$269 MODEL / 3 CHANNELS / 24-7 EXECUTION

The 3-Channel Agentic SDR System

How to run X, cold email, and LinkedIn as one coordinated outbound machine without building a large human SDR team first.

3	1	6-8	157
CHANNELS COORDINATED AS ONE SEQUENCE	CLEAN DATA LAYER BEHIND EVERY TOUCH	WEEKS FOR A SERIOUS ROLLOUT	MODELED CALLS/MO ON CONSERVATIVE MATH

WHAT THIS GUIDE GIVES YOU

Not another “use AI for outbound” post.

This guide shows the actual operating model behind a lean outbound machine: the sequence, the stack, the data structure, the judgment layer, and the rollout logic behind it.

- What each channel is responsible for
- Why the database comes before the messages
- What the agent should and should not automate
- How to keep the system useful instead of dangerous

Modeled economics, not guaranteed output. Results still depend on the offer, targeting, list quality, domains, and execution discipline behind the system.

SEQUENCE

The 3-channel rhythm.

DAY 1	X DM creates early relevance.
DAY 3	Email carries the clearest problem diagnosis.
DAY 5	LinkedIn adds identity and context.
DAY 7-14	Follow-ups add value instead of repeating the ask.

SYSTEM LAYERS

Structure before automation.

One orchestrator. One database. One command layer. Channel-specific executors. That is what keeps the machine useful.

SIGNALS · PROBLEM MAP · SEQUENCE · LOGGING · ESCALATION

Want this built for your company? Email **ghiles@muditek.com** if you want the stack, data model, and outbound playbook adapted to your market.

The bottleneck is rarely effort. It is coordination.

Traditional outbound breaks when volume rises. People miss follow-ups, drift off process, improvise messaging, and stop logging cleanly. The system below fixes that by turning three channels into one sequence with one memory.

WHY HUMAN-ONLY OUTBOUND STALLS

Output falls apart before the pipeline math does.

- Follow-ups get skipped when calendars get full.
- Messaging drifts the moment reps improvise.
- Logging gets messy, so the team stops learning.
- Channels run as separate campaigns instead of one system.

THE REAL GOAL

Create controlled familiarity across channels, not noise. By the time the prospect opens the email, they should already recognize the name.

DAY 1

X DM

Light first touch. Short, relevant, and intentionally low pressure.

DAY 3

Cold Email

Main commercial message. Name the problem clearly and make a useful, low-friction offer.

DAY 5

LinkedIn

Identity layer. Gives context, credibility, and a peer-to-peer place to inspect who you are.

DAY 7-14

Progressive Follow-up

Each follow-up adds something new: sharper diagnosis, clearer mechanism, example, framework, or proof.

WHAT THE PROSPECT SHOULD FEEL

Relevant. Then credible. Then familiar. Then worth replying to. That is the sequence.

Do not paste the same message everywhere.

A good multi-channel system does not repeat one pitch three times. It assigns a different job to each channel and keeps each one in character.

X / ATTENTION ARBITRAGE

Fast. Brief. Early.

X is still one of the cleanest places to create an early touch that feels light instead of tired.

- Create relevance, not a meeting request.
- Do not dump links or case studies.
- Its only job is to make the next touch feel less cold.

EMAIL / COMMERCIAL LOGIC

The real offer lives here.

Email wins when it names a real problem, points to a believable mechanism, and asks for a small next step.

- Problem first.
- Believable mechanism second.
- Small offer, not a forced meeting.

LINKEDIN / IDENTITY LAYER

Give the prospect context.

LinkedIn answers a basic question: is this a real operator in my world, or just another random sender?

- Keep the tone more peer-to-peer than email.
- Use it to reinforce relevance, not restart the pitch.
- Let the profile do some of the proof work.

FOLLOW-UP / COMPOUNDING VALUE

Progress or stop.

If the follow-up says the same thing again, the system is not following up. It is just repeating itself.

- Add a sharper angle.
- Add a framework or useful observation.
- Add compact proof when it helps.

SMALL OFFERS THAT WORK

Give them something easy to say yes to.

- short teardown
- problem breakdown
- setup guide
- decision framework
- compact diagnostic

WHAT WEAK OUTREACH SOUNDS LIKE

Do not make the first touch heavier than it needs to be.

- book a call so I can explain
- generic compliment as the opener
- long case study dump
- same pitch pasted across every channel
- follow-up that just repeats the ask

The stack is cheap. The system behind it is the real asset.

The vendor list is not the moat. The moat is one environment, one orchestrator, one command surface, one clean data layer, and channel-specific executors that behave predictably.

Hostinger VPS
ENVIRONMENT

Always-on machine for the orchestration layer. Cheap enough to stay lean, useful enough to stay consistent.

Tailscale
PRIVATE ACCESS

Basic hardening and controlled access. You do not want this system hanging off personal sessions and loose credentials.

OpenClaw
ORCHESTRATION

The central workflow layer. The exact product can change. The need for a central orchestrator does not.

Telegram Bot
COMMAND LAYER

Operational surface for commands, checks, alerts, and simple monitoring without living inside the server all day.

Instantly
COLD EMAIL

Sending infrastructure for the clearest commercial message in the sequence.

PhantomBuster
LINKEDIN EXECUTOR

LinkedIn actions inside a broader sequence. Useful when LinkedIn supports the logic instead of becoming the whole campaign.

