La console NetByUs, en clair

30 septembre 2026 · ADAM Chacour · résumé du rapport « Console NetByUs : architecture, données et fonctionnement »

La console est le **back-office de NetByUs**. Elle vend les formules, suit les abonnements, active les modules chez chaque client et traite le support. Elle vit dans `api/platform/`, sur `admin.byplus.io` et la base centrale, et **ne voit jamais les données métier** des clients.

1. Les 6 idées à retenir

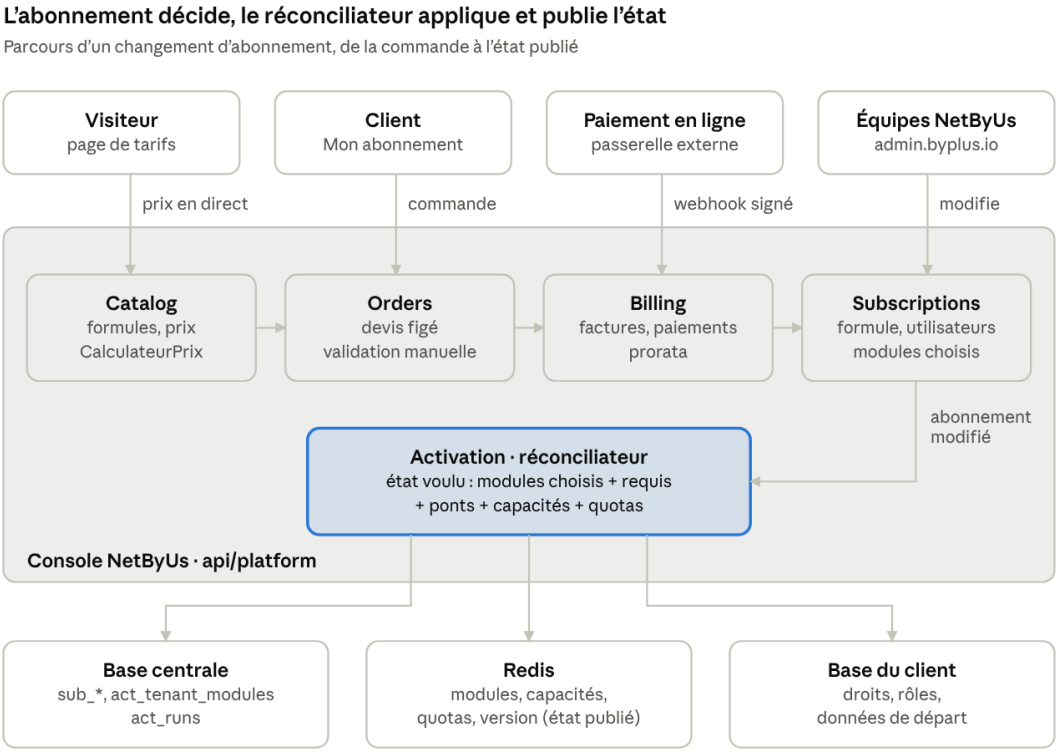
1. **Comme Odoo** : le client choisit une **formule**, un **nombre d'utilisateurs**, puis coche librement ses **modules**. `config/offers.php` disparaît.
2. **La console est une plateforme, pas un tenant.** Le tenant `netbyus` redevient un client ordinaire.
3. **L'abonnement est la source de vérité.** Il décrit ce que le client doit avoir ; un **réconciliateur** l'applique.
4. **Une seule sortie vers les clients** : l'état publié dans Redis (modules, capacités, quotas).
5. **Un seul calculateur de prix**, côté serveur, pour la page de tarifs, l'espace client, la console et le paiement.
6. **Les prix sont figés à l'achat** ; une nouvelle grille ne s'applique qu'au renouvellement.

2. Ce que l'on vend

Formule	Modules	En plus	Prix
Gratuit	1 module (parmi les éligibles) ; requis et ponts offerts	—	0 F, utilisateurs illimités
Standard	Tous les modules publiés	—	Par utilisateur et par mois
Entreprise	Tous les modules publiés	Multi-sociétés, API et webhooks, personnalisation avancée, extensions client	Par utilisateur et par mois

- **Capacités** : les fonctions avancées réservées à une formule (colonne « En plus »). La liste est technique ; la console décide qui y a droit.
- **Deux types d'utilisateurs** : **complet** (prix de la formule) et **léger**, au petit prix fixe, pour le libre-service (congé, note de frais, pointage, bulletin).
- **Options** payantes par quantité (caisse POS, porte Access, installation de passerelle), **promotions** (prix barré ou code) et **remise annuelle**.
- **Gratuit** : ajouter un deuxième module propose de passer en Standard.

