External Brokerage Access — Recommended Approach

How external sales agencies should access developer inventory, reservations and sales — without creating separate CRM agencies per brokerage.

1 THE SCENARIO

Hirat Real Estate Development has no internal sales team. All J-Haus Residences sales are made through **15 external brokerage agencies** (expected to grow beyond 30). Today, Hirat shares its inventory with them via Excel sheets. The goal is to replace Excel with direct CRM access: agencies view live inventory, generate sales offers, create reservations, and track sales and commissions — while Hirat approves reservations, finalizes sales and issues invoices from the Developer platform.

Key fact: none of these brokerages are Optima customers. Optima would provide their access (web + mobile) free of charge, including support.

2 THE APPROACH UNDER CONSIDERATION

Create a **separate agency environment** in Optima CRM for each brokerage, create their users inside it, enable the MLS module, and have Hirat share J-Haus inventory to each agency via MLS. Agencies then work from their own environment.

3 WHY THIS APPROACH DOES NOT SCALE

Problem	Impact
15–30+ empty CRM tenants	Each brokerage gets a full CRM environment whose only purpose is receiving Hirat's inventory. Leads, deals, settings and most modules sit unused — yet every environment must be provisioned, configured, supported and updated, free of charge.
Heavy onboarding per agency	Every new agency Hirat signs becomes a setup project for Optima instead of a 5-minute record entry by Hirat's coordinator.
Duplicate identities	A brokerage working with two developers on Optima would need two agency records and its agents two logins — duplicated data, duplicated maintenance.
Users tightly bound to agency/office	In the current CRM structure, users live inside an agency and office. An external agent doesn't belong to Hirat's offices, and creating fake agencies just to hold users misuses the model.
MLS designed for peer-to-peer sharing	MLS assumes two parties who each genuinely operate their own Optima environment. That is not this case: agencies need a *window into Hirat's data*, not a CRM of their own.

Bottom line: the separate-agencies approach multiplies infrastructure and support linearly with every agency, and breaks the moment one brokerage works with a second developer.

4 HOW SALESFORCE & ZOHO SOLVE THIS

Salesforce — Partner Portal (Experience Cloud)

A partner company never gets its own Salesforce org. The developer's org holds the partner as an **Account with Contacts**. A contact is "enabled" as a portal user — a restricted login **inside the same org**. The partner sees a branded, stripped-down portal showing only records related to their own company.

Maps to Service Provider record + contacts with login enabled.

Zoho — One Login, Many Organizations

The login (email + password) lives at **platform level**, separate from any organization. The same login can be a member of multiple orgs and the user switches between them from their profile. Data never mixes between orgs.

Maps to Central login + per-developer access grants + workspace switcher.

Procore applies the same pattern in construction: subcontractors receive scoped logins into each general contractor's environment — no one creates an environment per subcontractor.

5 PROPOSED SOLUTION

Combine both models: **Salesforce's portal experience** on the surface, **Zoho's central login** underneath. Every user is split into two parts:

- **Login (who you are)** — email + password, stored once in a **central platform-level identity table**. Like a UAE Pass: one identity for the whole platform.
- **Access (what you can see)** — grants stored per developer environment. Each developer independently decides whether that identity may enter and what it may see.

User	Login stored	Access stored	What they see
Ripa (Hirat coordinator)	Central table	Inside Hirat — agency & office bound, exactly as today	Full CRM per her role. No visible change.
Ahmed (agent, Dream Homes brokerage)	Central table	Partner grant in Hirat, tied to the Dream Homes Service Provider record	Hirat-branded portal: shared inventory, own clients, own offers, reservations, commissions. Nothing else.

> **No new agencies are created.** 15 brokerages = 15 Service Provider records with contacts. Enabling access is a single "Enable login" action on the contact. Agency #16 or #31 costs Hirat's coordinator five minutes — and costs Optima nothing.

Inventory sharing becomes a per-Service-Provider visibility setting inside Hirat's project (e.g. share only 1st-floor units, or specific units, to a given agency) — reusing the MLS filtering concepts without cross-tenant machinery. Existing role-based rules in the Reservation and Sales status machines already assume this partner user type.

6 ARCHITECTURE AT A GLANCE

PLATFORM IDENTITY LAYER — ONE LOGIN PER REAL PERSON

ripa@hirat.ae
Hirat staff identity

ahmed@dreamhomes.ae
Broker identity — exists once

sara@dreamhomes.ae
Broker identity — exists once

↓ access grant   ↓ grant   ↘ grant   ↓ access grant

DEVELOPER ENVIRONMENTS — DATA FULLY ISOLATED PER TENANT

HIRAT ENVIRONMENT
Agency → Offices → Staff users (unchanged)

Partner access
SP: Dream Homes · Ahmed, Sara

Sees: Hirat branding · J-Haus shared units · own clients, offers, reservations, commissions

MARINA DEVELOPMENTS ENVIRONMENT
(future 2nd developer)

Partner access
SP: Dream Homes · Ahmed

Sees: Marina branding · Marina inventory only. Hirat data never visible here.

7 WALKTHROUGH — HIRAT GOES LIVE TODAY

1. **Ripa creates the Service Provider** "Dream Homes Real Estate" and adds its sales agents as contacts — exactly the module that exists today.
2. **Ripa clicks "Enable login"** on Ahmed's contact. The system creates his identity in the central table and grants him partner access into Hirat, tied to the Dream Homes SP record.
3. **Ahmed receives an invite email**, sets his password, and opens the app (web or mobile).
4. **Ahmed lands directly in Hirat's branded portal** — one developer, so no switcher appears. He browses shared J-Haus inventory, generates offers, requests reservations. Hirat approves, finalizes sales, issues invoices.

