

Odoo OCA Module ↔ Matrix Chat Integration — Requirements (v18)

Audience: Students building a first, working OCA-style connector. **Goal:** Post Odoo case/hearing updates to Matrix rooms and (optionally) mirror Matrix replies into the Odoo chatter. **License:** AGPL-3 • **Edition:** Odoo 18 Community

1) Scope (What we ship in MVP)

- **Outbound notifications** from Odoo to Matrix when key events happen (new Case, stage change, new Hearing, new message in chatter if flagged).
- **Room linking:** link one Matrix room per Case (manually paste Room ID or create via API).
- **Simple message composer** on Case to send manual messages to the linked room.
- **Contact list (read-only sync) & quick DM:** fetch Matrix contacts and let users start a 1:1 DM from Odoo.
- **Health check** action to verify homeserver and token.

Out of scope (MVP): SSO, user-by-user DM sync beyond basic directory/recents, file uploads, encryption management, typing/reads, advanced webhook signatures, multi-room per case. Optional inbound sync listed as Stretch.

2) User Roles

- **Matrix User:** can send outbound case messages; configure room on cases they manage.
 - **Matrix Manager:** configure homeserver, tokens, and defaults; run health checks; see logs.
-

3) Data Model (Minimal)

Model	Purpose	Key Fields
<code>matrix.backend</code>	Homeserver config	<code>name</code> , <code>homeserver_url</code> , <code>personal_access_token</code> (Secret), <code>default_sender_display</code> , <code>verify_ssl</code> (bool), <code>is_active</code> (bool)
<code>legal.case</code> (extend)	Link room to case	<code>matrix_room_id</code> (char), <code>matrix_room_alias</code> (char, optional), <code>matrix_enabled</code> (bool)
<code>matrix.log</code>	Delivery log	<code>date</code> , <code>case_id</code> , <code>room_id</code> , <code>event_type</code> (case_created/stage_changed/hearing_created/manual/dm), <code>payload</code> (text), <code>status</code> (ok/failed), <code>error_message</code>
<code>matrix.contact</code>	Synced contacts	<code>matrix_user_id</code> (@user:server), <code>display_name</code> , <code>avatar_url</code> , <code>is_direct</code> (bool), <code>last_seen</code> (datetime, optional)
<code>res.users</code> (extend,	Map to Matrix	<code>matrix_user_id</code> (char), <code>matrix_display_name</code> (char)

optional) user

For a generic connector not tied to legal, replace `legal.case` with a generic `matrix.thread.mixin` in future iterations.

---|---|---| | `matrix.backend` | Homeserver config | name, homeserver_url, personal_access_token (Secret), default_sender_display, verify_ssl (bool), is_active (bool) || `legal.case` (extend) | Link room to case | `matrix_room_id` (char), `matrix_room_alias` (char, optional), `matrix_enabled` (bool) || `matrix.log` | Delivery log | date, case_id, room_id, event_type (case_created/stage_changed/hearing_created/manual), payload (text), status (ok/failed), error_message || `res.users` (extend, optional) | Map to Matrix user | `matrix_user_id` (char), `matrix_display_name` (char) |

For a generic connector not tied to legal, replace `legal.case` with a generic `matrix.thread.mixin` in future iterations.

4) Core Features & Acceptance

4.1 Backend Configuration

- Menu: **Settings** → **Matrix** to create one active `matrix.backend`.
- Button **Test Connection**: GET `/_matrix/client/versions` and validate token with a simple whoami call. **Accept**: Test shows green check or error with reason.

4.2 Room Linking on Case

- Fields on Case: `matrix_enabled`, `matrix_room_id`, `matrix_room_alias`.
- Button **Create Room** (optional if homeserver allows): POST to create private room and store `room_id`.
- Button **Open in Matrix** (URL action if alias present) **Accept**: After saving Room ID, **Send Test Message** posts a "Hello from Odoo" to the room and logs it.

4.3 Outbound Events → Matrix

- Triggers (config flags per backend):
 - Case created → "Case {name} created for {client}".
 - Case stage changed → "Case {name} moved to {stage}".
 - Hearing created → "Hearing {name} on {date_start} at {location}".
 - Manual message from a composer on the case.
 - Direct message (DM) from contact list composer (see 4.5).
- Delivery: HTTP POST to `/_matrix/client/v3/rooms/{roomId}/send/m.room.message`.