3. Comment ça marche



De la commande à l'état publié

Tout passe par l'abonnement. Une commande payée, ou une modification faite par un agent, met à jour **Subscriptions**. Le **réconciliateur** calcule alors l'état voulu (modules choisis + requis + ponts + extensions, capacités, quotas), le compare à l'état réel, prépare la base du client, écrit la base centrale en dernier, puis republie les clés Redis `tenant:{cle}:*` et prévient les navigateurs. Chaque étape est rejouable ; un passage nocturne corrige les dérives.

4. Les domaines de la console

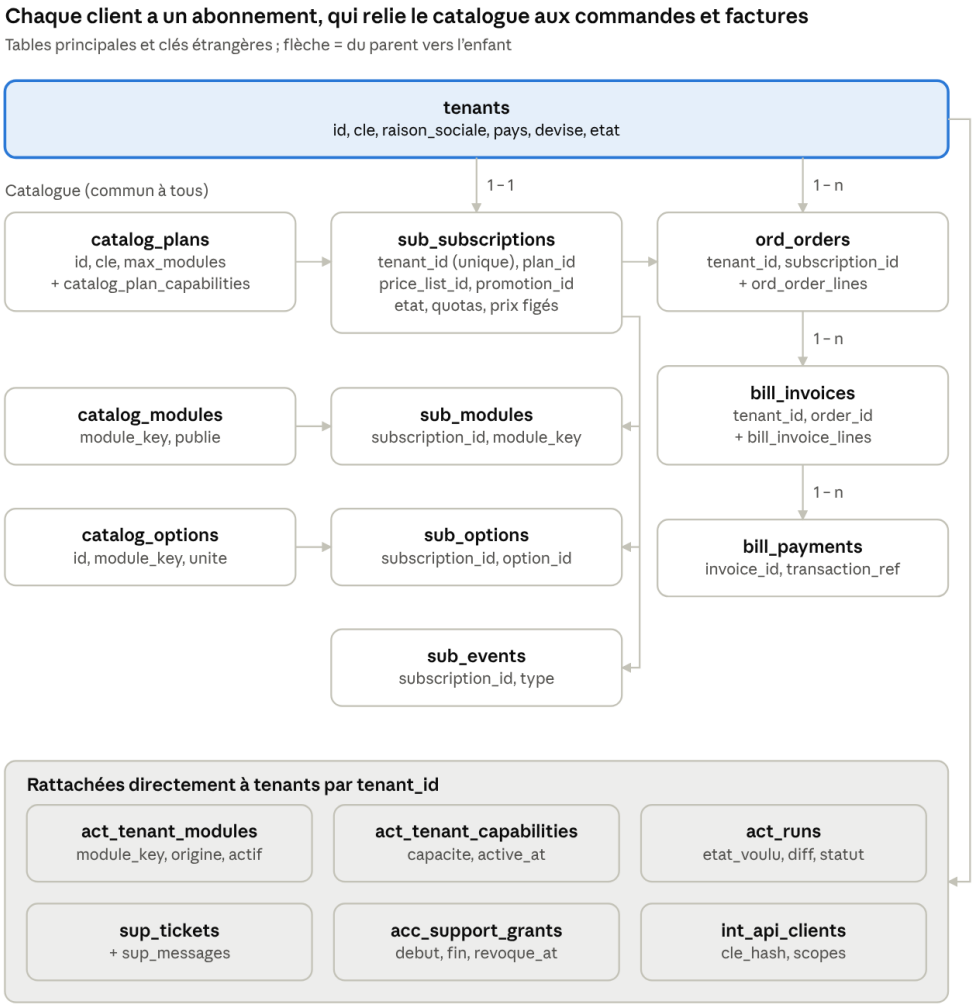
Un dossier par domaine, chacun avec l'anatomie d'un module (`Public/`, `Actions/`, `Models/`, `Http/`, `database/`, `tests/`). Le détail des tables est au chapitre 5.

