HIPAA-SAFE AUTOMATION, IN THE OPEN

The clinic *stack.*

One architecture. Four systems sharing a single safety spine.

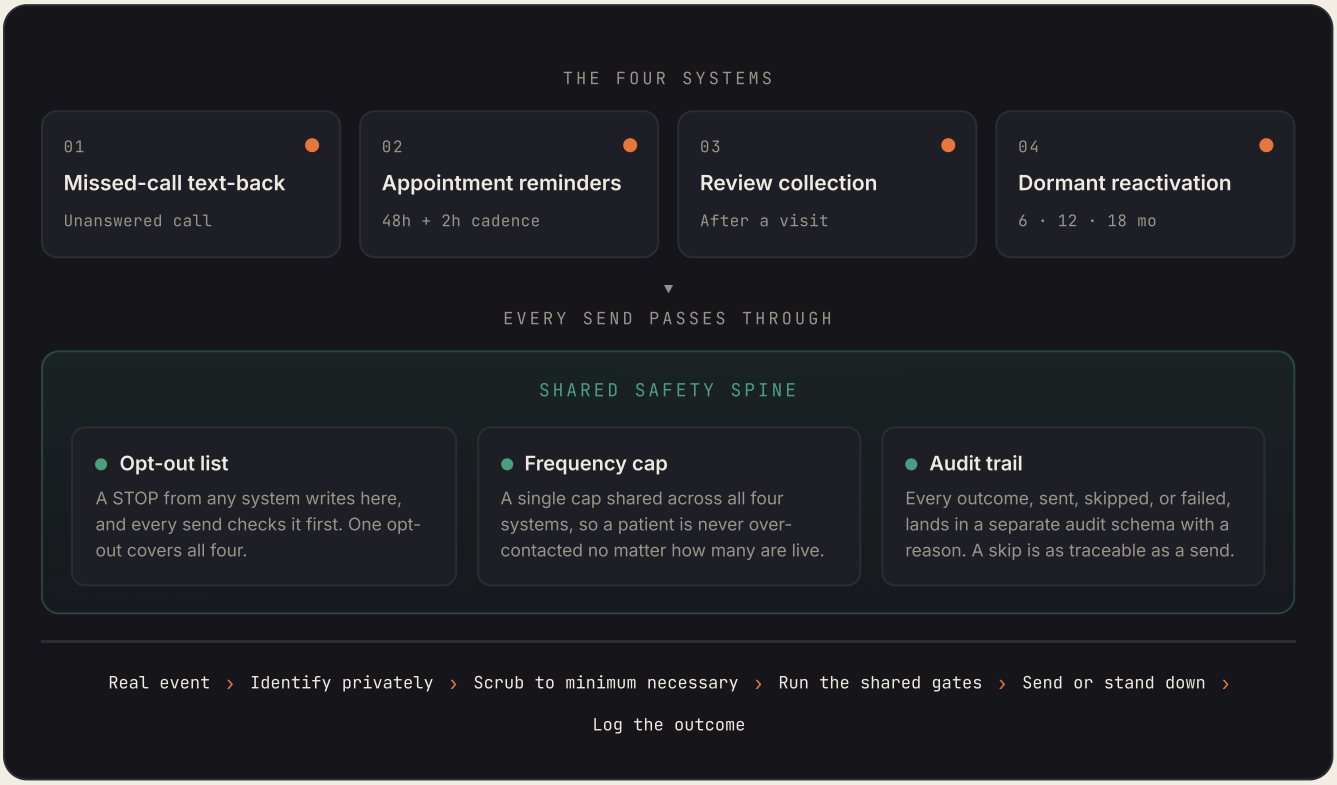
A working document from a year of building HIPAA-safe automation for independent dental practices. One page for the architecture and the shared safety spine, then one page per system: what it replaces, how it works, the compliance gates it never skips, and how it fails safe when something goes wrong.

How to read this. Every figure in this document is a design parameter, not a performance claim. The systems run in shadow mode against synthetic data. No patient data, no metrics, no pricing.

THE ARCHITECTURE

One architecture, one *safety spine.*

Not four separate tools. Four systems that share a single safety spine and hand off to each other, so one opt-out, one frequency cap, and one audit trail cover all of them at once.



