

ABUNDANT LIFE CHURCH OF DELIVERANCE · IT MINISTRY

Notifications

A Study of How the ALCOD Dashboard Talks to Its People

White Paper · Version 1.1

Effective 2026-07-10

alcod-dashboard.pages.dev

FOR STAFF WHO WORK THE QUEUES — PASTORAL · EXECUTIVE · SOCIAL CONNECT · IT · FINANCE

1 The problem this system solves

Church operations depend on **turnaround time**: a member submits an announcement, and someone has to notice, review, decide, and communicate back. Before this system existed, the failure mode was always the same:

"I didn't know a request was waiting for me — nobody told me."

That gap happens in three places:

1. **Intake gap** — the form is submitted, the SharePoint row is created, but no staff member sees it because they weren't looking at the dashboard.
2. **In-flight gap** — a request moves through statuses, but the submitter has no idea until someone remembers to email them.
3. **Awareness gap** — a staff member logs in but has to click every tab to see which queue has work waiting.

We close all three with a layered notification system. **No one has to remember to notify anyone.** The system does it.

2 The mental model — two layers, two audiences

Every notification in the dashboard is either **Layer 1** or **Layer 2**. The difference is *where the code lives*, and by extension, *who can change it*.

	Layer 1	Layer 2
Where it lives	Power Automate (cloud)	React app (this repo)
Who can edit it	Tenant admin in a browser	Developer with a code push
When it fires	The moment a SharePoint row is <i>created</i>	When staff <i>transition</i> a status inside the dashboard
Primary audience	Staff (the queue owner)	Submitter (the church member) — plus Finance for reimbursements
Sender identity	itministry@ (Power Automate connection owner)	The signed-in staff member's mailbox
Failure mode	Turn the flow off in Power Automate — dashboard keeps working	Redeploy the app

The **two audiences are the point**. Layer 1 says "staff, wake up." Layer 2 says "member, here's your update." They don't overlap; they don't fight each other; they don't send duplicate emails.

3 Layer 1 in detail — staff intake alerts

A church member submits a form. A Power Automate flow watches the SharePoint list. The instant a new row appears, the flow sends an email to the person whose queue it belongs to.

THE THREE FLOWS

Form submitted	SharePoint list	Email goes to
Announcement / Event	AnnouncementRequests	executiveadmin@ (or socialconnect@ on bypass)
IT Support	ITSupportRequests	itministry@
Purchase Request	PurchaseRequests	pastor@ + copastor@

THE ANNOUNCEMENT BYPASS

Announcements *only* (not events) skip Exec Sec and go straight to Social Connect. This mirrors the pre-dashboard workflow — announcements are a stream, not a decision. The Power Automate flow reads the `Status` column that was already set at intake and routes the intake email accordingly.

WHY LAYER 1 IS POWER AUTOMATE AND NOT REACT

The React app runs **in the church member's browser**, and church members are not signed in when they submit a form. So the app has no identity, no permission, and no way to send an email on their behalf. Power Automate runs in the cloud with a stored credential (`itministry@`) and doesn't need a user session.

PRACTICAL CONSEQUENCE

You can change intake behavior in 5 minutes without a code deploy. Runbook is at `docs/notifications-layer-1-power-automate.md`.

4 Layer 2 in detail — status transition emails

Once the request is in a queue, staff act on it in the dashboard. Every meaningful click either **moves the request** to a new status or **adds information** (notes). Both trigger emails.

