

Sign In

Enrollee / Experimenter / Admin No passwords: sign in with an identity provider or a one-time email code, and your role decides where you land.

Screens

Sign-in sign-in.htmlMy studies (enrollee) enrollee-landing.html

Console (experimenter) experiments-home.htmlConsole (admin) users-roles.html

Reads

the person profile

Key fields

the identity profile.id = auth.users.idwhat they may do roleswitched off? enabled

display handle usernameaccount email email

1 · Given / When / Then

Human wording first; the exact table/field in `mono`.

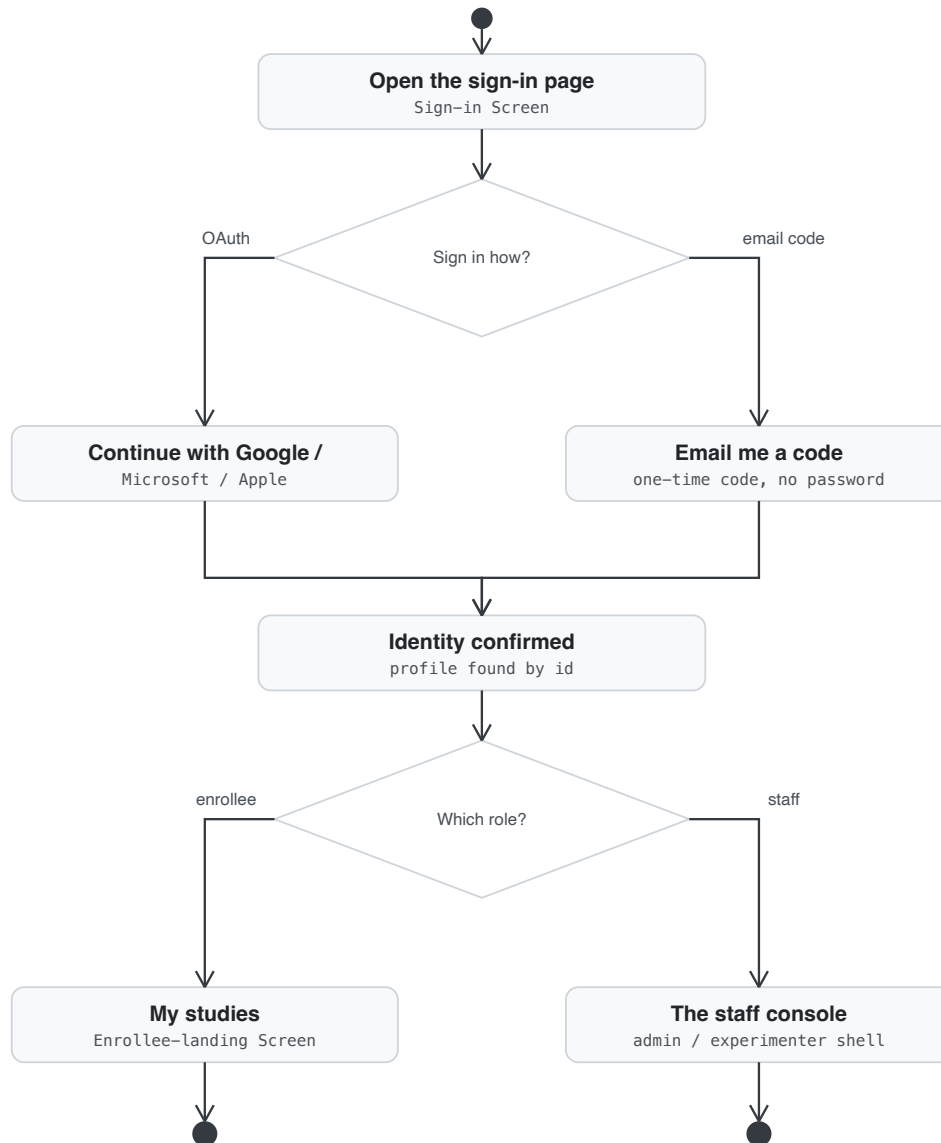
GIVEN	An admin has already invited the person, so a <code>profile</code> row exists with a <code>role</code> (admin , experimenter , or enrollee) and <code>enabled = true</code> . There are no passwords in ChatMaestro at all.
WHEN	On the Sign-in Screen they either continue with an identity provider (Google, Microsoft, or Apple) or ask for a one-time code by email. The login itself is handled by the managed login service, and ChatMaestro never sees a secret.
THEN	The login service returns a signed token whose user id is <code>profile.id</code> . The server loads that profile and routes by <code>role</code> . All three roles sign in through this same flow and differ only in where they land: a enrollee arrives on the My-studies Screen ; an experimenter arrives in the console with the Workspace tools; an admin arrives in that same console with the Admin group added. That same id is what row-level security, the database's per-user access rules, keys off for every later query. The session then lasts until the person signs out or an admin disables the account: there is no idle timeout, because the token proves identity only, and <code>role</code> , <code>enabled</code> and <code>opted_out_at</code> are re-read from <code>profile</code> on every query, so revocation is immediate rather than waiting for a token to expire (ALGORITHM §29). <code>reads</code> <code>profile</code> <code>writes</code> nothing (the session lives in the login service)

Variations and exceptions	
IF...	THEN...
The login succeeds but no <code>profile</code> matches the user id	No access: ChatMaestro is invite-only.
<code>profile.enabled = false</code>	Refused at the door even with a valid token. Deactivation is reversible and preserves all their data.
The one-time code has expired or was mistyped	The person requests another; codes are short-lived.
The person changes their email at the login provider	The verified change flow updates it and this copy follows; every join uses <code>id</code> , so nothing else moves.

