LTAP TRAVEL ASSISTANCE MODULES

Integrated Administrative and Claimant Documentation

Version 3.0 - Iteration Update

Field	Value
Primary modules	TA_Admin.svelte, TA_Claimant.svelte, Comp_AuthLogin.svelte, Comp_Contacts.svelte, Svc_Transmittal.ts
Platform	LTAP - Legal and Travel Assistance Platform
Revision objective	Incorporate OTP-only staff administration, TurboSMTP notification delivery, contact verification, claimant UX refinements, ledger containment, and migration boundary clarifications.
Source documents reviewed	TA_Admin Module Documentation v2.0; TA Claimant Module Documentation v1.0
Diagram support	Six generated system diagrams and UI wireframes are included. These are conceptual diagrams, not live browser screenshots.

Document purpose
This document consolidates the existing Admin and Claimant documentation into a single updated operating specification. It preserves the controls already documented in the uploaded PDFs while adding the newly delivered iteration: OTP-only admins from the SQLite admins table, real TurboSMTP delivery through Svc_Transmittal, verified contact email updates, improved claimant UX, and safer migration boundaries.

Contents

1. Executive Summary

The existing TA_Admin documentation already established a mature administrative control model: archive/current/overall lenses, hearing-date freshness, document source governance, workflow authorization, and ta_claim_history audit management. The claimant documentation already established the claimant orchestration model: activeUserStore context, history endpoint, phase machine, builder=true test mode, file/camera/test intake, AI validation, and ledger finalization.

This iteration adds one important architectural correction: staff/admin identity is no longer supposed to be hardcoded in the root server. The admins table in ltap.sqlite is now the authoritative staff source for Legal, Finance, and SuperUser users. Staff authenticate by OTP only; password_hash is nullable by policy and should not be used for named staff.

The iteration also hardens email behavior. AuthLogin OTP and contact-email verification must call the same notification service used by operational transmittals. The app must not claim that an email was sent when the server only logged a placeholder OTP. In live mode, Svc_Transmittal must route email through TurboSMTP and store notification evidence in notification_outbox.

- Admin staff source moves from hardcoded QA_STAFF_LOGINS to active rows in admins.
- Staff OTP lookup searches admins.email_address, admins.username, and admins.sms_number before member lookup.
- Contact email assignment verifies ownership of the new email, then updates members.email_address.
- TA_Claimant places the certification checkbox and Enter New Claim action in one intuitive block.
- Builder becomes a real toggle that adds or removes ?builder=true.
- Ledger Snapshot and Claim History are made collapsible and internally scrollable, preventing the history list from pulling the whole page.
- Members migration and TA claims migration remain separate tools with separate risk boundaries.

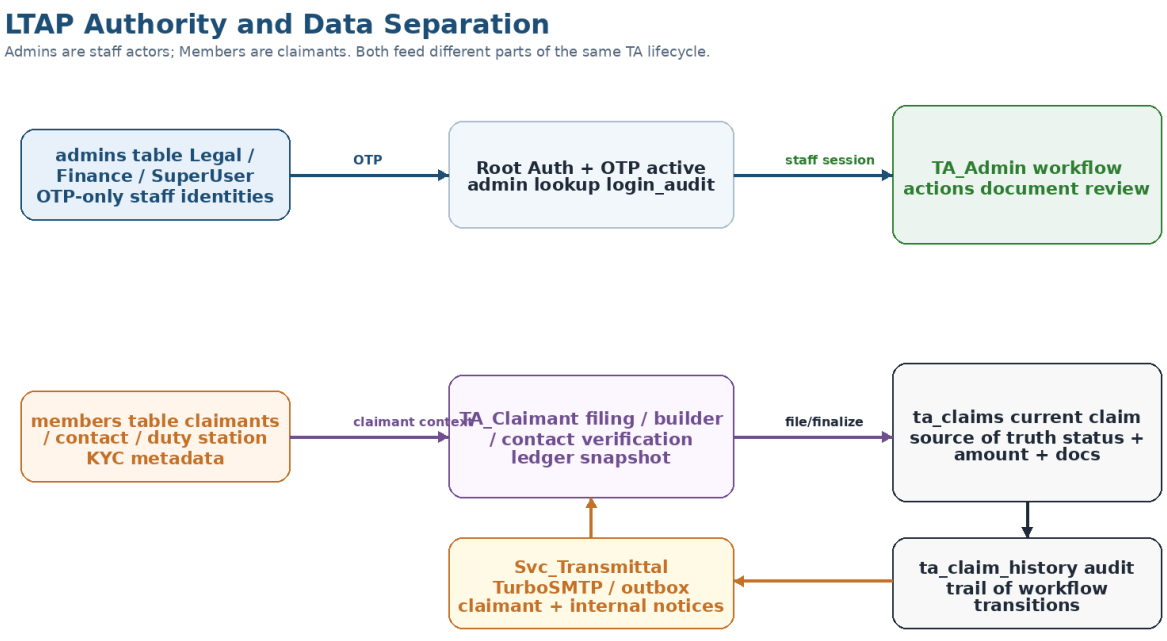

Figure 1. Updated LTAP authority and data-separation model.

2. Updated System Boundary and Source of Truth

The primary correction in this iteration is the separation of claimant identity from staff/admin identity. The members table is the claimant/member registry. The admins table is the staff/admin registry. These two tables should not be treated interchangeably.

Object	Authoritative Responsibility	Used By	Important Rule
members	Claimant/member identity, contact details, duty station, KYC metadata, and claimant profile state.	TA_Claimant, claimant login, TA_Admin claimant joins, migration/members.	Members are claimants, not Legal/Finance/SuperUser staff.
admins	Staff/admin identity for Legal, Finance, SuperUser, and future staff roles.	AuthLogin OTP, TA_Admin role context, admin_session.	Staff are OTP-only; named staff should not depend on password_hash.
ta_claims	Current claim source of truth.	TA_Claimant finalization, TA_Admin dashboard/workflow.	Workflow status belongs here; archive marker remains document_hash = LEGACY_TSV_IMPORT.
ta_claim_history	Workflow audit ledger.	TA_Admin status actions, migration/ta_claims legacy history.	Use ta_claim_history for TA. Avoid generic Claim_History in new code.
notification_outbox	Notification evidence and delivery attempts.	Svc_Transmittal, OTP/email verification, TA status notices.	Every live notification should leave observable evidence.