THE FULL TRIGGER TABLE

Module	Trigger action	Recipient	Purpose
Announcements	Exec Sec approves	Submitter	"Your request is moving to Social Connect."
Announcements	Exec Sec approves NEW	Social Connect team	"New request forwarded to your queue." Closes the previous silent-hand-off gap.
Announcements	Exec Sec denies	Submitter	Denial reason.
Announcements	Social Connect approves	Submitter	"Approved!" — item also lands on the 17 Schedule tab.
Announcements	Social Connect denies	Submitter	Denial reason.
Announcements	Marked completed	Submitter	Confirmation the announcement ran.
IT	Admin sets → In Progress NEW	Submitter	"We're on it."
IT	Admin sets → Waiting on User	Submitter	Specific info request.
IT	Admin sets → Resolved	Submitter	Fix explanation.
IT	Admin sets → Cancelled	Submitter	Reason for closing.
Purchase	Pastor approves	Submitter	"Approved, here's next step" (differs for reimburse vs church-buys).
Purchase	Pastor denies	Submitter	Reason.
Purchase	Marked Awaiting Reimbursement	Submitter	"Receipt received, reimbursement queued."
Purchase	Marked Awaiting Reimbursement NEW	Finance	Separate email w/ \$ + receipt link + submitter contact.
Purchase	Marked Completed	Submitter	Confirmation.

WHY THE SENDER IDENTITY MATTERS

Layer 2 emails go from **the signed-in staff member's mailbox**, not from a shared account. When Pastor Fox approves a purchase, the submitter sees an email from `pastor@abundantlifecod.org`. Replies come back to Pastor Fox directly.

This is deliberate for two reasons:

1. **Accountability.** A member can see who made the decision.
2. **Reply threading.** The submitter's reply lands in the decision-maker's inbox, not a shared queue where it might get lost. The dashboard *also* has an in-modal Email Thread for shared visibility across `requests@` — that's a separate mechanism.

THE FINANCE-SIDE REIMBURSEMENT EMAIL

The one exception to "Layer 2 = submitter" is the Finance reimbursement email. When a purchase moves to *Awaiting Reimbursement*, the submitter gets their "receipt received" email **and** Finance gets a *separate* email with the \$ amount, receipt link, submitter contact, and approver name. Reasoning: Finance can't act on the reimbursement without those details, and reading the submitter's email is a delegation loop.

THE PURCHASE VIEWED-BY INDICATOR (CONCURRENCY PROTECTION) NEW

Pastor and CoPastor both get the intake email. Both can approve. To prevent them from racing each other on the same decision, opening a card writes the viewer's name and timestamp to two SharePoint columns (`LastViewedBy` , `LastViewedAt`). If the other Pastor opens the same card **within 30 minutes**, they see:

- A gold "👁️ Pastor Fox · 5 min ago" pill on the lane card
- A yellow banner at the top of the modal warning them to check with the other approver first

Silent if it's you (no self-shaming) or if the last view is stale (>30 min). This is the only Layer 2 mechanism that writes to SharePoint on a *read* action rather than a status change.

5 Layer 2 in detail — in-app pending badges

An email is a *push*: it interrupts. A badge is a *pull*: it's visible only when the staff member opens the dashboard. Both matter, because email drowns and dashboards get ignored.

WHAT THE BADGE SHOWS

Every tab label carries a **red pending count** for that queue:

- Exec Sec Inbox — how many *Pending Exec Sec*
- Social Connect Queue — how many *Pending Social Connect*
- IT Support — how many *New*
- Purchase Requests — how many *New*
- Finance Reports — how many *Awaiting Reimbursement*

For admins (who see all tabs via a dropdown), the count appears **inside** the dropdown option, and a pulsing gold "*N pending*" pill sits next to the dropdown so you know without opening it whether anything needs your attention.

HOW IT STAYS FRESH

The `useCounts` hook queries the same SharePoint lists the tabs use, filters to just the "pending" statuses, and refreshes every 60 seconds. Hitting ↻ Refresh in the header also triggers an immediate count refresh.

EFFICIENT BY DESIGN

The hook only fetches data sources the current user's role can see. A pastoral-role user (no IT tab) never triggers an IT ticket fetch. A finance-role user only fetches purchase data. This keeps Graph API load minimal.

DEMO-MODE AWARE

Demo mode short-circuits both the count fetch and Layer 2 emails. Leadership viewing a demo sees the badges as zero and nothing hits real inboxes.

