Frontend Pages & Route Map

- `/login` — Supabase Auth UI (email/pass or SSO)
- `/mfa` — MFA challenge/verify (if enabled)
- `/org/settings` — Org profile + policy readouts
- `/org/teams` — List + create/edit/delete teams; manage team members
- `/admin/users` — Invite org users, edit role/clearance, disable/remove
- `/projects` — Project list (filters: team, owner, deadline)
- `/projects/[projectId]`
  - `overview` — Title, description, budget, deadline, members
  - `members` — Add/remove members
  - `files` — List/upload/download/rename/delete/change clearance
  - `audit` — Project audit stream
- `/p2p` — Inbox/Outbox (view-once messages); Compose modal
- `/reports`
  - `reassignments` — Clearance changes stream + filters
  - `projects/[projectId]/summary` — Per-project summary widgets

4) Endpoint Contracts (what FE sends/expects)

Organizations & Teams

GET `/organizations/me`

- **Headers:** auth, `x-org-id`
- **Response:**

```
{ org: Org; membership: Membership; }
```

POST `/org/teams`

- **Body:** `{ name: string; description?: string }`
- **Response:** `{ teamId: string; name: string; }`

PATCH `/org/teams/:id`

- **Body:** `{ name?: string; description?: string }`

- **Response:** `Team`

DELETE `/org/teams/:id`

- **Response:** `{ removed: true }`

POST `/org/teams/:id/members`

- **Body:** `{ userId: string; role_on_team?: string }`
- **Response:** `{ added: true }`

DELETE `/org/teams/:id/members/:userId`

- **Response:** `{ removed: true }`

Admin / Users

POST `/admin/users`

- **Body:** `{ email: string; name?: string; role: Role; clearance: ClearanceLevel; teamIds?: string[] }`
- **Response:** `{ userId: string; orgId: string; role: Role; clearance: ClearanceLevel; }`

PATCH `/admin/users/:id`

- **Body:** `{ role?: Role; clearance?: ClearanceLevel; teamChanges?: { add?: string[]; remove?: string[] } }`
- **Response:** `OrgUser`

DELETE `/admin/users/:id`

- **Response:** `{ removed: true }`

Projects (with metadata)

POST `/projects`

- **Body:**

```
{
  teamId: string;
```

```
  title: string;
  description?: string;
  budget_amount?: number;
  budget_currency?: string;
  deadline?: string;           // ISO date
  owner_id?: string;
}
```

- **Response:** `{ projectId: string; }`

PATCH `/projects/:id`

- **Body:** any of the fields above (partial)
- **Response:** `Project`

DELETE `/projects/:id`

- **Response:** `{ removed: true }`

GET `/projects`

- **Query:** `?teamId=&ownerId=&q=&deadlineStart=&deadlineEnd=&cursor=&limit=`
- **Response:** `{ items: Project[]; nextCursor?: string }`

POST `/projects/:id/members`

- **Body:** `{ userId: string; role_on_project?: string }`
- **Response:** `{ added: true }`

DELETE `/projects/:id/members/:userId`

- **Response:** `{ removed: true }`

Files (per-file clearance)

GET `/projects/:id/files`

- **Query:** `?q=&mime=&cursor=&limit=`
- **Response:** `{ items: FileItem[]; nextCursor?: string }`

POST `/files/upload-intent`

- **Body:**

```
{
  projectId: string;
  filename: string;
  mimetype: string;
  size: number;
  clearance_level: ClearanceLevel;
}
```

- **Response:**

```
{ fileId: string; uploadUrl: string; storageKey: string; }
```

FE flow:

1. Call `upload-intent` → get `uploadUrl`
2. `PUT` the raw file bytes directly to `uploadUrl` (no auth header)
3. On success, call `/files/complete` with `{ fileId, checksum }`

POST `/files/complete`

- **Body:** `{ fileId: string; checksum?: string }`
- **Response:** `{ status: "SCANNING" }`

GET `/files/:id/download-intent`

- **Response:** `{ downloadUrl: string; expiresAt: string }`

FE: perform a `GET` on `downloadUrl` (no auth header); browser downloads.

PATCH `/files/:id`

- **Body:** `{ filename?: string }`
- **Response:** `FileItem`

DELETE `/files/:id`

- **Response:** `{ removed: true }`

PATCH `/files/:id/clearance`

- **Body:** `{ new_clearance: ClearanceLevel }`
- **Response:** `FileItem` (with updated `clearance`)

Access Overrides (time-bound allow/deny)

POST `/access/overrides`

- **Body:**

```
{
  resource_type: "project" | "file";
  resource_id: string;
  subject_user_id: string;
  effect: "ALLOW" | "DENY";
  starts_at?: string;   // ISO
  ends_at?: string;     // ISO
}
```

- **Response:** `AccessOverride`

DELETE `/access/overrides/:id`

- **Response:** `{ removed: true }`

P2P (view-once secure messages)

POST `/p2p`

- **Body:** `{ recipient_id: string; text?: string; attachment_file_id?: string; expires_at?: string }`
- **Response:** `{ p2pId: string; state: "PENDING" }`

GET `/p2p/:id`

- **Response:**

```
{
  id: string;
  text?: string | null;
  attachment?: { downloadUrl: string; expiresAt: string } | null;
}
```

> **Note:** After this call, the message becomes DELIVERED and will be auto-deleted per policy.

DELETE `/p2p/:id`

- **Response:** `{ removed: true }` (only if still `PENDING`)

Audit & Reports

GET `/audit`

- **Query:** `?projectId=&fileId=&userId=&type=&since=&until=&cursor=&limit=`
- **Response:** `{ items: AuditEvent[]; nextCursor?: string }`

GET `/reports/reassignments`

- **Query:** `?since=&cursor=&limit=`
- **Response:** `{ items: AuditEvent[]; nextCursor?: string }` (where `eventType === "FILE_CLEARANCE_CHANGED"`)

GET `/reports/projects/:id/summary`

- **Response:**

```
{
  countsByClearance: Record<ClearanceLevel, number>;
  recentEvents: AuditEvent[];
}
```