

Access control policy

Version	1.0
Effective date	27 October 2026
Owner	Robin Ashford
Applies to	Every account that can reach Final Frame systems or customer data: staff, service identities and customer users
Review	27 October 2027
MPA CSBP	TS-1.2, TS-1.6, TS-1.7, TS-1.8, TS-6.0, TS-6.1, TS-6.2, TS-8.1, OR-3.1, OR-3.2

Approved by Robin Ashford, Founder, 27 October 2026

Principles

- Least privilege. An account gets what its task needs and nothing more.
- Default deny. No access exists until something grants it.
- Named accounts only. No shared logins.
- A second factor on every human account that can reach customer content or production configuration.
- Access is reviewed every quarter and the review is recorded.

Staff access to the company's systems

System	Rule
Google Cloud	MFA on the Google account. Owner stays with one person. Day-to-day work moves to a narrower role when a second person joins
Google Cloud organisation	Same
GitHub org	MFA on every member, enforced by the org 2FA requirement
WorkOS dashboard	MFA on. Changes to sign-in settings follow the Change management policy
Postmark, Resend, fal, TMDB, fanart.tv, EIDR, Squarespace, Apple Developer	MFA on wherever offered
Final Frame platform admin plane	Signed in with a passkey, or through WorkOS with a second factor, only
Password manager	Holds every credential and recovery code. MFA on

Service identities

Identity	Rule
(Cloud Run service and jobs)	No role on another company's project. Read-only roles for Integrations Health
Each Org's identity	Object rights on that Org's own buckets only. Created and deleted with the Org
Default compute and Cloud Build accounts	cloudbuild.builds.builder for image builds. Narrowed when CI deploys through Workload Identity Federation
(Postgres)	The tenancy suite checks it on every CI run

Identity	Rule
(Postgres)	Owns the schema. Used only by migration runs
Postgres superuser secret	Used by hand only, for bootstrap. Not mounted on any service

No service account key files are created. Services use the metadata server and IAM signBlob.

Customer users of the platform

- **Invitation only.** An Org admin adds a person. That writes a membership and sends an invitation.
- **A membership is the only grant.** It names a user, an Org, a scope (the Org, a Group or a Project) and a role. Access inherits down the Group tree.
- **Roles are permission bundles:** Admin (all permissions), Editor (fills projects in, no user management, handoff or ending), Viewer (reads and uploads). Stored as admin, user and guest. Code checks permissions, never role names.
- **Server side, every request.** requireOrg verifies a membership before an Org id is used. `authorise(permission)` decides. A route with no policy fails closed. reads the whole route table: every route answers 401 signed out, 404 to a member of another Org, and reaches a permission decision, unless it is listed with a reason.
- **Database enforcement.** Postgres RLS on every Org-scoped table, forced, default deny.
- **Leaving.** Removing a membership ends that person's access to the Org on their next request. Password reset revokes every session.

The Screening Room

The Screening Room shows an Org's own catalogue to its members and to people at other companies. It adds viewer roles and a second kind of principal, the screening guest.

- **Permissions.** `screening.view`, `screening.curate`, `screening.manage` and `screening.guests.manage`. Admin holds all four through its bundle. Editor and Viewer hold none. A Group-scoped Admin holds none from the role: the room is the whole Org's. Routes call `requireScreening`: a membership first, 404 without one, then the permission through `authorise()`.
- **Viewer roles.** Viewer (view), Curator (view and curate) and Manager (all four), given in the CMS to one member or to a viewer group. A viewer role adds to a membership and never stands in for one. With no membership in the Org a grant gives nothing, so a person whose membership ends loses the room with it.
- **Grants.** One grant per audience: the Org, a Group of the Org tree, one member, a viewer group or one guest. A grant names what it shows (the whole catalogue, chosen collections or chosen titles), what it hides, its markets, its languages and its watermark. A share to a Group reaches a membership at that Group or below it, by the same materialised path test as membership inheritance. Collections are shared with the Org, its Groups, members and viewer groups from the collection's Who sees it.
- **What a viewer sees** is decided on every request from the grants that reach them: a hide on any grant wins, a title must be live inside its window and its collection's, internal titles and collections never reach a guest, and markets and languages narrow anything tied to a market. A title outside it is a 404. Curators and Managers see the whole live room.
- **Viewer groups** are named lists of members and guests. They are not Groups of the Org tree. A viewer group's role reaches its members; its guests only view.