THE SCHEDULE TAB — PLANNING SURFACE FOR APPROVED ITEMS NEW

Once Social Connect approves an announcement or event, it also lands on the  **Schedule** tab (Social Connect + Admin). This is deliberately downstream of every notification: the calendar answers "what's on the books?" not "what needs my attention?" Three parts:

1. **Three summary tiles** — Approved on calendar · Events booked · Announcement plays
2. **Roster** — next 4 weeks of Sundays and Wednesdays, cards listing what's airing
3. **Strategic Calendar** — FullCalendar month view (burgundy events, gold announcement plays) for planning months out

Only *Approved* and *Completed* statuses appear. Pending items live in queue tabs, not the calendar.

6 Full lifecycle — one story, one member

Let's walk through what a church member and her staff counterparts actually experience.

6.1 Sister Adams submits an event request

Time 0:00 — Sister Adams fills out the MS Form for the Women's Ministry retreat. She hits Submit.

Time 0:05 — The Power Automate `ALCOD - Form to SharePoint` flow runs. It:

1. Creates a row in `AnnouncementRequests` with `Status = Pending Exec Sec`.
2. Sends Sister Adams a confirmation email: *"We got your request."*
3. **Layer 1** Sends `executiveadmin@` an email: *"New request in your Exec Sec inbox — Women's Retreat."*

Time 0:07 — Executive Admin's phone lights up with the email. She's at lunch but sees it. She'll get to it after.

Time 1:30 (later that afternoon) — Executive Admin signs into the dashboard. Even before she navigates, the header shows *"Exec Sec Inbox (3)"* — three items waiting. She knows without looking that there's work.

Time 1:31 — She clicks Exec Sec Inbox. The badge drops off the label. She reviews Sister Adams's request, clicks Approve.

Time 1:31 — **Layer 2** Three notifications fire:

- Sister Adams gets: *"Your request is progressing — Social Connect has it now."* (from `executiveadmin@`)
- **Social Connect team gets a hand-off email** at `socialconnect@`: *"Forwarded to Social Connect: Women's Retreat."* **NEW**
- The dashboard's badge on Social Connect Queue increments.

Time 1:31 (Layer 1 also fires?) — No. The item wasn't *created*, it was *updated*. Layer 1 flows only run on create. Hand-off notifications from staff-to-staff are Layer 2.

Time 5:00 (next day) — Social Connect Coordinator sees the email + badge, opens the tab, approves the request. **Layer 2** Sister Adams gets: *"Approved!"* (from `socialconnect@`). The item **immediately appears on the  Schedule tab** — this is when the clock officially starts.

Time 8:00 (event runs) — Social Connect marks Completed. **Layer 2** Sister Adams gets: *"Completed. Thank you for serving with us."*

TOTAL MESSAGES

4 emails to Sister Adams (intake + moving + approved + completed). **2 emails to staff** about this request (Layer 1 intake + Layer 2 hand-off). No spam. No missed steps. No manual copy-paste.

7 What Layer 2 does NOT do (and why)

Not every state change is worth an email. We deliberately left these silent:

Non-notifying event	Why
Staff types a note but doesn't move the status	Notes are drafts; they auto-save so the staff doesn't lose work, but the submitter doesn't need a "note edited" ping.
Admin adds/removes a user in User Access	It's an admin task, no cross-party impact worth an email. The reminder banner covers the M365 group side.
Hidden-from-my-view toggle	Personal filter, not workflow.
Refresh / countdown expires	Fully internal to the app.
Reopen (from Completed back to New/In Progress)	Rare, deliberate — staff acknowledge it in-person.

RULE OF THUMB

If a real human on the other end can't act on a notification, don't send it. Every silent email trains the audience to ignore *all* your emails.

8 What breaks if something fails

Layer 1 (Power Automate) failures

- **Symptom:** New requests appear in SharePoint / the dashboard, but no staff intake email.
- **Cause:** flow disabled, or a Power Automate connector reauthorization expired.
- **Fix:** Log in to `make.powerautomate.com` as `itministry@`, check the flow's run history for red X's. Common fix: click the flow, re-authorize the SharePoint or Office 365 Outlook connection.
- **Fallback:** In-app badges still work. Staff still see counts when they open the dashboard. They just lose the push notification until the flow is restored.

