SMS System Plan — Text messaging for Pilot

> **Status**: Draft (2026-04-17) — pre-implementation **Goal**: Let tenants send/receive SMS (and MMS) from their Pilot account the same way they send emails, scoped to a company-owned Twilio number, threaded into conversations, and reachable from projects + contacts.

What already exists (audit — verified 2026-04-17)

Piece	Table/File	State	Notes
Inbound webhook capture	`sms_inbound`	1 row, basic	`from_number`, `to_number`, `message_body`, `raw_payload`, `processed` — no threading, no contact link
Per-company Twilio number	`company_phone_numbers`	1 row (Air4, +18882981641 toll-free, unverified)	`twilio_sid`, `number_type`, `purpose`, `verification_status` (pending/approved for 10DLC/toll-free)
Per-user Twilio creds	`user_email_settings.sms_*`	fields exist	BYO Twilio account path — likely deprecate for platform-managed model
Outbound stub (unused)	`outgoing_emails.message_type='sms'`	**0 rows of 304**	Dead path. AWS SNS, one-way, no threading.
System SMS	`config/core/mfa.php::sendSMS()`	in use (login MFA)	AWS SNS, no Twilio, direct send not queued
Planned (not built)	airchat-mobile-app.md	design only	`sms_conversations` + `sms_messages` in Phase 3 roadmap

Conclusion: Infrastructure is seeded but incoherent. Outbound, threading, and project/contact linking don't exist.

Architecture decision: separate SMS pipeline (NOT bolted onto email)

Reject reusing `outgoing_emails` for SMS threads. Reasons:

- Email queue is entity-based (one email per `entity_id`); SMS is thread-based (many messages per conversation).
- Email backends are SES/Gmail/SMTP; SMS backend is Twilio (already in `company_phone_numbers`).
- Compliance (STOP/HELP/START keywords, quiet hours, 10DLC/toll-free verification gating, opt-in audit trail) has no home in email schema.
- MMS media handling, delivery receipts, Twilio SIDs have no columns.

Keep `outgoing_emails` SMS stub retired. Migrate `sendSMS()` MFA path to the new pipeline in Phase 2.

Schema

```
-- Conversations / threads
CREATE TABLE sms_threads (
    id INT AUTO_INCREMENT PRIMARY KEY,
    company_id INT NOT NULL,
    company_phone_id INT NOT NULL,       -- FK company_phone_numbers.id (our
number)
    peer_phone VARCHAR(20) NOT NULL,     -- E.164 customer number
    contact_id INT DEFAULT NULL,         -- FK contacts.id, nullable
    project_id INT DEFAULT NULL,         -- FK crm_projects.id, nullable
(last linked project)
    status ENUM('active','resolved','closed','blocked') DEFAULT 'active',
    opt_in_status ENUM('unknown','opted_in','opted_out') DEFAULT 'unknown',
    opt_in_source VARCHAR(50) DEFAULT NULL,   -- 'web_form', 'verbal',
'reply_yes', etc.
    opt_in_at DATETIME DEFAULT NULL,
    opt_out_at DATETIME DEFAULT NULL,
    last_message_at DATETIME DEFAULT NULL,
    last_direction ENUM('inbound','outbound') DEFAULT NULL,
    unread_count INT DEFAULT 0,
```

```
    message_count INT DEFAULT 0,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE KEY uq_thread (company_phone_id, peer_phone),
    INDEX idx_company (company_id, status, last_message_at),
    INDEX idx_contact (contact_id),
    INDEX idx_project (project_id)
);

-- Individual messages (inbound + outbound)
CREATE TABLE sms_messages (
    id INT AUTO_INCREMENT PRIMARY KEY,
    thread_id INT NOT NULL,
    company_id INT NOT NULL,
    direction ENUM('inbound','outbound') NOT NULL,
    body TEXT,
    media_urls LONGTEXT,                  -- JSON array of Twilio media URLs
(MMS)
    twilio_sid VARCHAR(64) DEFAULT NULL, -- Twilio message SID (outbound or
inbound)
    status
ENUM('queued','sending','sent','delivered','undelivered','failed','received')
DEFAULT 'queued',
    error_code VARCHAR(20) DEFAULT NULL,
    error_message TEXT DEFAULT NULL,
    segment_count TINYINT DEFAULT 1,     -- Twilio reports num_segments
    price_amount DECIMAL(10,6) DEFAULT NULL,
    price_currency VARCHAR(8) DEFAULT NULL,
    sender_user_id INT DEFAULT NULL,     -- which user sent (outbound only)
    entity_type VARCHAR(40) DEFAULT NULL,--
'project','manual','campaign','mfa', etc.
    entity_id INT DEFAULT NULL,
    scheduled_at DATETIME DEFAULT NULL,
    sent_at DATETIME DEFAULT NULL,
    delivered_at DATETIME DEFAULT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    INDEX idx_thread (thread_id, created_at),
    INDEX idx_status (status, scheduled_at),
    INDEX idx_twilio (twilio_sid),
    INDEX idx_entity (entity_type, entity_id)
);

-- Company-level Twilio account credentials (one row per company that owns a
number)
CREATE TABLE company_twilio_credentials (
```

```
    company_id INT PRIMARY KEY,
    account_sid VARCHAR(64) NOT NULL,
    auth_token_encrypted TEXT NOT NULL,  -- encrypted at rest
    account_type ENUM('subaccount','byo') DEFAULT 'subaccount',
    messaging_service_sid VARCHAR(64) DEFAULT NULL, -- optional Twilio
Messaging Service
    webhook_secret VARCHAR(64) DEFAULT NULL,        -- validate inbound
webhooks
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);

-- Templates (mirrors email_templates, SMS-specific)
CREATE TABLE sms_templates (
    id INT AUTO_INCREMENT PRIMARY KEY,
    company_id INT NOT NULL,
    name VARCHAR(100) NOT NULL,
    category VARCHAR(50) DEFAULT NULL,   -- 'project','onboarding','reminder'
    body TEXT NOT NULL,
    variables LONGTEXT,                  -- JSON array of {{vars}}
    is_active TINYINT(1) DEFAULT 1,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    INDEX idx_company (company_id, is_active)
);

-- Amend sms_inbound to link into the thread once processed
ALTER TABLE sms_inbound
    ADD COLUMN thread_id INT DEFAULT NULL AFTER id,
    ADD COLUMN message_id INT DEFAULT NULL AFTER thread_id,
    ADD INDEX idx_thread (thread_id);
```