Design decision
The frontend may continue to receive qaStaffCredentials for compatibility, but the source should be loadActiveAdmins(db), not a hardcoded constant. This preserves the UI contract while removing hardcoded staff identity from code.

2.1 Staff Roles and Intended Records

Role	Current named staff records	Authentication policy
SuperUser	Amador M. Raga	OTP only using registered admin email or SMS.
Legal	Lynn Frances A. Geronilla-Carneo; Roma Luisa M. Malvan	OTP only using registered admin email or SMS.
Finance	Cesar Malvan; Michael Angelo D. Lagaras; Trisha Mae A. Atendido	OTP only using registered admin email or SMS.

Audit and obsolete superadmin records should be removed or marked inactive unless they are intentionally reintroduced under a clear governance policy. The application should also filter inactive admins and ignore audit/auditor/superadmin roles when building the staff OTP list.

3. Authentication, OTP, and Admin Identity Model

The root authentication server remains the authority for claimant login, staff OTP, session cookies, login audit, and DevSuite data. The new model keeps claimant login unchanged but moves staff lookup from hardcoded QA entries to the admins table.

Step	Action	Source / Target	Server Authority
1	User enters email or SMS in AuthLogin.	Comp_AuthLogin.svelte	Normalize identifier: lowercase email or SMS digits.
2	Server looks for active staff.	admins	Match email_address, username, or sms_number. Exclude inactive/audit/superadmin records.
3	If no staff record is found, server looks for claimant.	members	Match email_address or sms_number.
4	Server generates OTP.	login_otps	Store hash, channel, member_id/admin_id, and expiry.
5	Server sends notification.	Svc_Transmittal -> TurboSMTP	In live mode, send real email and record notification_outbox evidence.
6	User verifies OTP.	login_otps	Hash compare, consume OTP, reset failed-attempt lock.
7	Server establishes session.	admin_session cookie or claimant active user payload	Staff receives admin_session. Claimant clears admin_session.

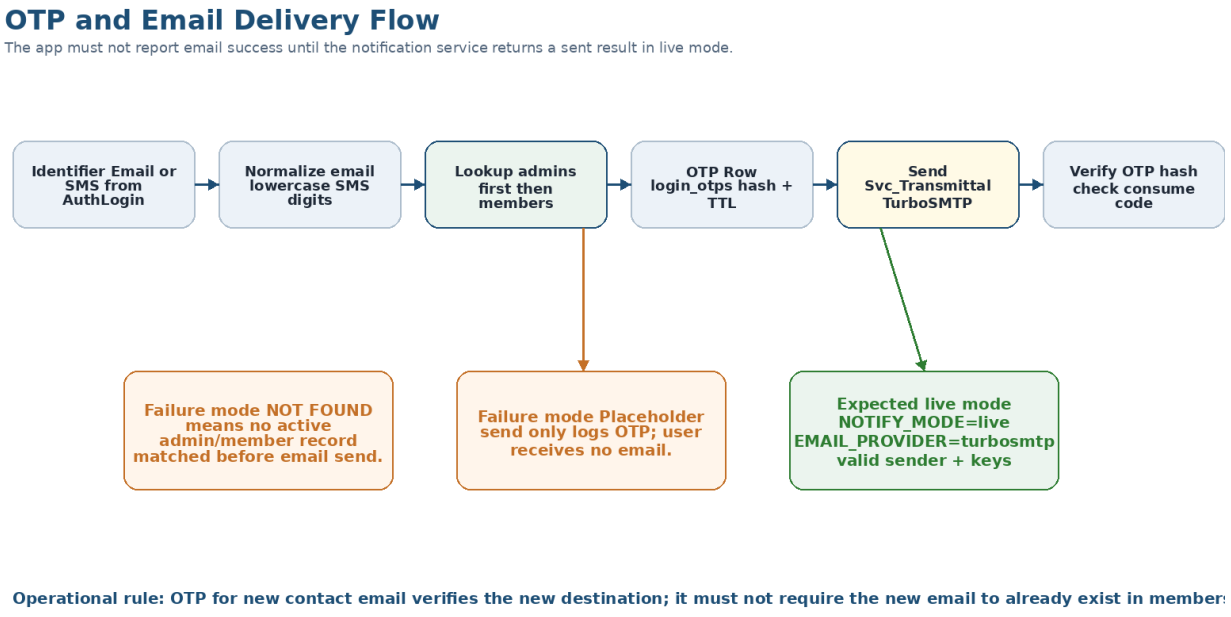

Figure 2. OTP lookup and TurboSMTP email-delivery flow.

Important failure interpretation
A NOT FOUND response during OTP request means the server could not match an active staff/admin or member record before attempting delivery. A TurboSMTP quota or sender problem should appear as a notification/send failure after a record is found, not as a record-not-found condition.

3.1 Required Environment for Live Email

The application must be explicit about live notification mode. In development mode, simulated notification responses are useful, but the login UI must not mislead users into believing an email was delivered when only a console placeholder ran.

Variable	Expected Value / Purpose

NOTIFY_MODE	live for real outbound delivery. dev/simulated modes must be labeled clearly.
EMAIL_PROVIDER	turbosmtp for the current implementation.
TURBO_SMTP_ENDPOINT	https://api.turbo-smtp.com/api/v2/mail/send unless TurboSMTP changes the API endpoint.
TURBO_SMTP_CONSUMER_KEY	TurboSMTP API credential.
TURBO_SMTP_CONSUMER_SECRET	TurboSMTP API credential.
NOTIFY_EMAIL_FROM	A verified/allowed sender in TurboSMTP.
NOTIFY_EMAIL_CC	Optional monitoring copy, useful during rollout.

4. TA_Admin Module - Updated Operational Specification

The TA_Admin v2.0 document remains valid for its core dashboard control model. It defines Archive Claims, Current Claims, and Overall Claims lenses; Pending/Accepted/Vouchered/Released/Declined/Bounced tabs; hearing-date freshness; document state badges; and ta_claim_history as the authoritative audit trail. This update adds the new admin identity and notification posture.

