**ADR‑0005 — RTM on Prod DB (Read‑Only) Parity Gate**

**Document:** docs/adr/0005-rtm-on-prod-db-readonly-gate.docx  
**Status:** **Accepted**  
**Date:** 2025‑09‑27  
**Project:** MCPX‑KendoBridge  
**Deciders:** DoSE (Accountable), SRE Lead, DBA Lead, Dev Lead, Security Lead  
**Consulted:** T‑Arch, QA Lead, DocFactory  
**Tags:** environments, parity, readiness, compliance, SP‑only, add‑only

**Guardrails (non‑negotiable):** GitHub‑first SDLC with merge‑queue and required checks; four environments **Alpha → Beta → RTM (validates on Prod DB read‑only) → Prod**; **Add‑only** schema; **Stored‑procedure‑only** data access; **No‑Hard‑Coding** of dynamic values (all runtime configuration from SQL via sp\_Config\_\*, sp\_Feature\_IsEnabled, sp\_Lookup\_Get); secrets live **only** in **GitHub Environments**.

**1) Context**

All dynamic behavior for MCPX‑KendoBridge (child process command/args/cwd, timeouts, SSE keep‑alive cadence, Origin allow‑list, feature flags) is **DB‑sourced via SPs** and must never be hard‑coded. To prevent configuration drift between staging and production, we require a **Release‑to‑Manufacturing (RTM)** environment that validates **against the Production database in read‑only mode** before any promotion to **Prod**. This ADR defines the environment semantics, gates, and verification required to make RTM a **hard parity gate**.

**2) Decision**

1. **RTM connects to Prod DB in read‑only mode.**
   * RTM uses a **distinct DB principal** with **EXECUTE‑only** on approved SPs (sp\_Config\_\*, sp\_Feature\_IsEnabled, sp\_Lookup\_Get) and **no table/view CRUD** privileges.
   * The RTM application **must not** perform writes; SPs are read‑only by contract.
   * The RTM connection string is provided as a **separate secret** (e.g., SQL\_CONNECTION\_STRING\_PROD\_RO) in the **rtm GitHub Environment**.
2. **Promotion is blocked if parity fails.**
   * The deploy.yml RTM job **must run**:  
     (a) **Readiness** (/ready),  
     (b) **Config parity**: /config/effective snapshot from RTM is compared to the **expected Prod values** (non‑secret), and  
     (c) **Contract tests** (OpenAPI lint/diff + required error examples).
   * Any drift or failed checks **fails the workflow** and blocks promotion to Prod.
3. **Evidence capture is mandatory.**
   * Attach RTM readiness output, config snapshot, parity diff, and contract test artifacts to the Release Evidence Pack; retain **≥ 1 year**.

**3) Options Considered**

| **Option** | **Pros** | **Cons** |
| --- | --- | --- |
| **RTM → Prod DB (read‑only)** *(chosen)* | Maximum parity; zero drift in DB‑sourced runtime values; catches mis‑seeds & allow‑list errors pre‑Prod | Requires secure network access; needs strict read‑only principal; careful ops |
| Dedicated **Prod‑like DB** snapshot | Safer isolation; no Prod network dependency | Drift re‑introduced the moment Prod config changes; snapshot staleness |
| RTM uses **its own DB** | Simpler to manage | High parity risk; misses Prod‑only flags/origins; undermines gate intent |

**We choose RTM→Prod (RO)** to eliminate config drift for DB‑sourced values while ensuring safety via read‑only permissions and SP‑only policy.

**4) Rationale**

* **DB‑sourced configuration** is our single source of truth; validating RTM against Prod DB prevents last‑minute surprises (e.g., Origin allow‑list mismatches, timeout mis‑tunes).
* **Add‑only** and **SP‑only** policies make RTM safe: SPs are read‑only and the app has **EXECUTE‑only** on those SPs.
* **Auditability:** parity artifacts (snapshots/diffs) become part of the Evidence Pack for regulatory and operational review.

**5) Implications & Constraints**

* **Networking/Security:** RTM requires controlled network access to the Prod DB endpoint; the RTM DB principal must be **least‑privilege** and **read‑only** in practice (EXECUTE‑only on SPs; no table CRUD).
* **Secrets handling:** the RTM DB connection secret is **not stored in DB or code**; it resides in the **rtm GitHub Environment** only.
* **Operational semantics:** /ready in RTM fails if SPs are unreachable. Legacy endpoints remain **flag‑gated** and **OFF** by default.