Layer 2 (React app) failures

- **Symptom:** Status changes work, but the submitter doesn't get an email.
- **Cause 1:** The signed-in staff's Mail.Send permission was revoked.
- **Cause 2:** Graph token expired mid-session (rare — MSAL refreshes silently).
- **Cause 3:** The submitter's email field on the SharePoint row is blank.
- **Diagnostic:** Open browser DevTools → Console. Every failed notify logs `[notifySubmitter] failed for X: <message>`.
- **Fallback:** Manually forward from the Email Thread section in the modal.

Badge failures

- **Symptom:** Badges show stale or zero counts.
- **Cause:** Graph list-read failed. The hook logs to console but never blocks the app.
- **Fallback:** Open the tab; the tab's own hook loads the data directly.

9 Adding a new notification (quick reference)

Scenario A — new *intake* alert (someone submits a new type of form and someone should be pinged):

1. Add the SharePoint list (if new).
2. Create a Power Automate *When an item is created* flow with a Send email step.
3. Follow the pattern in `docs/notifications-layer-1-power-automate.md`.
4. No code change needed.

Scenario B — new *status transition* email (someone changes state in the dashboard):

1. Add a template to the relevant `src/lib/notifications*.js`.
2. Call `notifyXxx(item, 'your_new_type', notes)` in the transition function in the tab's view file.
3. Test locally, then push. Cloudflare auto-deploys.

Scenario C — new *badge* on a tab:

1. In `src/hooks/useCounts.js`, add the view to `neededSources()` and the count-filter to the reducer.
2. `Layout.jsx` renders it automatically because it looks up by view name.

The one-page cheat sheet

Question	Answer
A form was just submitted. Who knows?	The queue owner (Layer 1 email) AND the submitter (PA-side intake email).
Staff moved a request in the dashboard. Who knows?	The submitter (Layer 2 email). Plus Finance if it's a reimbursement.
How does staff know work is waiting without opening every tab?	Red badge on the tab label + pulsing "N pending" pill in the admin dropdown.
How do I add a new user?	Dashboard → User Access tab → + Add user. Also add them to the ALCOD Requests M365 group (yellow reminder banner points to Admin Center).
Where does the "we got your request" email come from?	Power Automate, sent from itministry@ as part of the intake flow.
Where do status update emails come from?	The signed-in staff member's mailbox. Recipient sees who acted.
Can I test all this without spamming real people?	Yes. Enter Demo Mode — sendMail is stubbed and badges are frozen at zero.
What if a Power Automate flow breaks?	Badges still work — staff see counts when they log in. Fix the flow, no data lost.
How do I make the system quieter for a specific transition?	Delete the notifyXxx(...) call in the tab's transition function. Redeploy.

A Appendix — file map

For engineers or future admins who want to trace the code:

Concern	Files
Layer 1 flows (external)	Power Automate portal, not in the repo
Layer 1 runbook	<code>docs/notifications-layer-1-power-automate.md</code>
Layer 2 templates	<code>src/lib/notifications.js</code> , <code>src/lib/notificationsIT.js</code> , <code>src/lib/notificationsPurchase.js</code>
Layer 2 call sites	<code>ExecSecView.jsx</code> , <code>SocialConnectView.jsx</code> , <code>BoardView.jsx</code> , <code>ITSupportView.jsx</code> , <code>PurchaseRequestsView.jsx</code>
Badge fetch	<code>src/hooks/useCounts.js</code>
Badge UI	<code>src/components/layout/Layout.jsx</code>
Sender wrapper	<code>src/lib/graph.js</code> → <code>sendMail()</code>
Role-aware view routing	<code>src/lib/config.js</code> → <code>VIEWS_BY_ROLE</code>