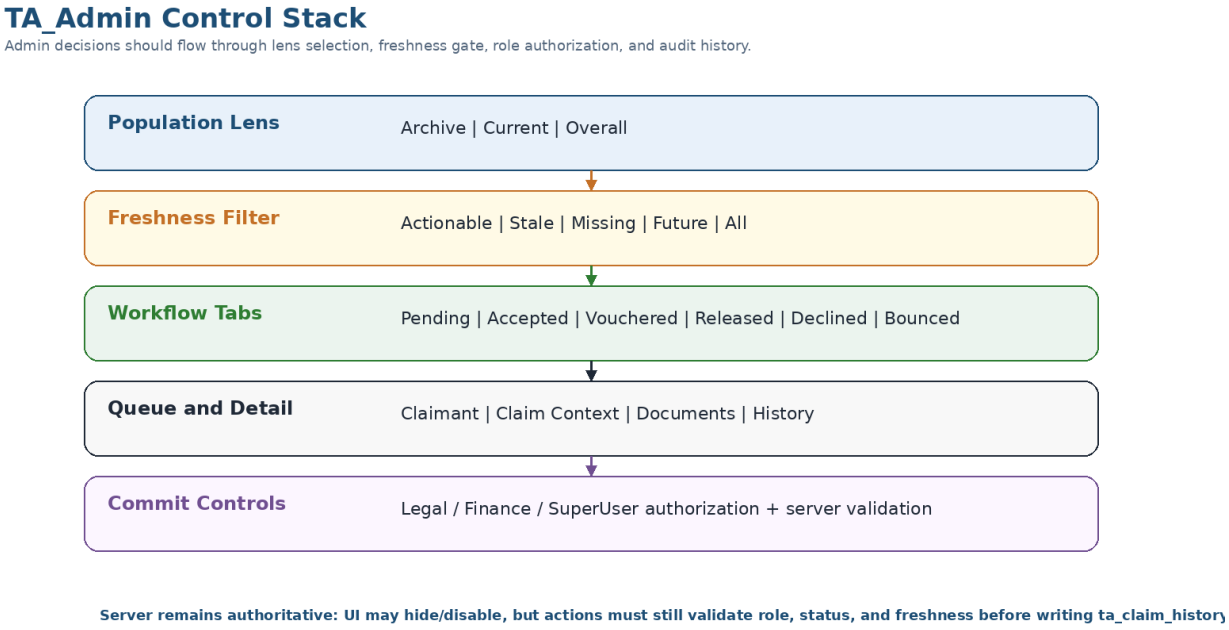

Figure 3. Recommended TA_Admin control stack.

4.1 Workflow and Authorization

Role	Status	Normal actions	Additional v3.0 requirement
Legal	Pending	Accept or Decline	Actor identity should come from admins/admin_session, not a department-password placeholder.
Legal	Accepted	Voucher or Decline	History row updated_by should store the named admin where possible.
Finance	Vouchered	Release or Decline	Release should generate notification evidence.
Finance	Released	Bounce	Bounce should preserve release context and actor identity.
SuperUser	Any permitted status	Inherited actions	Use explicit SuperUser role value consistently; do not mix Super User and SuperUser in storage.

4.2 Admin Source of Truth

TA_Admin should not depend on hardcoded role credentials for real staff identity. The root server may still provide adminCredentials or qaStaffCredentials to avoid breaking existing components, but those arrays should be assembled from the admins table.

- loadActiveAdmins(db) returns active staff rows for frontend credential display and staff OTP.
- findActiveAdminLogin(db, identifier) resolves staff OTP request.
- findActiveAdminById(db, admin_id) resolves OTP verification into a staff session.
- admin_session should carry id, name, username, role, department, loginGreeting, and previousLoginAt.

4.3 Notification Evidence

When TA_Admin actions change claim status, notifications should route through Svc_Transmittal so claimant and internal recipients are consistently handled. The outbox provides inspection evidence for sent, skipped, failed, and duplicate notifications.

Status change	Recipient expectation	Delivery channel
Any status creation/change	Claimant	Email; SMS remains governed by NOTIFY_SMS_ENABLED.
Accepted	Legal internal recipient if policy requires review/coordination	Email through Svc_Transmittal.
Vouchered	Finance recipient for release queue	Email through Svc_Transmittal.
Failed/Skipped notification	System/operator	Visible in notification_outbox; should not be silently ignored.

5. TA_Claimant Module - Updated Operational Specification

The existing claimant documentation correctly frames TA_Claimant as the claimant orchestration layer for identity context, document intake, AI validation, business-rule checking, ledger finalization, and status presentation. This iteration improves the user path around filing readiness, builder mode, and ledger history display.