One spine, not four

The opt-out list, the frequency cap, and the audit trail are shared infrastructure. A system is mostly the logic that decides when to reach out. The safety is common to all of them.

Fail-closed by default

If any check errors, the default is to not send. A skipped message is the safe outcome, and it is logged with a reason, the same as a sent one.

Tenant-isolated, portable

By default each system runs on a shared, BAA-covered stack, with every clinic isolated as its own tenant by row-level security and its own sending number. A practice that wants its own dedicated instance can have it deployed to infrastructure it owns, on request; either way the workflow and the records stay exportable.

SYSTEM 01

FIELD STUDY · SYNTHETIC DATA

Missed-call text-back

Every unanswered call gets a HIPAA-safe SMS within about a minute. Branded, no PHI, with a callback or booking link.

What it replaces

Today the front desk has to notice the missed call, find a free minute, look up who it was, and text them back by hand. In a clinic running full chairs that minute almost never comes, so the call just goes cold and the patient books somewhere else.

Why it's hard

Texting a patient is regulated. Send at 2am, miss an opt-out, or let a name slip into the message and a small convenience turns into a violation. The hard part was never sending an SMS. It is sending one that stays safe every single time, and being able to prove it afterward.

How it works

01 Catch the event, verified
Twilio posts the missed call to an n8n webhook. The request is checked against its `x-twilio-signature` (HMAC-SHA1) so only real Twilio events run, and an idempotency key drops duplicate retries before anything fires.

02 Identify privately
The caller number is normalized and hashed with HMAC-SHA256 plus a server-side pepper before it touches storage. The clinic config loads by tenant, so nothing cross-tenant is reachable.

03 Scrub, then draft
The prompt carries only clinic config (name, booking link, callback number, language) and a has-history flag, never a patient identifier, so Claude drafts the SMS in the clinic's voice with no PHI going in, and a regex scrub as a secondary backstop on the way out.

04 Gate, then send and log
The draft runs the unified safety check. Only if every gate passes does Twilio send. The run, the gate results, and the message id then write to separate `phi`, `ops`, and `audit` schemas.

Under the hood

- Multi-tenant: every record routes by clinic, so one clinic can never reach another's data.
- Idempotent and deduped end to end, so a retried webhook never double-sends.
- Phone numbers are referenced everywhere by a peppered HMAC-SHA256 hash; the raw number lives only in the RLS-protected phi schema, and logs carry just the last four digits.
- Schema separation keeps `phi`, `ops`, and `audit` apart.
- A roughly 60 second grace window holds the text so a front-desk callback can stand it down first; with no callback, the SMS goes out right after.
- Runs in shadow mode against synthetic data, logging the send instead of delivering.

By default it runs as a managed service: one operator-run n8n and a shared Postgres on BAA-covered AWS infrastructure, with every clinic isolated as its own tenant by row-level security and its own sending number, so no clinic can reach another clinic's data. Because it is built on standard n8n and Postgres, a practice that wants its own dedicated instance can have the whole thing deployed to infrastructure it owns, on request. Either way the workflow and the records stay exportable, never locked to a platform.

Compliance gates

✓ Opt-out (STOP) honored before every send	TCPA §227(b)
✓ Frequency cap shared across all four systems	TCPA
✓ Quiet-hours window, no overnight texts	TCPA
✓ No PHI placed in the outbound text	HIPAA §164.502(b)
✓ Inbound webhook authenticity	x-twilio-signature · HMAC-SHA1
✓ If any check errors, the default is do not send	Fail-closed

When it breaks

- A duplicate webhook never double-texts. The idempotency key on the call id means a Twilio retry is recognized and dropped.
- If a safety check itself errors, the run fails closed. The default is no message, never a message sent on a guess.
- If the draft step is slow or unavailable, the system falls back to a fixed, pre-approved template rather than leaving the patient in silence.
- Every outcome, whether sent, skipped, or failed, lands in the `audit` schema with a reason. A skipped text is as traceable as a sent one.

TRIGGER	RESPONSE	PHI IN THE TEXT	ON ERROR
Unanswered call	About a minute	None	Do not send

n8n · Twilio · Anthropic Claude · Postgres on AWS RDS

▷ What going live requires

Everything above runs today on synthetic data, in shadow mode. Turning it on for a real clinic and real patients clears a compliance and integration bar first. Here is that bar, in plain terms.