- **Screening guests.** A guest is an email address one Org invited into its Screening Room. No users row, credential, invitation or membership is written for a guest, so the app, /v1, MCP and the Org switcher never open for one. A guest holds screening.view alone, and only while invited or active, before their expiry, and with any NDA owed accepted in its current version.
 - **Admitted.** Someone holding screening.guests.manage invites the address, with a company, an expiry (the Org's default is 30 days; two years at most), the NDA, viewer groups and the guest's own grant. An address that is a member of the inviting Org is refused and given access as an internal user instead; the check runs in that Org only. The invitation email names the viewer's address and carries no token.
 - **Signs in** on the viewer host only. The guest asks for a link there. Every request gets the same answer whether the address has access or not, at most 5 an hour per address and 20 per IP. The link's token is stored as its SHA-256 hash, lasts 15 minutes, is used once by a press on the page it opens, and completes only in the browser that asked (a binding cookie). A factor is enrolled only while the guest holds none. An Org may lower guests to the link alone.
 - **Lockout.** Five wrong factors lock the guest for 15 minutes; ten hold them until a Manager unlocks them. Each go is counted under a lock on the guest's row.
 - **Scoped** to the guest's own grant and their viewer groups' grants, in one Org. The guests' field policy is applied by the server before any answer.
 - **Removed.** Revocation ends every session and stream of the guest in the same transaction. Expiry ends a guest's sessions at their next request. Saving a new NDA text ends the sessions of every guest asked for it, closing the viewer ends every guest session, and raising guests from the link alone to MFA ends the sessions the link alone made. A Manager's "Reset their second step" removes the factors and ends the sessions. A guest row is revoked, never deleted.
- **The platform plane.** requireScreening gives a platform admin all four permissions in any Org's room, logged at warn every time. A platform admin cannot sign in as a guest without the guest's factor.

Authentication

- **Two doors for people, on Final Frame's own sign-in card.** WorkOS: email and password followed by an authenticator-app code, or the person's company single sign-on (SAML or OIDC). A passkey: WebAuthn with user verification required, so the device and its unlock are both factors, bound to the app's own host and added in Preferences after a signed-in session. Passkeys are the recommended method. Both doors end in the same server session.
- **MFA is set to Required in WorkOS.** Every password sign-in asks for a second factor.
- **No other door on a deployed environment.**
- **Social sign-in** (Google, Microsoft, Apple) relies on the provider's own second factor. It is never the platform admin's door.
- **Password rules are held by WorkOS:** at least 16 characters, breached-password check on.
- **Rate limits** on every sign-in route, in the API and at the load balancer (Cloud Armor).
- **Screening guests** use the viewer host's own door, set out in section 3.
- **Machines and agents** use API keys minted and validated at WorkOS, or an AuthKit OAuth token for an MCP client. Each answers for one Org, never on the platform plane, with the permissions its row lists or its person holds at that moment.

Sessions

- Opaque 256-bit tokens. Only a SHA-256 hash is stored.
- Server-enforced idle timeout: 15 minutes, per CSBP TS-8.1. The web app asks "Still there?" 60 seconds before the idle sign-out.
- Absolute lifetime: 12 hours. Impersonation sessions: 30 minutes.
- Cookie `__Host-session` on deployed hosts: `HttpOnly`, `Secure`, `Path=/`, no Domain, `SameSite=Lax`. `Lax` rather than `Strict`: the WorkOS callback must receive its `__ff_oauth` state cookie on a cross-site top-level redirect, and transfer-grant file links are cookie-authenticated top-level GETs. `__session` on local http only. A leftover `__session` on a deployed host is cleared.
- **Screening guests' viewer sessions** live on the viewer host only. Token hash only. Cookie `__Host-sv-
<orgId>` per Org: `HttpOnly`, `Secure`, `Path=/`, no Domain, `SameSite=Strict`. The step between a used link and the second factor lasts 10 minutes and reaches only the factor routes. A session lasts (8 by default, 1 to 24) and ends after idle (60, 5 to 480). Every request checks the session, the guest and the NDA again; a session that fails ends there with its reason (signed out, revoked, expired, idle, capped, NDA changed, locked, reset, viewer closed, replaced). Writes must carry `X-FF-Viewer: 1`, and a call the browser marks cross-site or same-site is refused.

Privileged access to customer data

- Platform admin is Robin only.
- "View as" (impersonation) is used only to answer that Org's support request or to investigate an incident. Each use writes `impersonation.started` and `impersonation.ended` to the Org's audit trail with the real person behind it.
- No cross-Org access to media exists. There is no admin tool that reads another Org's bucket.
- Direct database access to production is for incidents and migrations only, and is noted in the incident or change record.

Joiners, movers, leavers (staff)

- **Join:** NDA signed, screening done, the security awareness module completed, then accounts created with the least role.
- **Move:** access changed the same day the job changes.
- **Leave:** every account disabled the same day. Secrets the person could read are rotated within five working days.

Quarterly access review

Every quarter Robin checks, and records the date and result in the risk register:

- Google Cloud project and organisation IAM bindings
- GitHub org members and repo access
- Every vendor dashboard's users and API keys
- Secret Manager: which identities can read which secrets
- Platform admin flags in the database
- Launch tokens minted in the quarter (impersonation)

- Per Org with a Screening Room: its Curators and Managers, its viewer groups' members, and its live guests with their expiry

Final Frame Ltd · Registered in England and Wales · Company number 17413345 · www.final-frame.com/trust