**6) Implementation (authoritative)**

**6.1 Environment secrets (names)**

| **Environment** | **Secret** | **Purpose** |
| --- | --- | --- |
| rtm | SQL\_CONNECTION\_STRING\_PROD\_RO | Connect RTM to **Prod DB (read‑only)** |
| alpha/beta/prod | SQL\_CONNECTION\_STRING | Environment‑local DB connections |
| any with UI | TELERIK\_LICENSE or TELERIK\_LICENSE\_PATH | **Build‑time only** license for Kendo UI |

**All secrets live in GitHub Environments; never in code/DB/logs or docs.**

**6.2 App configuration**

* RTM build (image) is identical to Beta/Prod; the only difference is the **connection secret** and the **API base URL**.
* The app uses **ADO.NET SP calls** with CommandType.StoredProcedure, async I/O, and 30s default timeout per the Test Strategy.
* The API endpoint /config/effective returns only **non‑secret** keys to support parity.

**6.3 CI/CD (deploy.yml — RTM job excerpt)**

# .github/workflows/deploy.yml (RTM stage excerpt)

rtm:

needs: beta

environment: rtm

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- name: Smoke: readiness & health

run: |

curl -fsS https://rtm.example.com/api/ready | jq .

curl -fsS https://rtm.example.com/api/healthz | jq .

- name: Snapshot non-secret config

run: |

curl -fsS https://rtm.example.com/api/config/effective | jq -S . > rtm-config.json

- name: Parity check vs expected Prod config

run: |

# expected-prod.json is committed as non-secret baseline or produced by a template action

jq -S . expected-prod.json > prod-expected.json

diff -u prod-expected.json rtm-config.json | tee parity.diff

- name: Fail on drift

run: |

if [ -s parity.diff ]; then

echo "❌ RTM↔Prod parity drift detected"; exit 1; fi

- name: Contract tests (OpenAPI lint/diff)

run: |

# Example placeholder; actual project uses your OpenAPI linter/differ

echo "Run redocly lint + diff here"

- name: Upload Evidence

uses: actions/upload-artifact@v4

with:

name: rtm-parity-evidence

path: |

rtm-config.json

parity.diff

*Note:* You may source expected-prod.json from a **templated policy** (non‑secret) that defines acceptable values or from a one‑time baseline. Secrets are never part of this snapshot.

**7) Parity Check Algorithm (non‑secret)**

1. **Fetch** RTM /config/effective JSON (sorted keys).
2. **Normalize**: trim strings, sort, and lower‑case comma‑delimited lists (Security:AllowedOrigins).
3. **Mask/ignore**: values marked as **environment‑specific** by policy (e.g., base URLs), but **never** secrets.
4. **Compare** to **expected Prod** configuration file; produce a unified diff.
5. **Fail** the RTM job if any non‑ignored key differs; attach parity.diff to Evidence Pack.

**Recommended keys to compare (initial set):**  
Mcp:ChildCommand, Mcp:ChildArgs, Mcp:ChildCwd, Security:AllowedOrigins, Network:SseKeepAliveSeconds, Network:RequestTimeoutSeconds, EnableLegacyHttpSse. (Adjust as the config surface evolves.)

**8) DBA: RTM Read‑Only Principal (pattern)**

Replace names per environment. Do **not** include secrets here; bind credentials out‑of‑band.

-- Create role with EXECUTE-only on approved SPs (reuse app\_sp\_execute if already present)

IF NOT EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = N'app\_sp\_execute' AND type = 'R')

CREATE ROLE [app\_sp\_execute] AUTHORIZATION [dbo];

GO

-- Grant EXECUTE on read-only SPs

GRANT EXECUTE ON OBJECT::dbo.sp\_Config\_GetValue TO [app\_sp\_execute];

GRANT EXECUTE ON OBJECT::dbo.sp\_Config\_GetAll TO [app\_sp\_execute];

GRANT EXECUTE ON OBJECT::dbo.sp\_Feature\_IsEnabled TO [app\_sp\_execute];

GRANT EXECUTE ON OBJECT::dbo.sp\_Lookup\_Get TO [app\_sp\_execute];

GO

-- Create RTM user (login created by ops tooling) and add to role

-- CREATE USER [mcp\_proxy\_rtm] FOR LOGIN [mcp\_proxy\_rtm\_login];