01 **Carrier registration (A2P 10DLC)**
The clinic's texting brand and campaign get registered with the mobile carriers before the first real text-back. It keeps a legitimate reply from being filtered as spam, and ties every message to a named, accountable sender.

02 **A signed BAA, on BAA-covered infrastructure**
A Business Associate Agreement is in place, and the system runs on HIPAA-covered infrastructure under that BAA: an isolated per-clinic tenant by default, or a dedicated instance the practice owns on request. Nothing about a caller touches a service that is not under that agreement.

03 **A one-time consent attestation**
At setup the clinic attests, once, that texting back a missed caller is consistent with its patient consent. The system records that attestation and re-checks the opt-out list before every send.

04 **A real phone line to watch**
The clinic's number is routed so a genuinely missed call posts an event the system can act on. A connected call or a duplicate webhook never triggers a text.

05 **A shadow period, then one clinic at a time**
Every new clinic starts in shadow mode, logging the text it would have sent with zero real messages, until a full day of calls has been watched. Only then does the live switch flip, and only for that clinic.

SYSTEM 02

FIELD STUDY · SYNTHETIC DATA

Smart appointment reminders

Two-touch reminder cadence (48h + 2h) with confirmation parsing and auto-reschedule offers. Tuned to each clinic's voice.

What it replaces

Right now reminders are a person working down tomorrow's schedule, calling or texting each patient, and writing down who confirmed. It is the first thing that gets dropped on a busy day, which is exactly the day no-shows cost the most.

Why it's hard

A reminder is only useful if the reply is understood. "Can we do Thursday instead" has to become a reschedule, not a confirmation, and a "STOP" has to be honored instantly and for good. Reading intent from a free-text reply, without ever storing that reply as PHI, is the hard part.

How it works

01 **Read the schedule**
A scheduled n8n run reads each clinic's upcoming appointments, deduped by appointment id so a reminder is never queued twice.

02 **Two-touch cadence**
A 48-hour and a 2-hour reminder, each one gated and quiet-hours aware, written in the clinic's voice.

03 **Parse the reply**
Inbound SMS is classified as confirm, reschedule, or stop with Anthropic Claude. A STOP writes straight to the opt-out list; the reply is classified in-workflow and its raw body is never stored as PHI.

04 **Offer the next slot**
A reschedule reply gets the next open time, and the decision and the message id are logged to the audit trail.

Under the hood

- Multi-tenant and deduped by appointment id.
- Shares the same opt-out list and frequency cap as the other systems, so one cap covers all of them.
- Built and shadow-tested on AWS RDS, running in shadow mode against synthetic data.

> Like the others, it runs on the shared, BAA-covered AWS stack by default with the clinic isolated as its own tenant, or on a dedicated instance the practice owns on request. The cadence, the reschedule logic, and every logged reply stay scoped to that one clinic and exportable to it.

Compliance gates

✓ Opt-out (STOP) honored before every send	TCPA §227(b)
✓ Frequency cap shared with the other systems	TCPA
✓ Quiet-hours window	TCPA
✓ Reply body never stored as PHI	HIPAA §164.502(b)
✓ If any check errors, the default is do not send	Fail-closed

When it breaks

- An ambiguous reply is never guessed at. If intent is not clear, it routes to the clinic rather than auto-confirming a slot that is not really booked.
- A STOP wins over everything. It is honored before the next send and written to the shared opt-out list the moment it arrives.
- Double-queueing is impossible. The appointment id dedup means rescheduling does not stack a second reminder on top of the first.

CADENCE	REPLY TYPES	REPLY STORED AS PHI	QUIET HOURS
48h + 2h	Confirm · Reschedule · Stop	No	Enforced

n8n · Twilio · Anthropic Claude · Postgres on AWS RDS

▷ What going live requires

Everything above runs today on synthetic data, in shadow mode. Turning it on for a real clinic and real patients clears a compliance and integration bar first. Here is that bar, in plain terms.

01 **Carrier registration (A2P 10DLC)**
The clinic's texting brand and campaign get registered with the mobile carriers before a single real reminder goes out. It keeps a legitimate appointment text from being filtered as spam, and ties every message to a named, accountable sender.