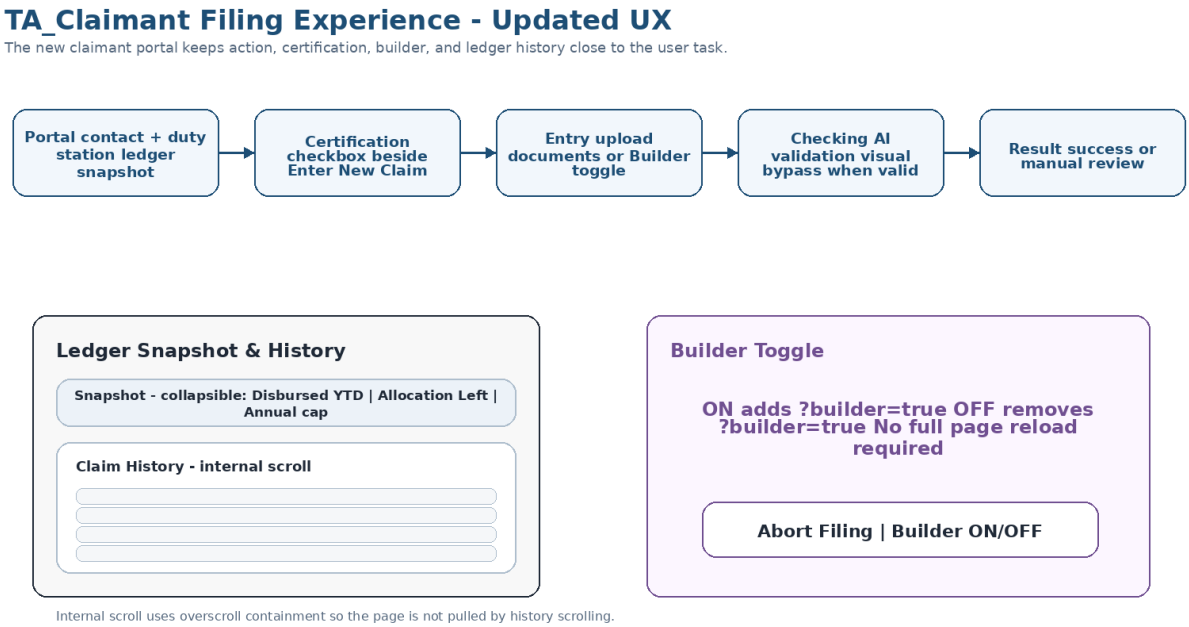

Figure 4. Updated claimant filing experience and ledger containment.

5.1 Certification and Enter New Claim Placement

The claimant must explicitly review contact and duty-station details before filing. The previous placement separated the certification checkbox from the claim-entry action, which made the relationship less intuitive. The updated design places the checkbox and Enter New Claim button in the same action block.

UI element	Updated behavior	Reason
Contact Information and Office/Duty Station cards	Remove visible OK/REVIEW labels if they add clutter; readiness remains enforced by logic.	The cards should focus on editable details rather than redundant badges.
Certification checkbox	Displayed directly beside the Enter New Claim button.	Makes the gating relationship obvious.
Enter New Claim	Disabled until certification is accepted.	Keeps the claimant in control while preserving policy.

5.2 Builder Toggle

Builder mode is still governed by the root URL parameter builder=true, but the claimant UI now provides a proper toggle. ON adds ?builder=true and OFF removes it. This is safer than a one-way enable action and makes QA/test mode visible to the operator.

State	URL	Result
Builder OFF	No builder parameter	CompTestCommander is hidden.
Builder ON	?builder=true	CompTestCommander is shown.
Toggle OFF	Remove builder parameter	Return to normal filing without a full page reload.

5.3 Ledger Snapshot and History Containment

The claimant ledger can grow enough to drag the whole page scroll. The updated design makes the ledger panel internally scrollable and uses collapsible sections: Snapshot, Claim History, and individual case groups.

- Use overscroll-behavior: contain on the history scroll area.
- Reduce line height, padding, and badge height in history rows.
- Keep the first few case groups open by default and allow older groups to collapse.
- Preserve master-detail history: ta_claims is the current claim view; ta_claim_history is the expandable audit detail.

6. Contact Verification and Notification Delivery

Comp_Contacts is now part of the authenticated claimant profile workflow. Its email assignment process differs from login OTP: login OTP verifies an existing identity, while contact-update OTP verifies ownership of a new email address before saving it to the members table.

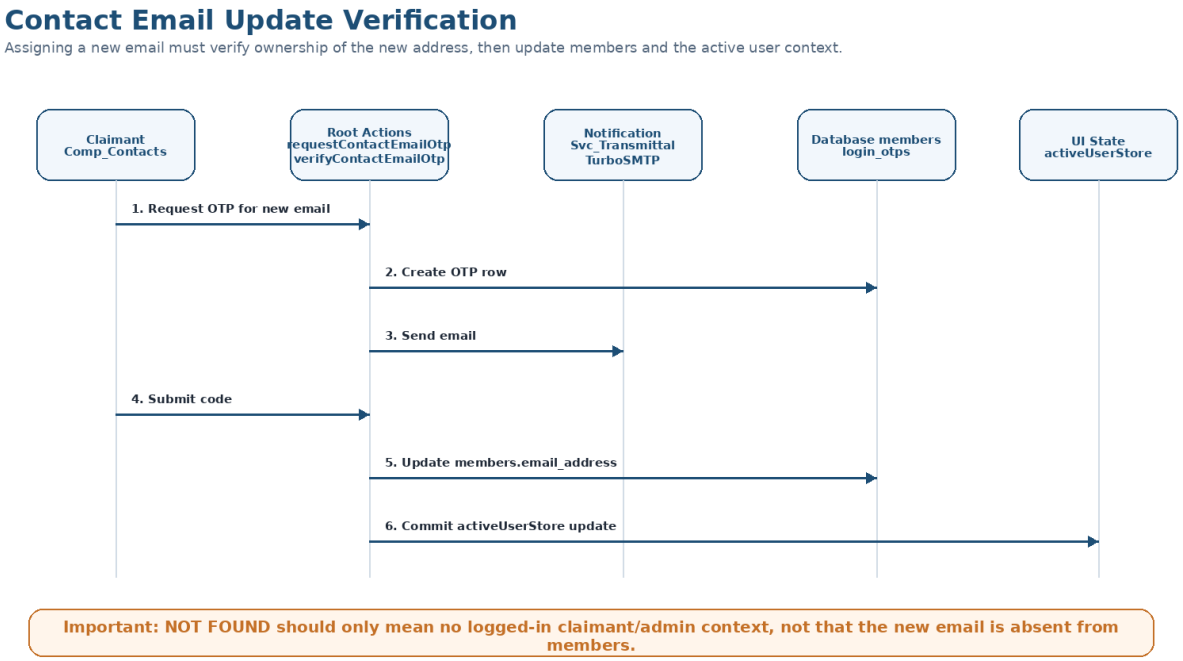

Figure 5. Contact email update and OTP verification sequence.

Workflow	Lookup rule	Expected behavior
AuthLogin OTP	Identifier must match active admins or members before OTP is sent.	NOT FOUND is valid when no account exists.
Contact email update OTP	New email does not need to exist in members yet. The logged-in claimant context supplies the member identity.	Server sends OTP to the new email, verifies code, then updates members.email_address.
Contact SMS update	Should mirror email verification policy where SMS delivery is available.	If SMS delivery is disabled, UI should disclose that clearly.

