



Skillstinget — IT Administrator Guide

Skillstinget: build, approve, and distribute Microsoft 365 Copilot Cowork skills — tenant-resident.

Publisher: Team KRM · Version 2.2 · July 2026

1. What Skillstinget is

Skillstinget is a single-page web application for managing Copilot Cowork skills (**SKILL.md**) across their full lifecycle: authoring (with optional AI drafting), a corporate catalog with an approval workflow, one-click activation to users' OneDrive, and publishing of approved skills as Cowork plugins to the Microsoft 365 organizational app catalog.

2. Architecture and design

Tenant-resident by design. Skillstinget has no backend database. All organizational data lives in SharePoint lists inside *your* Microsoft 365 tenant, created automatically on first sign-in:

List	Purpose
EssentialsCoworkSkills	The skill catalog: content, versions, status (Pending/Approved/Rejected), publish tracking
EssentialsCoworkConfig	One JSON configuration item: approver group, catalog site, OneDrive path, AI settings, license key
EssentialsCoworkPublishLog	Append-only audit log of every org-catalog publish, update, and removal

Because the data is ordinary SharePoint content, it inherits your existing retention, eDiscovery, DLP, Purview labeling, backup, and data-residency posture with no additional configuration.

Delegated-only identity. Every call Skillstinget makes to Microsoft Graph runs as the signed-in user, via MSAL (authorization-code + PKCE) in the browser. The application holds no client secret, no application permissions, and no standing access. If a user cannot access a resource, neither can Skillstinget on their behalf.

Static hosting. The app itself is static HTML/JS (Azure Static Web Apps or any static host). The only outbound calls are to Microsoft identity/Graph endpoints and, if configured, your chosen AI endpoint and the Team KRM license service.

Multi-tenant model. One deployment serves any number of organizations. Each tenant's admin grants consent once (standard Entra admin-consent flow); the app then provisions and uses lists in that tenant only. No data ever crosses tenants.

3. Permissions requested (all delegated)

Permission	Why Skillstinget needs it	Used
User.Read	Sign-in and basic profile	Always
Sites.ReadWrite.All	Read/write the three Skillstinget lists	Always

Permission	Why Skillstinget needs it	Used
<code>Sites.Manage.All</code>	Create the lists on first run (provisioning only)	First run + schema upgrades
<code>Files.ReadWrite</code>	Write activated skills to the signed-in user's own OneDrive	On activation
<code>GroupMember.Read.All</code>	Check the signed-in user's membership of the approver group	Sign-in
<code>AppCatalog.ReadWrite.All</code>	Publish/update/verify/remove Cowork plugins in the org app catalog	Admins, requested on demand at first publish

Notes for reviewers: `Files.ReadWrite` is deliberately the self-scoped variant — the app cannot touch other users' drives. `AppCatalog.ReadWrite.All` is not part of the base consent; it is requested interactively the first time an approver publishes. There are no application (app-only) permissions anywhere.

4. Authorization model inside the app

Approver rights are gated on membership of an Entra security group you choose (checked via `checkMemberGroups`). Approvers see the admin area: approvals queue, published-skills tracking with catalog verification, configuration. Until a group is configured (bootstrap), all signed-in users are treated as approvers so first-time setup is possible — configure the group as your first action.

Known limitation (by design, on the roadmap): the app's rules are enforced in the app. Users with direct SharePoint access to the catalog site can edit list items outside the app. Mitigate by hosting the lists on a dedicated site with restricted membership (configurable catalog site URL) and enabling list versioning. A hardening option routing writes through a server-side function is planned for regulated environments.

5. AI drafting (optional)

Skill drafting with Claude uses, in priority order: your organization's AI proxy endpoint (recommended — key stays server-side), an organization Anthropic API key stored in the config list, or a personal key stored only in the user's browser. If the org key option is used, note it is readable by anyone with access to the config list — prefer the proxy for production. AI drafting sends only the skill description typed by the user; no tenant documents or data.

6. Licensing

Commercial deployments validate a per-tenant license key against the Team KRM license service. The check sends: product identifier, tenant ID, license key — nothing else. A 7-day grace period applies if the service is unreachable; unlicensed state disables publishing and approvals but never blocks reading or personal use.

7. Deployment models

Skillstinget ships in two models — same code, different trust boundaries:

SaaS (default). TeamKRM hosts the app; a multi-tenant app registration in the TeamKRM tenant serves every customer. Install = one admin-consent click; licensing validates online against the TeamKRM license service (tenant ID + key only), with a self-service 14-day trial. Right for most organizations.

Sovereign (self-hosted). Everything runs in *your* tenant with zero calls to TeamKRM: you host the static app, identity is a single-tenant app registration you create (script:

New-SkillstingetAppRegistration.ps1 — sets the SPA redirect and the exact delegated permissions, prints the client ID for **appconfig.json** with **multiTenant: false**, and licensing uses an **offline signed key (SKST-OFF....)** issued by TeamKRM: verified locally against an embedded public key — signature, tenant binding, and expiry checked with no network call. Updates arrive as versioned releases. Right for regulated environments that cannot depend on a vendor-controlled identity or accept outbound license calls.

8. Least-privilege operation

Sites.Manage.All exists only to create the three lists on first run. The app requests it only during provisioning (incremental consent), and after the lists exist an admin can remove it from the tenant's delegated grant with **Revoke-SkillstingetProvisioningPermissions.ps1** — everyday operation continues on **Sites.ReadWrite.All**. Re-granting (via the consent link) is only needed if lists must be re-provisioned. **AppCatalog.ReadWrite.All** is likewise requested on demand at first publish, never at sign-in, and **Files.ReadWrite** is the self-scoped variant that cannot touch other users' drives.

9. Deployment checklist

1. Entra app registration (SPA): redirect URI = the app origin; supported account types = single tenant, or any organizational directory for multi-tenant; delegated permissions per §3.
2. For multi-tenant use: complete Microsoft publisher verification (MPN) on the registration.
3. Host the static build; set **appconfig.json** (clientId, multiTenant, licenseEndpoint).
4. Admin consent: via the in-app link (multi-tenant) or Entra portal (single tenant).
5. First sign-in: lists provision automatically. Set the approver group in Config immediately.
6. **Tenant prerequisite for Cowork distribution:** migrate to *unified app management* (Teams admin center banner) — without it, published plugins remain Teams-catalog-only and do not surface in Cowork.
7. After each publish: deploy the plugin in Microsoft 365 admin center → Agents → Tools (auto-acquired by targeted users).

10. Offboarding

Remove the enterprise application (revokes consent tenant-wide), delete the three SharePoint lists, and remove any published plugins via Teams admin center or the app's "Remove from org catalog". Users' activated OneDrive skills are their own files and can be deleted by them (or via standard OneDrive administration). Nothing else exists.

11. Contact

TeamKRM · <https://teamkrm.com> · Publisher of Skillstinget.