Design notes:

- Thread uniqueness is `(company_phone_id, peer_phone)` — if the same customer texts two of our numbers, they're two threads (correct — different business contexts).
- `contact_id` and `project_id` are **denormalized last-link** on the thread, messages carry `entity_type/entity_id` for full history.
- `unread_count` surfaces sidebar badge without a join (same pattern as AirChat).

Backend services & files

New services

1. **`storage/services/scripts/sms-queue-service.php`** — daemon (Service Manager ID TBD, e.g. 30)
   - Polls `sms_messages WHERE status='queued' AND (scheduled_at IS NULL OR scheduled_at <= NOW())`
   - Sends via Twilio REST, stores `twilio_sid`, updates status → `sending` → (webhook updates to `sent`/`delivered`/`failed`)
   - Retry with exponential backoff (mirror email daemon pattern)
   - Respects 10DLC/toll-free verification: if `company_phone_numbers.verification_status != 'approved'`, queue stays held with `error_message='number_unverified'` (avoid Twilio carrier rejection)
   - Quiet hours honored: US rule = no marketing texts 9pm–8am recipient local time (phone-to-timezone lookup via area code or `contacts.timezone`)

2. **`pilot/includes/handlers/twilio-webhook-handler.php`** — public endpoint (validate signature)
   - Routes: `inbound-sms`, `status-callback`
   - Inbound: upsert `sms_threads`, insert `sms_messages(direction='inbound')`, increment `unread_count`, match contact by phone, trigger in-app notification
   - Auto-handles `STOP` / `START` / `HELP` keywords → sets `opt_in_status`, sends compliance auto-reply
   - Status callback: update `sms_messages.status` + `delivered_at`/`error_*`

3. **`pilot/includes/services/twilio-service.php`** — `TwilioService` class
   - Wraps Twilio REST (no SDK — use curl, like the rest of the codebase)
   - `sendMessage($companyId, $fromNumber, $to, $body, $mediaUrls=[])`
   - `purchaseNumber($companyId, $areaCode, $tollFree=false)`
   - `submitTollFreeVerification($companyPhoneId, $businessData)` → stores `verification_sid`
   - `checkVerificationStatus($companyPhoneId)` → polls + updates `verification_status`

4. **`pilot/includes/handlers/sms-handler.php`** — page/AJAX handler for SMS Management

- Actions: `list_threads`, `get_thread`, `send_message`, `mark_read`, `link_contact`, `link_project`, `opt_out`

Modified

- `config/core/mfa.php::sendSMS()` → route through `TwilioService::sendMessage` (MFA becomes just another `entity_type='mfa'` outbound message). Retire AWS SNS.
- `outgoing_emails.message_type='sms'` → mark deprecated; optional cleanup migration later.
- Service Manager registry → add `sms-queue-service` (systemd daemon).
- AirChat mobile unified inbox → query `sms_threads` alongside airchat_conversations.