Acceptance test — *Given* an enabled enrollee profile, *When* they sign in with a one-time email code, *Then* they arrive on the My-studies Screen with only their own runs visible; *and* a disabled profile is refused despite a valid token.

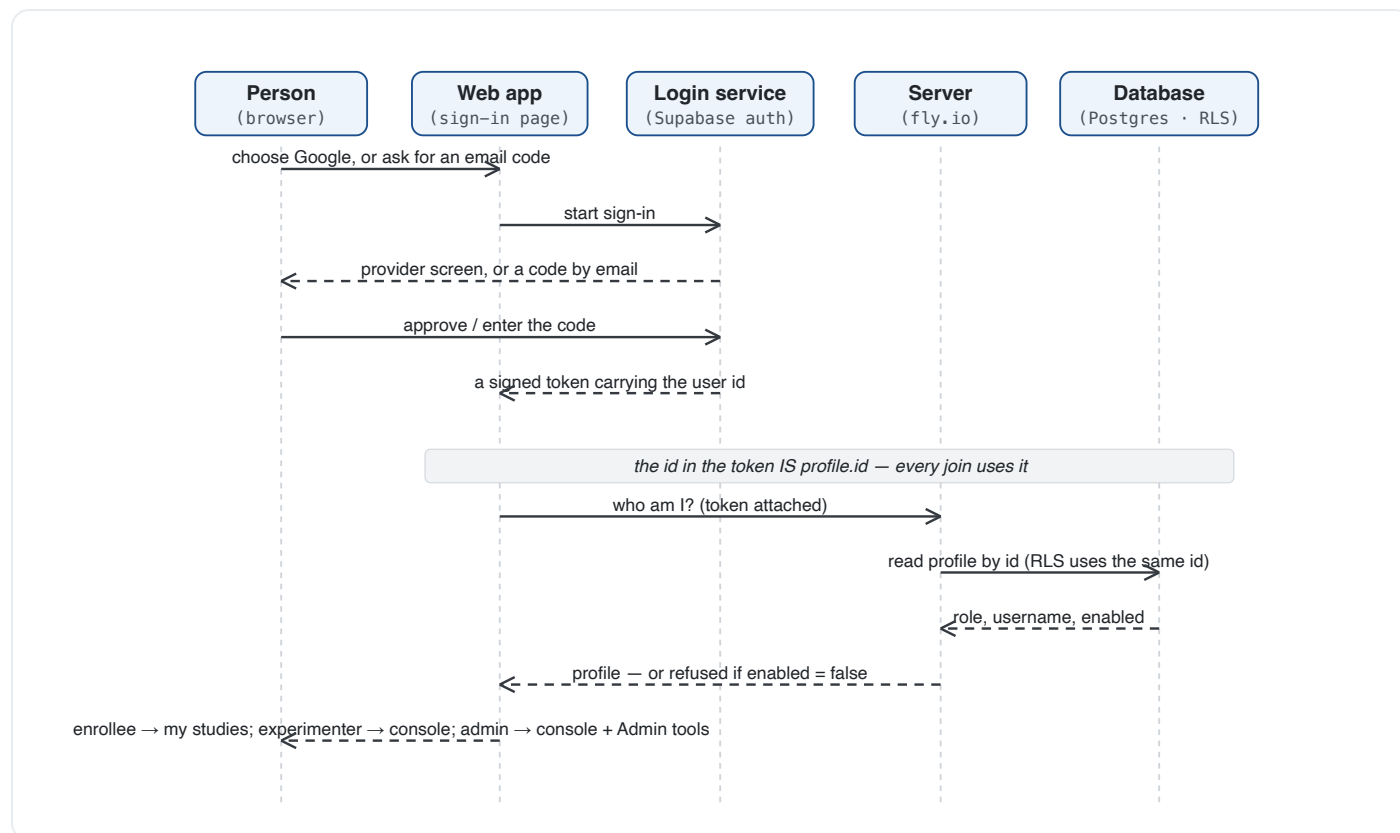
2 · The Journey, Screen by Screen

Two ways in, one identity, two destinations.



3 · Behind the Scenes

Who verifies what, and why the user id matters more than the email.



The login service is the verification authority; `profile` mirrors the email for display and keys everything off `id`.

4 · In Plain English

There are no passwords here. You either sign in with an account you already have, such as Google, Microsoft, or Apple, or you ask for a one-time code by email.

ChatMaestro is invite-only: signing in successfully is not enough, because somebody must have added you first. What you see afterward depends entirely on your role. Enrollees go straight to the studies they have been assigned. Experimenters land in the console with the everyday Workspace tools. Admins get that same console plus the Admin group (users, catalog, audit, database).

One deliberate design choice is worth knowing: everything internally hangs off a permanent user id, not your email address. That means you can change your email later and nothing breaks. And enrollees appear to each other under an assigned handle rather than a real name, so studies stay blind.

Fits the schema cleanly. `profile.id` = `auth.users.id`, and `role` + `enabled` are the only gates. No password column exists anywhere by design. `username` is a system-assigned handle (for example `enrollee-0001`) that keeps names off the screens staff read; it is a convenience, not anonymity — the handles are sequential and the same in every study.