EXEC sp\_addrolemember @rolename = N'app\_sp\_execute', @membername = N'mcp\_proxy\_rtm';

GO

-- Auditor query: confirm only EXECUTE grants (no table CRUD)

SELECT r.name AS role\_name, p.permission\_name, o.name AS object\_name, o.type\_desc

FROM sys.database\_permissions p

JOIN sys.database\_principals r ON p.grantee\_principal\_id = r.principal\_id

LEFT JOIN sys.objects o ON p.major\_id = o.object\_id

WHERE r.name = N'app\_sp\_execute'

ORDER BY p.permission\_name, o.name;

This pattern matches our **SP‑only**/least‑privilege model and prevents RTM from mutating data.

**9) Testing & Evidence**

* **Contract tests:** Run OpenAPI lint/diff at RTM; no breaking changes.
* **Readiness/Health:** /ready and /healthz return 200 at RTM (validated against Prod DB).
* **Parity:** parity.diff is empty; otherwise fail.
* **Performance smoke:** Optional SSE TTFB spot check at RTM (same budgets as Prod).
* **Evidence Pack (retention ≥ 1 year):** Attach RTM /ready, /healthz, /config/effective, parity.diff, OpenAPI lint/diff outputs.

**10) Monitoring & Alerts**

* **Dashboards:** Display RTM availability, readiness, and **error codes** (e.g., not\_ready, feature\_disabled, origin\_forbidden).
* **Alerts:** Notify on RTM /ready failures and recurring parity failures (e.g., three consecutive release attempts).
* **Post‑release:** include an RTM panel screenshot in the Evidence Pack.

**11) Security & Compliance**

* **No secrets in DB.** Only non‑secret config values are present; secrets exist only in **GitHub Environments**.
* **No‑Hard‑Coding.** App reads **all** dynamic values via SPs; literals are prohibited.
* **SP‑only and Add‑only.** Enforced via grants and migration policy; RTM cannot write by contract.
* **Logs:** never include payload bodies or secrets; use the canonical **error envelope** { code, message, requestId? }.

**12) Risks & Mitigations**

| **Risk** | **Impact** | **Mitigation** |
| --- | --- | --- |
| RTM cannot reach Prod DB | RTM gate blocks promotion | Fix network path; temporarily validate via a **fresh Prod snapshot** with sign‑off; document exception in Evidence |
| Drift in allowed origins or timeouts | Prod failure post‑promotion | Gate fails on parity diff; correct DB config via **Config Rollback** (add‑only seed), re‑run RTM |
| Hidden writes in SPs | Data modification risk | Require DBA review; **read‑only** SP contracts; EXECUTE‑only role; SRE audit of SP bodies |
| Human error updating expected baseline | False positives/negatives | Generate baseline via policy template; review diffs; store baseline alongside Evidence |

**13) Backout Plan**

If RTM parity gate causes a release delay but Prod needs an urgent fix:

1. **Open an exception record** in the release (Evidence Pack) stating the risk and scope.
2. **Mitigate** any configuration mismatch directly in Prod DB (non‑secret keys) via **add‑only** seed while preserving audit logs.
3. Re‑run RTM parity; proceed only when parity passes. (Avoid bypassing the gate except for P1 incidents.)

**14) Related & Derived Artifacts**

* **ADR‑0001:** Transport choice (Streamable‑HTTP + SSE).
* **ADR‑0003:** Session‑per‑child & sticky routing.
* **ADR‑0004:** No‑Hard‑Coding, SP‑only, Add‑only DB.
* **OpenAPI 3.1:** api/openapi/mcp-proxy.yaml (servers: Alpha/Beta/RTM/Prod, error envelope).
* **Runbooks:** deploy / rollback / incident / scale\_out (RTM steps & parity checks).
* **DB Grants Appendix:** docs/07a\_db\_grants\_sp\_signatures.docx.
* **Evidence Pack:** docs/12\_evidence\_pack.docx.

**15) Acceptance Criteria**

* RTM uses **Prod DB (read‑only)** via a distinct environment secret and principal.
* App principal in RTM has **EXECUTE‑only** on approved SPs; **no** table CRUD rights.
* /ready, /healthz, /config/effective succeed at RTM.
* **Parity diff** is empty; any drift **fails** the RTM job and blocks promotion.
* RTM parity evidence is attached to the Release and retained **≥ 1 year**.

**Record maintained by DocFactory. Changes to RTM semantics r**