02 **A signed BAA, on BAA-covered infrastructure**
A Business Associate Agreement is in place, and the system runs on HIPAA-covered infrastructure under that BAA: an isolated per-clinic tenant by default, or a dedicated instance the practice owns on request. Nothing about a patient touches a service that is not under that agreement.

03 **A one-time consent attestation**
At setup the clinic attests, once, that the numbers on its appointments carry SMS consent. The system records that attestation and re-checks it on every send. A clinic that has not made it cannot send.

04 **A live link to the schedule**
The clinic connects its practice-management system, such as Open Dental, so reminders read real appointment times and write confirmations back. Until that link is live the system sits idle rather than guess at a time or a number.

05 **A shadow period, then one clinic at a time**
Every new clinic starts in shadow mode, logging exactly what it would have sent with zero real messages, until a full appointment cycle has been watched. Only then does the live switch flip, and only for that clinic.

SYSTEM 03

FIELD STUDY · SYNTHETIC DATA

Review collection on autopilot

A post-visit Google review ask that asks fewer people, more carefully: opt-out honored, never the same patient inside six months, one soft follow-up, then it stops. No sentiment routing.

What it replaces

The honest version of this is a front desk that either forgets to ask for reviews or, worse, only remembers to ask the patients who already seemed annoyed. Asking consistently, and only when the moment is right, is more discipline than a busy desk can keep up.

Why it's hard

The temptation is the trick most tools use: ping patients privately first and steer only the happy ones to the public review while diverting the rest to a form. That is review-gating, and Google's policy and the FTC's 2024 rule both ban it. The hard part is the restraint to not do it: ask fewer people, the same way every time, and never filter by who might say something nice.

How it works

01 **Trigger after a visit**
A completed-visit event starts the flow, deduped by visit id.

02 **Gate before asking**
Opt-out and a six-month recency window are checked first, so a STOP is always honored and the same patient is never asked twice in the same season. There is no sentiment check anywhere in the flow.

03 **Send the ask, then stop**
A Google review link goes out through a per-clinic redirect service so it is branded and trackable. One soft follow-up if there is no response, then the system stops.

04 **Count the click, not the patient**
The redirect logs the click against a random token only. No patient identity is stored next to a review.

Under the hood

- The per-clinic redirect service captures real link clicks without storing who clicked.
- No sentiment node anywhere. The gates are opt-out, a six-month recency window, and one soft follow-up, then it stops.
- Multi-tenant and fail-closed: if eligibility cannot be confirmed, no ask goes out.

> It runs on the shared, BAA-covered AWS stack by default with the clinic isolated as its own tenant and the redirect service keyed per clinic, or on a dedicated instance the practice owns on request. The review funnel and its click data stay scoped to that clinic and exportable to it.

Compliance gates

✓ Six-month recency, never the same patient twice	Asks-less, not more
✓ Opt-out (STOP) honored	TCPA §227(b)
✓ Frequency cap shared across systems	TCPA
✓ No PHI in the request or the redirect log	HIPAA §164.502(b)
✓ No sentiment routing, no incentives, no review-gating	Google policy · FTC 2024

When it breaks

- A patient asked in the last six months is skipped, not asked again. Nobody gets pestered, even across several visits.
- The redirect never learns who the patient is. It counts a click against a random token, so the click log on its own ties to no one.
- There is no sentiment step to game. Every eligible patient is asked the same way, so the reviews reflect the work, not a filter.

RECENCY
Never twice in 6 mo

STORED BY A REVIEW
A random token, no name

INCENTIVES
None

REVIEW-GATING
Never, by design

n8n · Twilio · Postgres on AWS RDS · Redirect service

▷ What going live requires

Everything above runs today on synthetic data, in shadow mode. Turning it on for a real clinic and real patients clears a compliance and integration bar first. Here is that bar, in plain terms.

01 **Carrier registration (A2P 10DLC)**
The clinic's texting brand and campaign get registered with the mobile carriers before any real ask goes out. It keeps a legitimate review request from being filtered as spam, and ties every message to a named, accountable sender.

02 **A signed BAA, on BAA-covered infrastructure**
A Business Associate Agreement is in place, and the system runs on HIPAA-covered infrastructure under that BAA (an isolated per-clinic tenant by default, dedicated on request). The ask carries no patient detail, and nothing about a patient touches a service outside that agreement.

