

Annexe technique — architecture, accès, flux

Produit : Noa — application Microsoft Teams

Version : 1.0 — 7 septembre 2026

Aligné sur le code de `move-break-microsoft-teams-app` et la [politique de confidentialité](#) du 7 septembre 2026.

Document de premier échange. Il ne publie pas de secrets, de procédures d'administration internes ni de modes opératoires d'exploitation.

1. Composants

Composant	Rôle
Conversation 1:1 (bot)	Rappels, cartes Adaptive Cards, commandes (<code>stop</code> , onboarding, etc.)
Onglet Config	Horaires, zones, matériel, options, tuile Mes données (export / effacement / recréation)
Onglet Bibliothèque	Contenus à la demande, favoris, lecteurs vidéo
Backend Heroku	API HTTP, scheduler de rappels, jobs RGPD — offre actuelle
PostgreSQL Heroku	Persistance — offre actuelle
Azure Bot Service / Entra	Identité du bot , validation JWT des messages
Microsoft Teams (client)	Contexte utilisateur pour les onglets

Pas d'application mobile native Noa dans ce périmètre. Pas de Microsoft Graph.

Hébergement actuel vs ouvertures. Le tableau ci-dessus décrit la **production mutualisée** (Heroku + Postgres Heroku). MOVELYTICS est ouverte, par **avenant cadré** (pas une offre catalogue) :

- à un **autre PaaS** accepté par le client ; ou
- à une **instance dédiée** (même applicatif + PostgreSQL) sur des serveurs et une base **appartenant au client**.

Le bot Teams / Azure Bot Service n'est pas déplaçable hors de Microsoft. Un changement d'hôte implique un nouveau `validDomains` / URL d'onglets dans le manifeste.

2. Identifiants de l'application (production)

Élément	Valeur
ID manifeste Teams	<code>2a175964-2610-4c07-af50-d6a5d691c924</code>
ID bot (MicrosoftAppId)	<code>8bf8b0c0-4eb2-44a0-92a2-aba236e64e52</code>
Hôte HTTPS	<code>move-break-microsoft-teams-app-04b9b0a5156b.herokuapp.com</code>
Listing Store	WA200008766
Domaines valides (manifeste)	le hôte Heroku ci-dessus
Permissions RSC	aucune
<code>webApplicationInfo</code>	absent (pas de SSO d'onglet)

3. Authentification et habilitations

3.1 Canal bot

`POST /api/messages` est protégé, hors développement local, par `authorizeJWT` (jeton Azure Bot Service). Un appel sans jeton valide est rejeté. C'est le chemin le plus robuste.

3.2 Onglets Config et Bibliothèque

Les pages et API d'onglet s'appuient sur l'**identifiant utilisateur du contexte Teams** (`getContext` / `userId`). Il n'y a **pas** de mot de passe Noa, **pas** de cookie de session d'entreprise distinct, **pas** de jeton Entra d'onglet tant que la vérification éditeur n'est pas terminée.

Limite assumée (à ne pas dissimuler) : un `userId` connu pourrait théoriquement interroger des API d'onglet. Le risque est borné au graphe fonctionnel Noa (préférences, pauses), pas au SI Microsoft 365 du client (pas de Graph). Le SSO d'onglet est la mitigation prévue **après** MPN / domaine custom.

3.3 Référent organisation (tableau de bord tenant)

Un utilisateur marqué administrateur **du tenant** (`is_admin`) accède aux API de **pilotage agrégé** (volumes, taux, minutes, actions QVT). Il reçoit son propre nom/e-mail en en-tête de tableau de bord, pas l'historique nominatif des collègues.

3.4 Exploitation MOVELYTICS

Des comptes internes (`is_movebreak_admin`) servent au support et au suivi global. Les consultations sensibles sont journalisées (`admin_access_log` : identifiant technique, route, cible éventuelle). Effectif : trois personnes.

Les campagnes d'onboarding (messages d'introduction relayés dans Teams) sont des opérations **support**, limitées à des comptes habilités, hors du tableau de bord RH.