Domaine	Ce qu'il gère	Tables
Staff	Comptes NetByUs, rôles, 2FA	`staff_*`
Tenants	Clients, domaines, bases, provisionnement	`tenants`, `tenant_*`
Catalog	Fiches des modules, formules, capacités, prix, options, promotions, calculateur	`catalog_*`
Subscriptions	Abonnement de chaque client : formule, utilisateurs, modules, historique	`sub_*`
Orders	Paniers, commandes, devis figés, validations manuelles	`ord_*`
Billing	Factures NetByUs, paiements, prorata, passerelles	`bill_*`
Activation	Réconciliateur, état réel des clients	`act_*`
Support · Access	Tickets ; accès temporaire du support chez un client	`sup_*`, `acc_*`
Integrations	Jetons de licence (JWT), clés API, webhooks sortants	`int_*`
Operations · Announcements · Audit	Migrations multi-bases, sauvegardes ; annonces ; journal central	`ops_*`, `ann_*`, `aud_logs`

Les clients parlent à la console par cinq guichets : **GuichetAbonnement**, **CalculateurPrix**, **GuichetSupport**, **GuichetAcces** et **GuichetIntegrations** (Entreprise). Le client est toujours déduit de la requête, jamais d'un paramètre du navigateur.

5. Structure des données : la base centrale

La base centrale compte **44 tables**, réparties par domaine ; chaque table porte le préfixe de son domaine. Montants en entiers (F CFA, sans décimales), identifiants en UUID ; toutes les tables ont `created_at` / `updated_at` (non répétés ci-dessous).



Relations entre les tables principales de la base centrale

Un client a un seul abonnement. Le catalogue est commun à tous et n'est référencé que par clé (`plan_id`, `module_key`, `option_id`).

Staff : équipes NetByUs

Table	Colonnes principales	Rôle
`staff_users`	id, nom, email (unique), password, totp_secret, actif, derniere_connexion_at	Comptes des équipes NetByUs, séparés des utilisateurs clients
`staff_roles`	id, cle (super_admin, commercial, support, finance, ops), nom	Rôles internes
`staff_role_permissions`	role_id, permission	Permissions par rôle (ex. abonnements.modifier)
`staff_user_roles`	staff_user_id, role_id	Rôles d'un agent

Tenants : clients

Table	Colonnes principales	Rôle
tenants	id, cle (ex. sdp), raison_sociale, pays, devise, email_contact, telephone, etat (provisionnement, pret, archive), cree_par	Le client
tenant_domains	id, tenant_id, domaine (unique), principal	Sous-domaines (sdp.byplus.io)
tenant_databases	tenant_id, nom_base, version_schema, derniere_migration_at, etat	Base du client et son niveau de migration
tenant_contacts	id, tenant_id, nom, email, telephone, role (facturation, technique, decideur)	Interlocuteurs côté client

Catalog : ce qui se vend

Table	Colonnes principales	Rôle
catalog_categories	id, cle, nom, ordre	Familles affichées (Ventes, Finance, RH…)
catalog_modules	module_key (= clé du manifeste), categorie_id, nom_affiche, description_courte, icone, publie, eligible_gratuit, validation_requise, ordre	Fiche commerciale d'un module
catalog_plans	id, cle (gratuit, standard, entreprise), nom, max_modules (null = illimité), actif, ordre	Formules
catalog_plan_capabilities	plan_id, capacite	Capacités données par une formule
catalog_price_lists	id, version, devise, effective_from, effective_to, publiee	Grille tarifaire versionnée
catalog_prices	id, price_list_id, cible_type (plan, utilisateur_leger, option), cible_id, periodicite (mensuel, annuel), montant_unitaire	Prix de chaque élément dans une grille
catalog_options	id, cle, module_key, nom, unite (caisse, porte, installation), recurrente	Options payantes
catalog_promotions	id, nom, code (null = automatique), type (pourcentage, montant), valeur, plans (json), debut, fin, utilisations_max, utilisations	Prix barrés et codes promo