6.1 Svc_Transmittal as Common Notification Layer

Svc_Transmittal should be the common sender for OTP email, contact verification, TA status change notifications, and internal workflow notices. It centralizes validation, provider handling, outbox storage, duplicate detection, and status classification.

Status	Meaning	Operational use
sent	Provider accepted the email/SMS request.	Proceed and show success.
failed	Provider rejected or threw an error.	Show actionable failure and retain outbox error.
skipped	Missing target, disabled channel, duplicate, or validation skip.	Do not imply delivery.
queued	Reserved state for future asynchronous delivery.	Useful if delivery becomes background-processed.

Email quota note
TurboSMTP quota exhaustion is not the same as NOT FOUND. Quota or sender problems should surface as provider errors after the recipient record and destination are known.

7. Migration Tool Boundary Rules

The iteration clarified a recurring risk: Members migration and TA claims migration must be handled by separate route modules and separate server actions. Members migration must never delete TA claims. TA claims migration must never rewrite claimant master records.

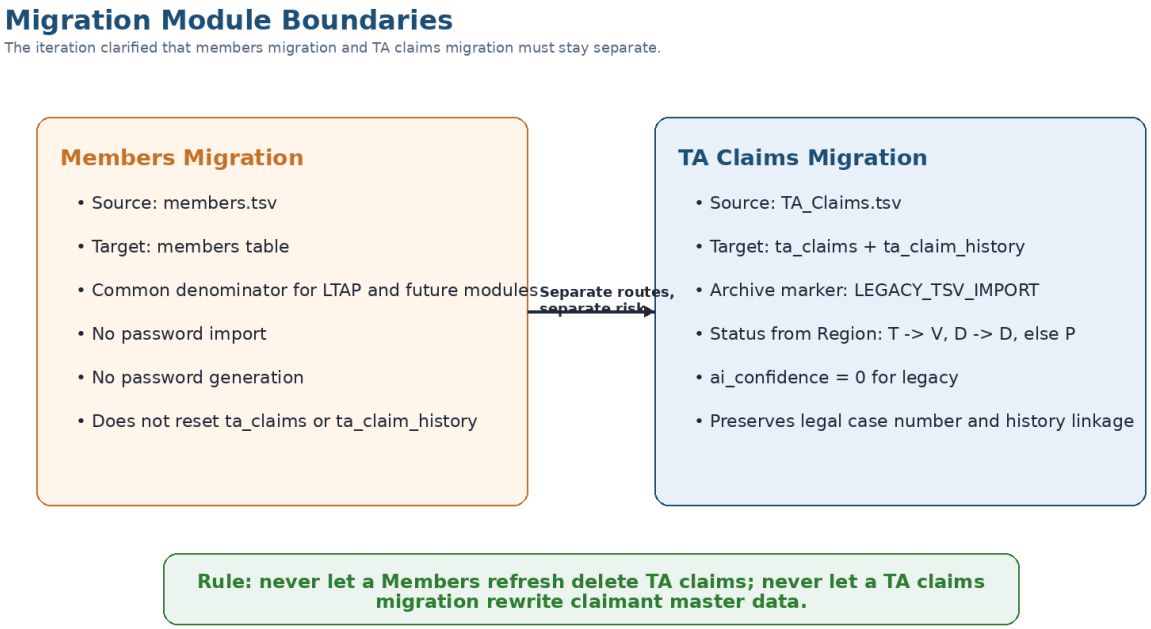

Figure 6. Migration module boundary rules.

Route	Source	Target	Must not do
/tools/migration/members	members.tsv	members	Must not delete ta_claims or ta_claim_history. Must not import or generate passwords.
/tools/migration/ta_claims	TA_Claims.tsv	ta_claims + ta_claim_history	Must not rewrite the claimant master registry. Must preserve claim_id history linkage.
/tools/migration/ta_pastdues	Future source	Future table or update policy	Should remain reserved until source and conflict policy are approved.

7.1 Members Migration Requirements

- Preserve current members schema.
- Refresh claimant/member registry from members.tsv.
- Do not import password and do not generate password.
- Protect downstream claim history by not truncating ta_claims or ta_claim_history.
- Report counts, skipped rows, and schema problems explicitly.

7.2 TA Claims Migration Requirements

- Mark legacy claims with document_hash = LEGACY_TSV_IMPORT.
- Use ai_confidence = 0 for legacy TSV records.
- Add legacy marker 0.1321 in remarks/voucher_info as provenance.
- Use Region-derived status: T -> V, D -> D, otherwise P.
- Create ta_claim_history rows linked to inserted ta_claims.id.
- Preserve legal case number in ta_claims.case_number and history remarks as needed.

8. Database Objects and Ownership

Object	Owner module/process	Key columns / concepts	Control note
admins	Root Auth / TA_Admin	admin_id, username, admin_name, role, department, email_address, sms_number, is_active, nullable password_hash	Named staff; OTP-only.
members	TA_Claimant / migration/members	member_id, contact fields, duty station, KYC fields	Claimant master registry.
login_otps	Root Auth / Contact verification	identifier, channel, member_id, otp_hash, expires_at, consumed_at	OTP codes must expire and be consumed.
login_attempt_locks	Root Auth	device_key, attempts, locked_until	Protects OTP brute force.
login_audit	Root Auth	username, display_name, role, login_type, success, failure_reason, logged_at	Supports usage trace and welcome-back messages.
notification_outbox	Svc_Transmittal	recipient_target, status, provider, response_json, error, event_key	Notification audit evidence.
ta_claims	TA_Claimant / TA_Admin / migration/ta_claims	status, amount, hearing_date, docs, document_hash	Claim source of truth.
ta_claim_history	TA_Admin / migration/ta_claims	claim_id, status, previous_status, updated_by, remarks	Permanent workflow audit trail.

8.1 Role Value Normalization

Use one storage spelling for SuperUser. The preferred stored role is SuperUser without a space. UI labels may display Super User if desired, but code should normalize before authorization checks. Avoid storing both SuperUser and Super User.

Stored value	Display label	Notes
SuperUser	Super User	Authoritative stored role for inherited permissions.
Legal	Legal	Legal review and voucher preparation duties.
Finance	Finance	Release and finance-side workflow duties.
Audit / Auditor	Audit	Not active in current staff OTP list unless policy reintroduces it.
superadmin	System Admin	Obsolete/demo role; remove or deactivate.