3.5 Licences

Le droit d'usage « entreprise » est porté par la table `tenant_module_licenses` (statuts `active` / `trialing` , dates). Ce n'est **pas** un abonnement Stripe dans Teams.

4. Surfaces HTTP utiles au RSSI

Surface	Auth	Commentaire
<code>POST /api/messages</code>	JWT Bot Framework	Canal unique des activités Teams
<code>/tabs/config</code> , <code>/tabs/bibliotheque</code>	Iframe Teams	UI
<code>GET /api/me/data-export</code>	<code>userId</code> contexte	Export RGPD
<code>POST /api/me/erase-account</code>	<code>userId</code> + confirmation <code>SUPPRIMER</code>	Effacement
<code>POST /api/me/recreate-account</code>	<code>userId</code>	Uniquement si tombstone <code>user_request</code>
API métier <code>/api/*</code> (préférences, bibliothèque, etc.)	<code>userId</code> contexte	Voir limite § 3.2
<code>GET /api/admin/tenant-dashboard</code>	admin tenant	Agrégats uniquement
Webhooks Notion historiques	Désactivés (410)	Plus de sync Notion

Les en-têtes de sécurité s'appliquent au backend : HSTS 180 jours, `nosniff` , CSP `frame-ancestors` (origines Teams / Office), pas de `X-Frame-Options: DENY` (casserait les onglets). Limitation de débit : 120 requêtes / minute / IP sur `/api/*` et `/admin/*` , **sauf** `POST /api/messages` .

5. Flux de données (création → usage → sortie)

1. **Déploiement DSI** : l'app est mise à disposition dans le tenant.
2. **Premier contact bot** : création d'une ligne `microsoft_users` (`userId` , `tenantId` , e-mail/nom Teams, `conversation_reference`).

3. **Rappels** : un planificateur lit les horaires et envoie un message proactif via la référence de conversation.
4. **Onglets** : lecture/écriture des préférences et de la bibliothèque pour ce `userId`.
5. **Tableau de bord client** : agrégats SQL par `tenantId`.
6. **Effacement utilisateur** : suppression du profil, détachement des historiques de pauses, tombstone. Recréation volontaire possible.
7. **Fin de contrat** : plus de licence entreprise active/trial → délai 90 jours → purge du tenant (tombstones `tenant_churn`, pas de recréation self-service).

Client payant = au moins une licence entreprise `active` ou `trialing` non expirée. Un salarié inactif n'est **pas** purgé tant que ce signal est vrai.

6. Journalisation (PII)

Journal	Contenu personnel
Erreurs bot / envoi de rappels	Identifiant technique ; pas e-mail, pas nom
Messages non reconnus	Hash SHA-256 + longueur, pas le texte
Accès admin interne	<code>admin_user_id</code> , route, cible éventuelle
Logs PaaS Heroku	Métadonnées techniques (durée de conservation selon l'hébergeur)

7. Traitements optionnels

Fonction	Condition	Prestataire
Move Invitation (ICS / e-mail)	L'utilisateur active l'option	Brevo SMTP
Vidéo Bibliothèque	L'utilisateur lance une vidéo	YouTube ou Vimeo (iframe)
Mood / humeur	Inactif	—

8. Dépendances applicatives notables

Stack volontairement courte : Express, `pg`, SDK Microsoft Agents Hosting, `node-cron`, `nodemailer` (Brevo). Pas de SDK Stripe, pas de SDK Graph dans le runtime actuel.

Les tests automatisés du dépôt couvrent notamment les en-têtes, la limitation de débit, l'export/effacement RGPD et la logique de purge tenant.

9. Ce que le client contrôle dans Teams

- Qui a l'application (déploiement org-wide, groupes, pilote).
- Politiques Teams / accès aux onglets / éventuel blocage d'iframes vidéo.
- Désinstallation org-wide (n'équivaut pas à une purge immédiate côté MOVELYTICS — passer par le DPA / la fin de contrat ou une demande d'effacement).

Le collaborateur contrôle : horaires, `stop`, export, effacement de **son** compte Noa.