Subscriptions : ce que chaque client a acheté

Table	Colonnes principales	Rôle
sub_subscriptions	id, tenant_id (unique), plan_id, price_list_id, periodicite, etat, debut, fin_periode, fin_essai, fin_grace, utilisateurs_complets, utilisateurs_legers, prix_utilisateur_fige, prix_leger_fige, promotion_id, renouvellement_auto	L'abonnement du client, prix figés
sub_modules	id, subscription_id, module_key, ajoute_at, retrait_prevu_at, retire_at	Modules choisis par le client (sans prix)
sub_options	id, subscription_id, option_id, quantite, prix_unitaire_fige, ajoute_at, retire_at	Options souscrites
sub_extensions	id, subscription_id, extension_key, contrat_ref, debut, fin	Extensions client attribuées
sub_events	id, subscription_id, type (cree, formule_changee, module_ajoute, etat_change…), avant (json), apres (json), acteur_type, acteur_id, created_at	Historique, en ajout seul

Orders : commandes

Table	Colonnes principales	Rôle
ord_orders	id, tenant_id, subscription_id, source (client, console, public), etat (brouillon, a_payer, a_valider, payee, appliquee, annulee), devis_json, total, valide_par	Demande de changement avant application
ord_order_lines	id, order_id, type (plan, utilisateurs, module, option, remise, prorata, taxe), cible, quantite, prix_unitaire, montant	Lignes du devis figé

Billing : facturation NetByUs

Table	Colonnes principales	Rôle
bill_invoices	id, numero (unique), tenant_id, order_id, periode_debut, periode_fin, sous_total, remise, taxes, total, etat (emise, payee, en_retard, annulee), echeance, pdf_path	Factures de NetByUs à ses clients
bill_invoice_lines	id, invoice_id, libelle, quantite, prix_unitaire, montant	Lignes de facture
bill_payments	id, invoice_id, passerelle (fedapay, kkiapay, virement…), transaction_ref (unique), montant, etat, recu_at, payload (json)	Paiements
bill_gateway_events	id, passerelle, event_ref (unique), type, payload (json), traite_at	Webhooks entrants, pour l'idempotence

Activation : état réel des clients

Table	Colonnes principales	Rôle
act_tenant_modules	tenant_id, module_key, origine (choisi, requis, pont, extension), actif, active_at, desactive_at	Modules réellement actifs, avec leur origine
act_tenant_capabilities	tenant_id, capacite, active_at	Capacités réellement actives
act_runs	id, tenant_id, declencheur, etat_voulu (json), diff (json), statut (en_cours, ok, echec), erreur, debut, fin	Chaque passage du réconciliateur

Support, accès, intégrations

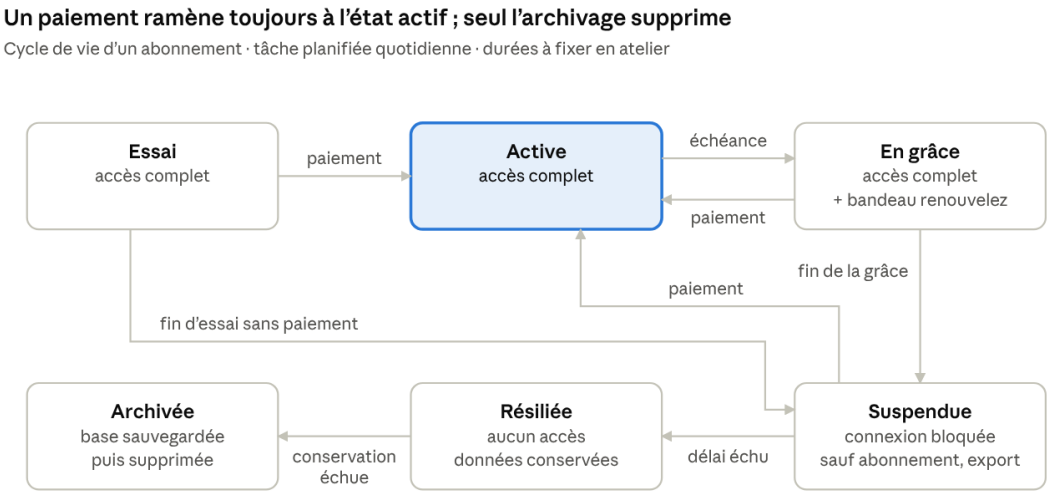
Table	Colonnes principales	Rôle
sup_tickets	id, numero, tenant_id, auteur_user_id (id dans la base client), auteur_nom, sujet, module_key, priorite, statut, assigne_a, premiere_reponse_at, resolu_at	Tickets
sup_messages	id, ticket_id, auteur_type (client, staff), auteur_id, corps, interne	Échanges ; interne = note invisible du client
sup_attachments	id, message_id, chemin_s3, nom, taille	Pièces jointes
acc_support_grants	id, tenant_id, accorde_par_user_id, ticket_id, debut, fin, revoque_at	Accès temporaire du support, accordé par le client
acc_support_sessions	id, grant_id, staff_user_id, debut, fin, ip	Chaque entrée du support chez un client
int_api_clients	id, tenant_id, nom, cle_hash, scopes (json), dernier_usage_at, revoque_at	Applications externes (mobile, passerelle Access)
int_webhook_endpoints	id, tenant_id (null = interne NetByUs), url, secret, evenements (json), actif	Destinations des webhooks sortants