UI / touchpoints

Phase 1 surfaces (minimum viable)

1. **Project detail page** — "Send Text" button next to "Send Email"
   - Opens modal: pre-fills recipient from `project.primary_contact.phone`, template picker, variable substitution, send
   - Message stored as `entity_type='project', entity_id=<project_id>`, thread gets `project_id` updated
2. **Contact detail page** — "Send Text" button
   - Same modal, recipient = `contact.phone`, `entity_type='contact'`
3. **SMS Management page** (new, mirrors Email Management)
   - Tabs: Threads (default), Sent, Templates, Numbers, Compose
   - Threads tab: Air4List of `sms_threads` — columns: peer, contact name, last message preview, unread badge, status, last_message_at
   - Row click → thread detail modal: full message history + reply input (like AirChat conversation view)
   - Sent tab: Air4List of `sms_messages` where direction=outbound (with same eye-icon view action pattern we just added to email)
   - Numbers tab: manage `company_phone_numbers`, purchase new, submit verification
4. **Sidebar badge** — red pill on SMS menu item when `SUM(unread_count) > 0` (reuse `notification-badge-handler.php` pattern from AirChat)

Phase 2 surfaces

5. **Contact import/edit** — validate + normalize phone to E.164, track opt-in source
6. **Airmail campaigns** — add SMS channel (reuse templates, target `contacts` with opted-in phones)
7. **Quiet hours & timezone** — admin settings panel

Phases / milestones

Phase	Scope	Outcome
1	Schema migration + TwilioService + sms-queue daemon + webhook handler + STOP/HELP keywords	Can send/receive SMS end-to-end (via API only, no UI)
2	SMS Management page (Threads, Sent, Compose tabs) + sidebar badge	Can use SMS in browser
3	"Send Text" from project + contact pages + templates	Project/contact integration
4	Number management UI (purchase + toll-free verification submission)	Tenants self-serve 800 numbers
5	Airmail SMS campaigns + quiet hours + opt-in audit trail	Marketing-grade SMS
6	Retire `outgoing_emails` SMS stub + migrate MFA to sms_messages	Cleanup

Phase 1-3 is the MVP for "send texts from projects" requested in this conversation.

Compliance checklist (must-haves before Phase 2 ships)

- [ ] STOP keyword auto-handled → `opt_in_status='opted_out'`, auto-reply "You're unsubscribed..."
- [ ] START / UNSTOP re-opt-in
- [ ] HELP keyword → configurable auto-reply per company
- [ ] Block outbound when `opt_in_status='opted_out'` (hard gate, raise error)
- [ ] Block outbound when `company_phone_numbers.verification_status != 'approved'` for toll-free (Twilio rejects anyway)
- [ ] Audit trail: every outbound has `sender_user_id`, every opt-in change has source + timestamp

- [ ] Twilio webhook signature validation (prevent spoofed inbound)

Open decisions (need input)

1. **Twilio account model** — Air4 master account with per-tenant subaccounts (cleaner billing, we collect + mark up) vs BYO (tenants bring their own Twilio)? `user_email_settings.sms_*` suggests BYO path was started, `company_phone_numbers.twilio_sid` suggests platform-managed. **Recommend platform-managed subaccounts** — simpler tenant UX, enables us to resell.
2. **Per-user vs per-company phone numbers** — one shared company number (all users send from it) vs per-user numbers? Recommend **per-company** for v1, per-user later if needed.
3. **Service ID for sms-queue daemon** — next available (check service-registry.md).
4. **Credits/storage model** — should SMS consume AI credits? Or dedicated "SMS credits" counter on `companies`? Twilio costs ~$0.008/SMS + ~$0.02/MMS + verification fees.
5. **AirChat mobile integration** — fold SMS threads into unified inbox from day 1 (yes, already planned).

First implementation step

Once Open Decisions #1 + #4 are answered:

1. Write migration: `storage/migrations/2026-04-17-sms-system.sql` (create tables above)
2. Scaffold `TwilioService` with `sendMessage()` + auth wiring
3. Scaffold `sms-queue-service.php` daemon
4. Scaffold `twilio-webhook-handler.php` with signature validation + STOP/HELP
5. Manual end-to-end test: queue → send → webhook status callback → verify row updates
6. Then UI (Phase 2+)