Drippi
X SUPPORT

X scraping and outreach support around the channel with the least template fatigue and some of the best attention economics.

Apollo
ENRICHMENT

Contacts, enrichment, and the data needed to keep the sequence working from one clean record.

Shared Database
SYSTEM MEMORY

Without one system of record, you do not have an SDR machine. You have disconnected tools pretending to be one.

ARCHITECTURE IN ONE LINE

One environment. One orchestrator. One command surface. One clean data layer.

Signals
triggers, enrichment, selection logic

Problem Map
what likely issue each signal implies

Sequence
X, email, LinkedIn, follow-ups

Logging
touches, replies, status changes, escalation

Human Handoff
pricing, intent, objections, high-value replies

Most outbound systems are broken before the first message goes out.

The main failure point is usually not the model. It is bad memory, bad structure, and weak documentation. That is why the data layer and the skill layer have to exist before the system touches prospects.

> An agent without documentation is just automated randomness.

WHAT THE DATABASE SHOULD KNOW

One clean system of record.

- Who the prospect is
- Why they were selected
- What signal triggered the outreach
- Which angle is active
- Which channel already touched them
- What happened next
- Whether a human should step in

MINIMUM DOCUMENTATION STACK

The judgment layer behind the model.

- **ICP and exclusions** so the system knows who to target and who to avoid.
- **Signal-to-problem map** so the touch starts from a believable problem, not a compliment.
- **Channel rules** so each platform behaves differently.
- **Offer library** so the system can make small useful offers instead of forcing a meeting.
- **Escalation rules** so high-value replies and real objections reach a human fast.
- **Logging rules** so the system can actually learn.

THE REAL JOB OF AI HERE

Read the signal. Map it to a likely problem. Select the right message pattern. Send or queue the right touch. Log what happened. Escalate exceptions. That is enough. The model is not your GTM brain.

The message system needs rules. The security layer needs rules too.

Most people automate the wrong layer and leave the dangerous parts undocumented. These are the rules that keep the machine sharp enough to work and restrained enough not to become a liability.

01

Lead with the problem

The prospect cares more about whether you understand the problem than whether you read their profile carefully.

02

Make the offer small

Use teardown, setup, framework, or diagnostic language. Do not force the meeting before you have earned attention.

03

Keep each channel in character

X should feel fast. Email should carry the clearest logic. LinkedIn should feel like a real peer exchange.

04

Make follow-ups progressive

Every follow-up needs new value. Sharper angle. Short framework. Useful observation. Proof. Not repetition.

05

Stop scaling weak messages

If the offer is vague or the list is weak, automation just scales failure faster.

06

Escalate when judgment matters

Pricing, legal, strong buying intent, and high-value objections should trigger a human handoff fast.

BAD AUTOMATION

What to avoid

- Fake personalization
- Compliment-led openers
- One message copied to every channel
- “Just checking in” follow-ups
- Running from personal browser sessions

HARDENING BASICS

What needs to be true

- Secrets stay outside the workspace
- Access is controlled and minimal
- Dedicated accounts where possible
- Model fallbacks configured
- Heartbeat and cron checks verified

Build the operating logic first. Then earn the right to scale.

A serious rollout is usually a 6–8 week project, not a weekend install. The core software cost can be lean. The real cost is weak judgment, weak list quality, and vague offers sent at scale.

$269	4,140	157	6–8
APPROXIMATE LEAN CORE STACK / MO	MODELED DAILY TOUCHES ACROSS CHANNELS	CONSERVATIVE MODELED CALLS / MO	WEEKS FOR A PROPER ROLLOUT

01

Strategy

Define the ICP, exclusions, problem map, offer library, and logging structure before tools touch live prospects.

02

Infrastructure

Provision the environment, secure access, configure the orchestrator, and connect the channel tools.

03

Message System

Write the channel rules, follow-up rules, escalation rules, and test them on a small list first.

04

Hardening

Move secrets, restrict access, verify fallbacks, test failure paths, and make sure monitoring is real.

05

Controlled Launch

One audience, one offer, one sequence, low volume. Scale only after the data is clean enough to trust.

06

Optimization

Tighten the message, cut dead segments, improve the problem map, and keep the system learning instead of just sending.

IMPORTANT

The exact numbers are not the point. The point is that once the system is documented properly, the economics start to behave like software instead of headcount. If the offer or list is weak, the same stack will simply scale irrelevance faster.

This works best when the strategy already exists.

The system is not a substitute for a clear offer, a sharp ICP, or disciplined messaging. It is an execution layer for teams that already know what they are trying to say and to whom.

BUILD THIS IF

You already have signal.

- You can describe the ICP clearly.
- You can name the buyer problem sharply.
- You have a believable mechanism behind the offer.
- You are willing to document the judgment layer.
- You want repeatability, not just more activity.

DO NOT BUILD THIS IF

You are still guessing.

- The offer is still vague.
- The list quality is weak.
- You cannot explain the mechanism clearly.
- You are hoping AI will replace strategy.
- You are not ready to track what the system is doing.

Want this set up for your company?

If you want the stack, the data model, and the outbound playbook adapted to your market, reach out. I can help you design the sequence, structure the memory layer, and tighten the offer before anything gets automated.

ghiles@muditek.com

WHAT I CAN HELP WITH

Practical setup, not theory.

- Sequence design
- Data model and logging
- Skills and SOP definition
- Offer and message tightening
- Controlled rollout planning