8 SCENARIO A — A SECOND DEVELOPER JOINS OPTIMA

Marina Developments becomes an Optima developer customer and also works with Dream Homes.

1. Marina's coordinator creates **their own** Dream Homes Service Provider record, adds Ahmed as a contact and clicks "Enable login" with **ahmed@dreamhomes.ae**.
2. The system finds the email already in the central table — **no second login is created**. It simply adds a second access grant: Ahmed → Marina.
3. Ahmed still has one email, one password, one app. On next login he sees a chooser: **[Hirat] [Marina]**. Each workspace shows only that developer's branding and data.
4. Neither developer sees the other's inventory, clients or deals — and neither knows what business Ahmed does with the other.

> **With the old model**, Ahmed would need a second account under a different email, two passwords and duplicated records. With central login, developer #2 is a non-event: one new row in the grants table.

9 SCENARIO B — THE BROKERAGE BECOMES AN OPTIMA CUSTOMER

Dream Homes later subscribes to the Optima brokerage platform for its own business (own leads, listings, secondary market).

- A full **Dream Homes agency environment** is created — a real paying tenant, with offices and users bound to it as in the standard CRM structure.
- Ahmed's **same identity** gains a third grant: owner/member of his own agency. His chooser now reads: **[Dream Homes — My CRM] [Hirat — Partner] [Marina — Partner]**.
- No migration, no duplicate users, no special flows. "Full tenant member" and "partner guest" are just two types of access grant on the same identity.

10 DECISION NEEDED & NEXT STEPS

#	Item	Owner
1	Confirm authentication can move to a **central platform-level identity table**, while all authorization (roles, office visibility, partner scoping) stays inside each tenant. This is the only structural change — everything else reuses existing modules.	Backend
2	Add **"Enable login"** to Service Provider contacts + invite flow. Enabling auto-creates a **Partner user** record in the Users module (see p.5); revoking removes it.	Product / Backend
3	Define the **partner role permission matrix** (agency agent vs. Hirat coordinator vs. admin) across inventory, offers, reservations, sales, commissions.	Product
4	Per-Service-Provider **inventory visibility settings** (floor / unit-level sharing) inside project configuration.	Product
5	Workspace **switcher UX** (hidden while an identity holds a single grant).	Product / Frontend

> **Recommendation:** proceed with the partner-access model inside Hirat's environment. Do not create separate agencies or use MLS cross-tenant sharing for external brokerages. Build logins on the central identity table from day one so that developer #2 — and any brokerage upgrading to a full Optima customer — requires zero rework.

11 WHERE PARTNER USERS FIT IN THE USERS MODULE

Today every agency has a **Users module**: users are created inside the agency, bound to an office, given a role (Admin, Senior Manager, Manager, Agent Pro), a "Reports to" line, commission tiers and preferences. **All access, roles and permissions are fetched from this module.** That stays true — with one addition.

The Users module becomes home to **two kinds of user**:

	Staff user (today's user)	Partner user (new)
Who	Hirat employees — Ripa, coordinators, admins	External agency contacts with login enabled — Ahmed from Dream Homes
Created in	Users module, as today	Never created here. Auto-created by **"Enable login"** on the Service Provider contact; auto-removed on Revoke
Bound to	Agency → Office (office hierarchy unchanged: master office sees sub-office data)	The Service Provider record — no office binding. Data visibility rule: records where provider = theirs
Role	Existing roles, unchanged	One new role class: **Partner Agent** — a fixed, limited permission set (shared inventory, own clients, own offers/reservations/sales, own commissions)
Reports to	As today	Not applicable — hierarchy lives at the agency's side, not Hirat's
Commissions tab	As today (brokerage platform)	Hidden — commission terms live on the Service Provider (agency defaults), not per user
Credential	Both: password lives in the **central identity table**, keyed by email. The user record stores no password. Recovery is self-service via Forgot password on the login screen only.	
Managed from	Users module	Service Provider module (Contacts & Access tab). The Users module **shows** partner users — read-only row with a link to the SP contact — so admins audit "who can access my environment" in one list

> **Why surface partner users in the Users module at all?** Because it is the single source the permission engine reads from. Under the hood every login needs a user record for authorization to work — partner users get one automatically, flagged **user_type = partner**. Salesforce works the same way: portal users appear in user management with a restricted license, but are enabled from the Contact, never created directly.

12 WHAT CHANGES AT THE USERS LEVEL

#	Change
1	Add **user_type** ('staff' | 'partner') to the user record. Staff behavior is untouched.
2	Partner user records carry **service_provider_id** instead of office_id; visibility scoping reads from it.
3	New **Partner Agent** role with a fixed permission set; not assignable to staff, not editable per user.
4	Partner rows in the Users list are **read-only**: no edit, no role change, no office assignment. The only actions are "Open provider contact" and "Revoke access" (both round-trip to the SP module).
5	Credentials move to the central identity table for **all** users — staff included. User ID stays the email. No password fields or admin resets anywhere in the Users module.
6	Users list gains a **Type** filter (Staff / Partner) and a context-aware column: Office for staff, Provider for partners.

> **Net effect:** one permission engine, one users list, two lifecycles. Staff managed in Users as always; partners managed from the Service Provider module and merely visible in Users. Nothing about the existing office hierarchy, roles, or staff flows changes.