9. Operational Checklists

9.1 Before Replacing Auth Files

1. Back up src/routes/+page.server.ts and src/routes/+page.svelte.
2. Verify admins table has active Legal, Finance, and SuperUser staff.
3. Verify password_hash is nullable for OTP-only staff.
4. Run a CLI TurboSMTP test using the same .env credentials.
5. Confirm NOTIFY_MODE=live and EMAIL_PROVIDER=turbosmtp when expecting real email.
6. Restart the SvelteKit/Vite process after changing .env or server files.

9.2 OTP Troubleshooting

Symptom	Most likely cause	Check
NOT FOUND during AuthLogin OTP	Identifier is not in active admins or members.	Query admins.email_address/username/sms_number and members.email_address/sms_number.
UI says sent but no email arrives	Placeholder send path or NOTIFY_MODE not live.	Search root server for sendOtpEmailPlaceholder and check .env.
TurboSMTP CLI sends but app does not	App not using Svc_Transmittal or environment not loaded in app process.	Restart app and inspect notification_outbox.
Contact email update says NOT FOUND	Server action is missing or wrongly requiring the new email to pre-exist.	Confirm requestContactEmailOtp and verifyContactEmailOtp actions exist.
OTP verifies but staff lands incorrectly	admin_session or buildStaffSession shape mismatch.	Inspect returned member object and cookie payload.

9.3 Claimant UX QA Checklist

- Certification checkbox and Enter New Claim are adjacent and visually connected.
- Builder ON/OFF toggles the URL parameter without a full reload.
- Ledger history scrolls inside the panel; scrolling history does not drag the page.
- Snapshot and case groups collapse and expand without losing history data.
- Contact email verification updates activeUserStore after successful save.

9.4 Admin QA Checklist

- TA_Admin identifies the named admin actor, not just a generic department.
- Legal sees Legal actions; Finance sees Finance actions; SuperUser inherits actions.
- Stale/missing hearing date gates still block ordinary workflow actions server-side.
- Document state badges distinguish current uploads from legacy TSV records.
- Each workflow transition writes ta_claim_history and triggers notification where policy requires.

10. Diagrams and Wireframes

The diagrams in this document are generated visual aids. They are not live browser screenshots because no running application/browser session was available in the documentation generation environment. They are suitable for architecture review, onboarding, and implementation planning.

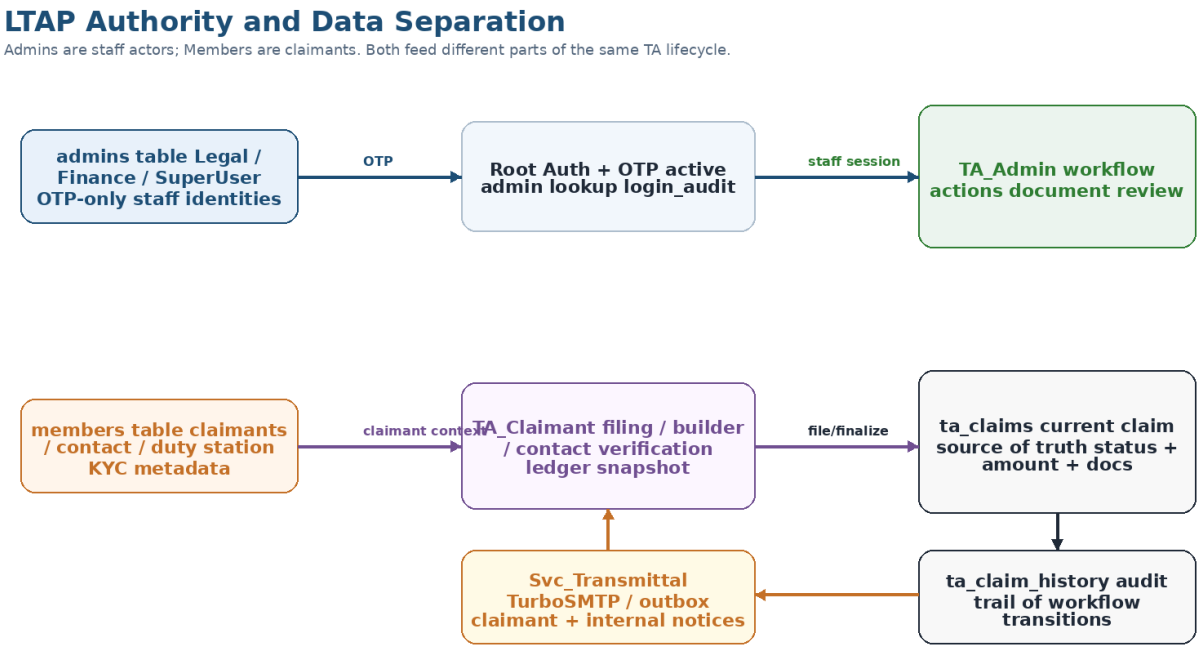

Figure A. LTAP authority and data separation.

