(s)** | **Endpoint(s) / UI Area** | **Verification (example)** |
| --- | --- | --- | --- |
| FR-AUTH-01/02/03 | UC-01 | Login / JWT validation | E2E login test; 401/403 contract tests |
| FR-WEB-01/02 | UC-02..UC-12 | App shell & routes | UI route tests; component presence checks |
| FR-HEALTH-01/02 | UC-02 | /healthz, /ready, Dashboard | Smoke tests; dashboard render test |
| FR-CONF-01..05 | UC-03, UC-04 | /config/\*, Config pages | Contract + audit snapshot tests |
| FR-FLAG-01..03 | UC-05 | /flags/\*, Flags page | Toggle test; audit written |
| FR-LOOK-01/02 | UC-06 | /lookups/\*, Lookup pages | CRUD tests; add-only enforcement |
| FR-JOB-01..04 | UC-07 | /jobs, /jobs/{id}/events | SSE TTFB & lifecycle tests |
| FR-AUD-01/02 | UC-08 | /audit, Audit page | Filter/export tests |
| FR-CORS-01..03 | UC-09 | /security/cors, CORS page | Edit → approval → audit |
| FR-ACCESS-01..03 | UC-10 | /access/\*, Access page | Assign/remove or Change Request path |
| FR-EVID-01/02 | UC-11 | /evidence/\*, Evidence page | Artifact presence & checksum checks |
| FR-PAR-01/02 | UC-12 | /config/parity | Critical drift blocks promotion |
| FR-ERR-01..03 | UC-13 | Global | Envelope conformance; 429 policy tests |
| FR-OBS-01/02 | UC-02, Global | Dashboard; logs | requestId propagation; SLI widgets present |
| FR-EXP-01 | UC-03, UC-08, UC-11 | Grid exports | CSV contents/headers verified |
| FR-A11Y-01/02 | Global | All pages | Keyboard nav & focus order tests |

**9. Assumptions & Constraints (Functional)**

* App roles/groups exist in AAD; assignments are maintained by IT or via the portal (if Graph enabled).
* All mutable data paths enforce **SP-only** DAL with audit logging.
* Figma Make is the design SoT; delivery uses **KendoReact Fluent 2** only (no shadcn/ui in production).
* Environments carry distinct CORS allow-lists; origin must match environment URL(s).

**10. Open Issues**

1. Define approval workflow details for CORS edits (one- vs two-person rule) and required approver roles.
2. Confirm whether Graph write is enabled in the tenant; if not, finalize the Change Request artifact fields and storage.
3. Confirm parity severity thresholds (which diffs are “critical” and block promotion).

**End of Document — TJ-MCPX-DOC-04 v1.0.0**

If you’re ready, say **proceed** and I’ll deliver **05 – Non-Functional Requirements (NFRs)** in the same controlled format.