03 **A one-time consent attestation**
At setup the clinic attests, once, that a post-visit message is consistent with its patient consent. The system records it, honors opt-outs, and waits out a recency window so nobody is asked twice.

04 **A live link to the schedule**
The clinic connects its practice-management system so the ask only follows a visit that actually happened, and only once it is complete.

05 **A shadow period, then one clinic at a time**
Every new clinic starts in shadow mode, logging the ask it would have sent with zero real messages, until a full cycle has been watched. Only then does the live switch flip, and only for that clinic.

SYSTEM 04

FIELD STUDY · SYNTHETIC DATA

Dormant patient reactivation

Identifies patients overdue for recall (6, 12, 18 months) and runs a personalized reactivation cadence per cohort.

What it replaces

Most practices know a list of lapsed patients exists somewhere in the system. Almost none have the time to pull it, sort it by how overdue each person is, and run a different message for someone three months out versus eighteen. So the list just sits there.

Why it's hard

Reactivation is where it is easiest to become annoying. The same person can be due for recall and already getting a reminder for something else. The hard part is reaching across cohorts without ever contacting someone the other systems just messaged.

How it works

01 Find the lapsed
Once connected to the practice-management system, query patients overdue for recall, bucketed into 6, 12, and 18-month cohorts.

02 Match the cadence
Each cohort gets a different reactivation rhythm. The longer the lapse, the gentler the restart.

03 Draft, then send fully gated
The reactivation touch is drafted in the clinic's voice with Anthropic Claude, from cohort and clinic config only and never a patient identifier, then runs the same safety gates as every other system, with no PHI in the message.

04 Hand bookings back
Confirmed replies route to the schedule, and every touch and outcome is logged.

Under the hood

- Built and shadow-tested against the same multi-tenant, fail-closed spine as the others, in shadow mode against synthetic data.
- Reuses the shared opt-out list and frequency cap so reactivation cannot over-contact someone the other systems already messaged.

> Designed to run where the others do: the shared, BAA-covered AWS stack by default with the clinic isolated as its own tenant, or a dedicated instance the practice owns on request, reading that clinic's own recall data. Nothing about the patient list ever crosses to another clinic.

Compliance gates

✓ Opt-out (STOP) honored before every send	TCPA §227(b)
✓ Frequency cap shared across systems	TCPA
✓ Quiet-hours window	TCPA
✓ No PHI in the message	HIPAA §164.502(b)
✓ If any check errors, the default is do not send	Fail-closed

When it breaks

- The shared frequency cap is the real safeguard. If reminders or missed-call already texted someone, reactivation stands down rather than piling on.
- Cohort logic fails safe. A patient who cannot be cleanly bucketed is left out of the run, never dropped into the most aggressive cadence by accident.

COHORTS	CADENCE	CROSS-SYSTEM CONTACT	PHI IN THE MESSAGE
6 · 12 · 18 mo	Gentler the longer the lapse	Capped and shared	None

n8n · Twilio · Anthropic Claude · Postgres on AWS RDS

▷ What going live requires

Everything above runs today on synthetic data, in shadow mode. Turning it on for a real clinic and real patients clears a compliance and integration bar first. Here is that bar, in plain terms.

01 **Carrier registration (A2P 10DLC)**
The clinic's texting brand and campaign get registered with the mobile carriers before any real outreach. It keeps a legitimate reactivation text from being filtered as spam, and ties every message to a named, accountable sender.

02 **A signed BAA, on BAA-covered infrastructure**
A Business Associate Agreement is in place, and the system runs on HIPAA-covered infrastructure under that BAA: an isolated per-clinic tenant by default, or a dedicated instance the practice owns on request. Nothing about a patient touches a service outside that agreement.

03 **Prior express consent, not just attestation**
Reactivation reaches a patient who has not booked in a while, so it clears a higher bar than a transactional reminder. The clinic confirms prior express consent to be contacted, the system honors every opt-out, and it caps how often anyone is reached.

04 **A live link to the schedule**
The clinic connects its practice-management system so the system can tell who is genuinely overdue, and never contacts someone who already has an upcoming visit on the books.

05 **A shadow period, then one clinic at a time**
Every new clinic starts in shadow mode, logging the outreach it would have sent with zero real messages, until a full batch has been watched. Only then does the live switch flip, and only for that clinic.