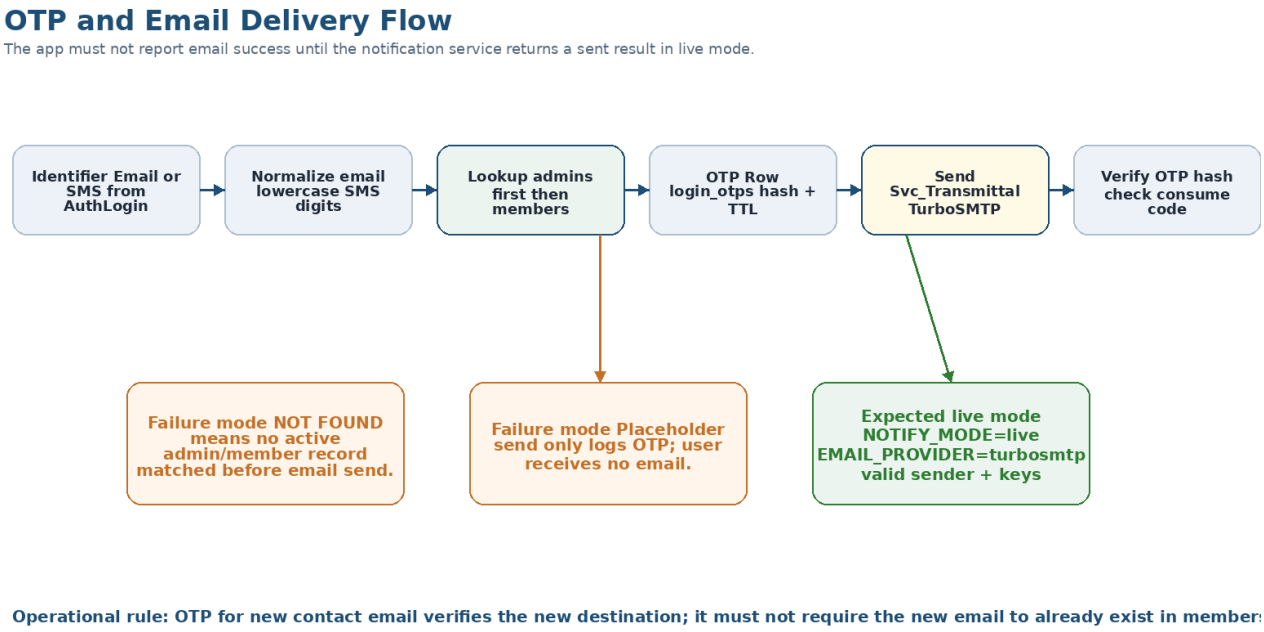

Figure B. OTP and email delivery flow.

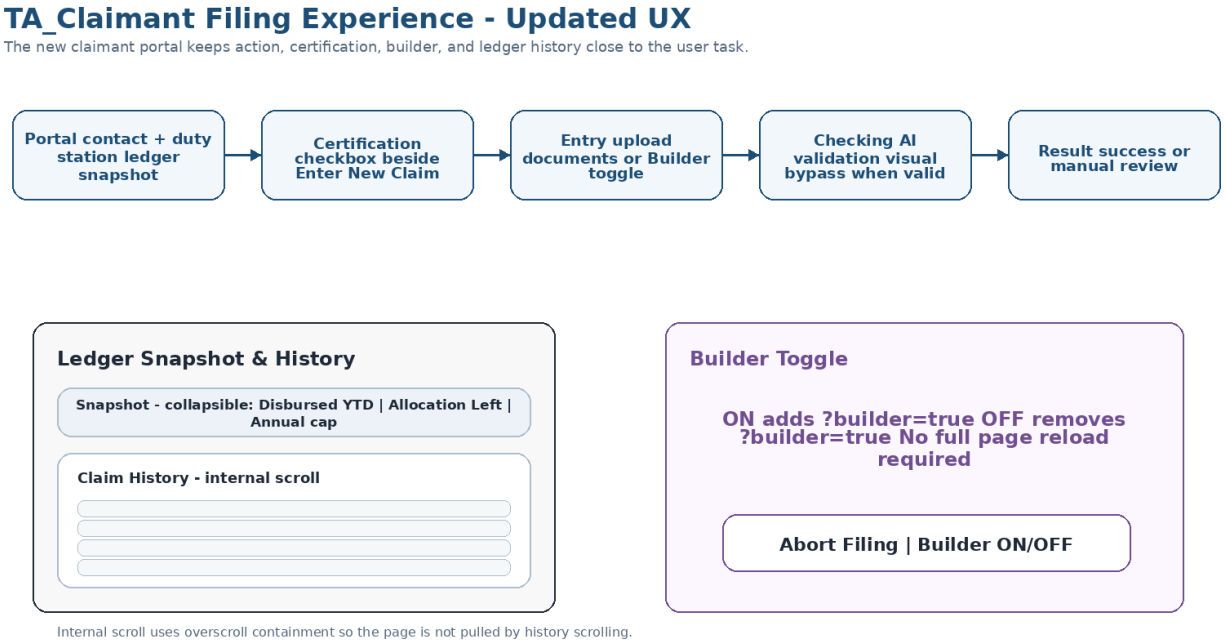

Figure C. Updated claimant UX and ledger containment.

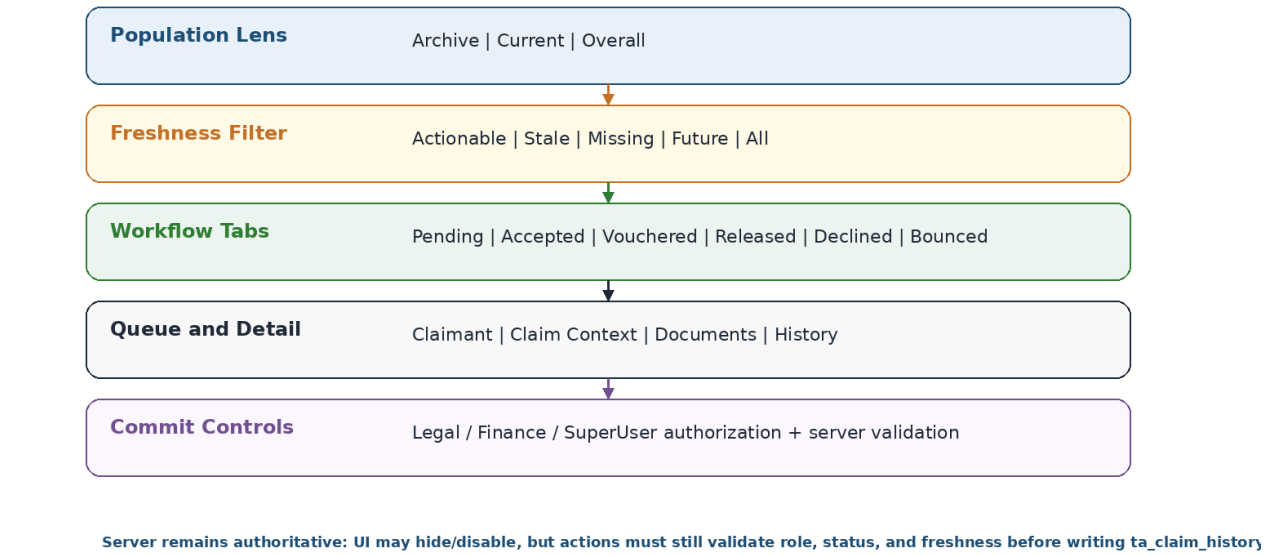

Figure D. Admin dashboard control stack.