- Retry once on failure; record in `matrix.log`. **Accept:** Performing each action posts a message that appears in Matrix; status recorded as OK.

4.4 Manual Message Composer

- On the Case form, a text field and **Send to Matrix** button.
- Posts plain text with `msgtype: m.text`. Newlines preserved. **Accept:** Sending from Odoo appears in the linked Matrix room.

4.5 Contacts & Direct Messages (New)

- **Sync contacts** via:
 - `/_matrix/client/v3/contacts` equivalent sources: use **User Directory Search** `/_matrix/client/v3/user_directory/search` for on-demand lookup, and/or
 - **Recents** by listing joined rooms and extracting members from DMs (rooms with `is_direct=true`) using `/_matrix/client/v3/joined_rooms` and `.../rooms/{roomId}/members`.
- Store minimal local cache in `matrix.contact` (`user_id`, `display_name`, `avatar_url`, `is_direct`). Provide **Sync Contacts** wizard.
- **Contact List** menu with search; open a contact to **Start DM**:
 - If an existing DM room exists, reuse it; otherwise **create room** with `is_direct=true` and `invite=[matrix_user_id]` via `/_matrix/client/v3/createRoom`.
 - Composer on the contact form to send a first message. **Accept:** Running Sync populates contacts; clicking *Start DM* either reuses or creates a DM room, then sends a message visible in Matrix and logs an entry.
- On the Case form, a text field and **Send to Matrix** button.
- Posts plain text with `msgtype: m.text`. Newlines preserved. **Accept:** Sending from Odoo appears in the linked Matrix room.

4.5 Inbound (Optional Stretch)

- Webhook endpoint (controller) to receive Matrix events (via AppService or webhook bot) and create a `mail.message` in case chatter for `m.room.message` events.
 - Minimal security: shared secret token in query/header; ignore non-text events. **Accept (Stretch):** A text message in Matrix appears in the Odoo case chatter within seconds and attributes to a generic Matrix Bot user.
-

5) UI/UX

- Menu: **Matrix**
 - **Backends** (form/list)
 - **Logs** (list with filters: status, case, event_type)
 - Case form: *Matrix* page with room fields, *Send Test*, *Send to Matrix*, and a small help text.
-

6) Security & Access

- Groups: `group_matrix_user`, `group_matrix_manager`.
 - Access rules: Managers full on backend/logs; Users read backend minimal fields (not token), create logs, write case fields they own.
 - Hide token in UI; store as `ir.config_parameter` or `fields.Binary` with encryption depending on Odoo 18 capabilities.
-

7) Technical Checklist

- `__manifest__.py`: depends `["base", "mail"]`. If legal app is used, add it to depends only in that module; otherwise keep connector generic.
 - Python:
 - `matrix.backend` model with test action; HTTP via `requests` (or `odoo.url_open`), timeouts, SSL options.
 - Mixin or case extension to send events.
 - Controller for optional inbound webhook.
 - Views: backend, logs, case additions, composer button, actions.
 - Data: security groups, `ir.model.access`, menus, server actions for triggers.
 - Tests: mock HTTP; test connection; send message; failure logs; (optional) inbound controller parsing.
 - Lint & OCA pre-commit; i18n.
-

8) Demo Data

- One demo backend pointing to a placeholder homeserver URL.
 - One demo case with a sample room ID.
 - A few log entries (OK/failed) for screenshots.
-

9) Stretch Goals

- Create room with invite list (lawyer emails mapped to Matrix IDs).
 - Attachments → upload via `m.room.message` with `url` (requires media API).
 - Per-stage templates for messages; jinja-like formatting.
 - Map Matrix user to Odoo user for inbound attribution.
 - Encryption-aware mode (document only, no implementation in MVP).
-

10) Acceptance Criteria (Summary)

- Backend connection test passes and errors are surfaced clearly.
- Case can store a Matrix Room ID and send a test message.
- Case lifecycle events post to the room and create delivery logs.
- Manual composer posts plain text to the linked room.
- (Stretch) Inbound webhook creates chatter messages in the linked case.

End of MVP Requirements