Table	Colonnes principales	Rôle
`int_webhook_deliveries`	id, endpoint_id, evenement, payload (json), tentative, statut_http, prochain_essai_at, livre_at	Suivi des envois

Operations, annonces, audit

Table	Colonnes principales	Rôle
`ops_migration_runs`	id, version_cible, lance_par, debut, fin, statut	Migration lancée sur toutes les bases
`ops_migration_run_tenants`	run_id, tenant_id, statut (en_attente, ok, echec), erreur	Avancement par client
`ops_backups`	id, tenant_id, chemin, taille, statut, fait_at	Sauvegardes suivies
`ann_announcements`	id, titre, corps, type (nouveaute, maintenance), cible (json : formules, modules, clients), debut, fin	Annonces
`ann_reads`	announcement_id, tenant_id, user_id, lu_at	Lectures
`aud_logs`	id, acteur_type, acteur_id, action, cible_type, cible_id, tenant_id, avant (json), apres (json), ip, created_at	Journal central, en ajout seul

6. Cycle de vie et changements



Un paiement ramène toujours à l'état actif ; seul l'archivage supprime

Changement	Quand	Paiement
Monter de formule	Immédiat	Prorata jusqu'à l'échéance
Descendre de formule	À l'échéance (en Gratuit, le client choisit le module gardé)	Aucun remboursement
Ajouter un module	Immédiat	Inclus dans la formule
Retirer un module	À l'échéance ; refusé s'il est requis ; données conservées	—
Ajouter des utilisateurs ou options	Immédiat	Prorata jusqu'à l'échéance
Retirer des utilisateurs ou options	À l'échéance	Aucun remboursement

Au-delà du quota d'utilisateurs, la création est refusée avec un lien vers « Mon abonnement ».

7. Sécurité

- **Comptes NetByUs nominatifs**, séparés des comptes clients, avec **2FA obligatoire** (et IP ou VPN pour super_admin, finance, ops).
- **Rôles** : commercial, support, finance, ops, super_admin. **Double validation** pour l'archivage, la suppression de données, une nouvelle grille tarifaire ou une grosse remise.
- **Entrer chez un client** : seulement avec un accès accordé par son administrateur, limité dans le temps, tracé des deux côtés.
- **Paiements et webhooks** : signatures vérifiées, dédoublonnage, envois signés HMAC. **Jetons de licence** JWT vérifiables hors ligne par la passerelle Access.
- **Journaux** (`aud_logs`, `sub_events`, `sup_messages`) en ajout seul.

8. Plan de construction

9 lots, dans l'étape 2 du redémarrage. Le lot 3 est le critère de sortie de cette étape.

Lot	Contenu	Terminé quand
1	Staff + Audit	Un agent se connecte et chaque action est tracée
2	Tenants + provisionnement	La console crée un client, sa base et son domaine
3	Catalog + Subscriptions + Activation + Redis	Un abonnement attribué produit le bon état publié
4	Calculateur de prix, grilles, promotions	Tous les cas de prix calculés juste
5	Orders + GuichetAbonnement + page de tarifs	Un visiteur compose sa formule et voit le prix en direct
6	Billing + paiement + cycle de vie	Un inconnu paie et obtient son abonnement seul
7	Support + accès support	Ticket traité dans la console ; accès tracé
8	Integrations (JWT, webhooks)	La passerelle Access vérifie une licence hors ligne
9	Operations + Announcements, bascule de `netbyus`	Migration multi-bases suivie ; `netbyus` est un client ordinaire