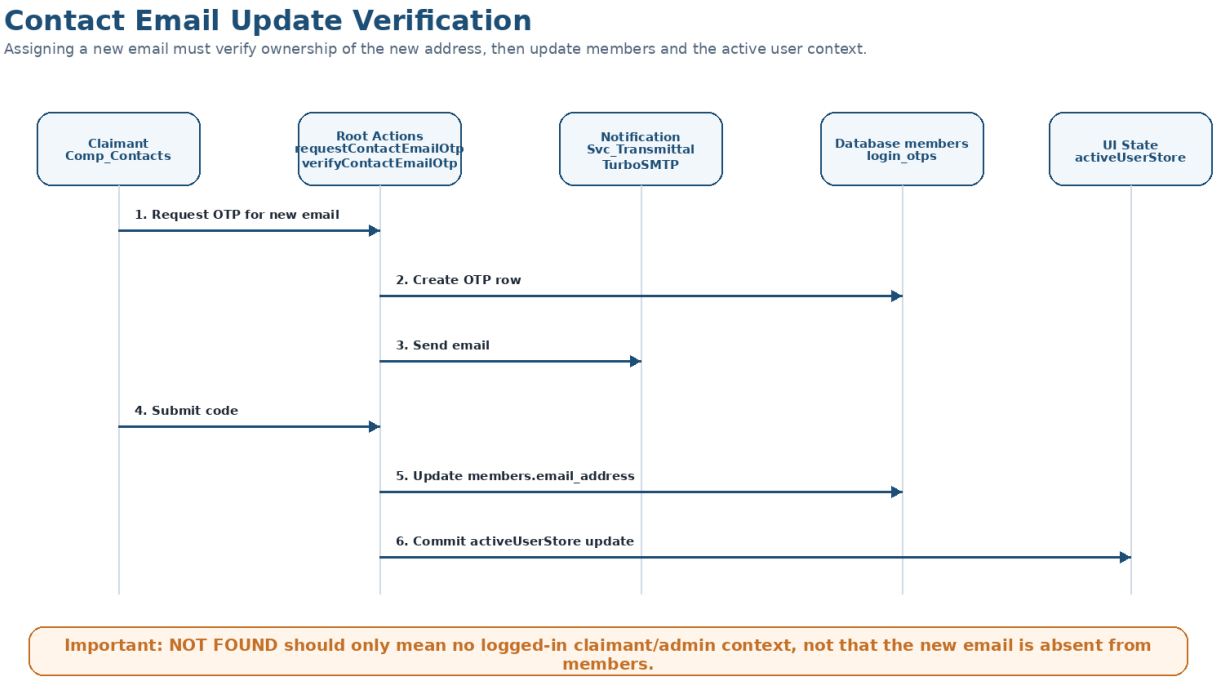

Figure E. Contact email verification.

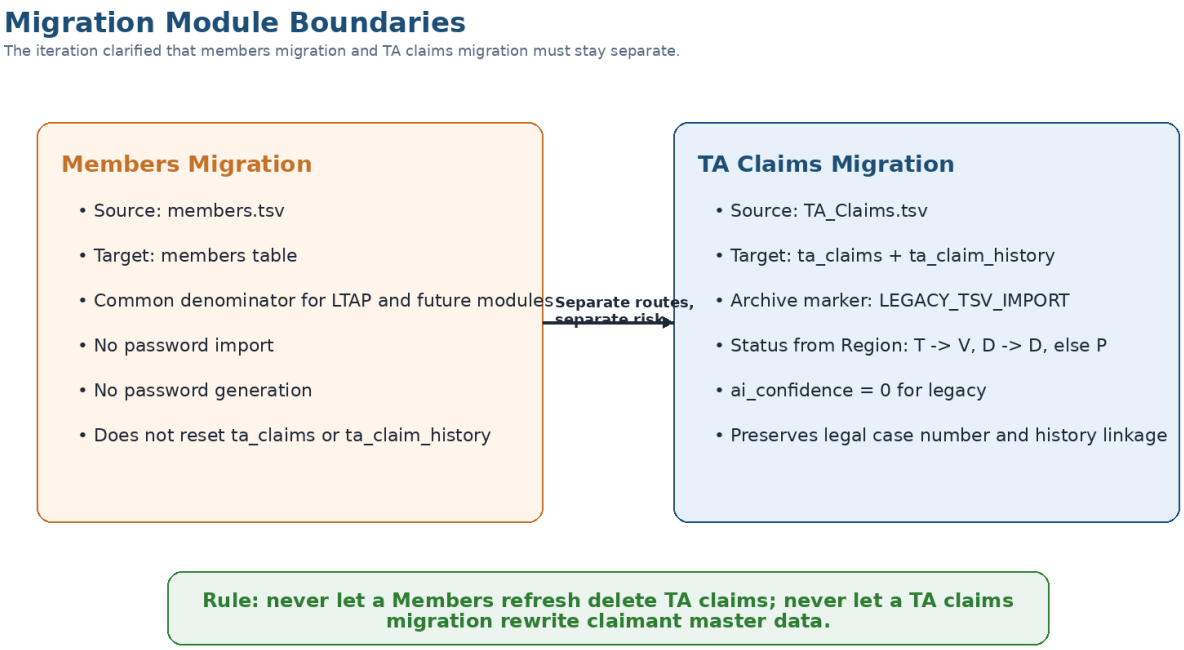

Figure F. Migration boundaries.

11. Conclusion

The current iteration strengthens LTAP by replacing hardcoded staff identity with an admins table, enforcing OTP-only staff authentication, centralizing email delivery through Svc_Transmittal and TurboSMTP, fixing the contact-email verification path, clarifying migration tool boundaries, and improving claimant usability around filing readiness, builder mode, and

ledger history. The Admin and Claimant modules should continue sharing the same claim source of truth, while preserving clear ownership between members, admins, ta_claims, ta_claim_history, login_audit, and notification_outbox.