OPERATING HUB TEMPLATE

Technical Implementation Guide

How to actually build each offer — companion to the Delivery Plan

July 19, 2026 · For the Build Team, Implementation Pod, and Outbound Pod

How to use this guide

The Delivery Plan answers who owns what and when it ships. This document answers how each build actually gets done — the concrete technical steps, in order, for whoever is sitting down to build it.

Two offers (Reactivation, and the Acquisity-based offers) already have a working reference build — those sections describe the real, tested architecture. The rest are build specs written from the offer description and standard practice for each tool; treat them as a strong first draft and adjust once you're inside the actual accounts.

When to start each build

• Nothing in this guide starts until the offer is sold — contract signed and first payment collected, matching the funnel's “Contract signed · cash collected” stage. Don't scope, build, or request client access speculatively ahead of that.

• This is per offer, not per client. An existing client buying a second offer (e.g., Reactivation client later buys AI Agents) still starts a fresh Phase 1 kickoff for that offer on its own timeline — it doesn't ride on the first offer's build.

• The one exception is the reference builds already in hand (Reactivation, and the demo assets for the rest) — those exist to sell the offer, not as pre-built client instances. Nothing client-specific gets touched until that client's contract is signed.

Before you start any build

• Confirm you have the access listed in that offer's “Access needed” table — most build delays come from waiting on a login, not the build itself.

• Every build ends with the QA step in that section before it touches a live client list, ad account, or phone number.

Shared architecture pattern

Reactivation, AI SDR, AI Receptionist, and AI Agents all reuse the same core pattern. Building Reactivation first and treating it as the template is the fastest path through the other three.

The pattern

1. Trigger — schedule, webhook, or inbound event starts the flow.
2. Fetch / score — pull records from the CRM or source list, rank or route them.
3. Personalize — Claude drafts or the voice agent improvises using the specific record's context (deal size, history, intent).
4. Act on channel — Vapi (voice), SendGrid/Acquisity (email), or Twilio (SMS) sends the actual touch.
5. Log — write the outcome back to the CRM.
6. Notify — post to Slack when a human needs to act.

n8n is the orchestration layer for the pattern above in every offer except the two Acquisity offers, where Acquisity's own sequence engine replaces the n8n “act on channel” step for email.

OFFER 02

Reactivation

The reference build — this is the real, working architecture.

Access needed

Tool / Access	What you need	Who provides it
HubSpot	Private app token, scopes: crm.objects.contacts (read/write), crm.objects.deals (read), timeline events (write)	Client
n8n	Workspace + credentials configured for HubSpot, Vapi, SendGrid, Twilio, Slack	Build Team
Vapi	Account, phone number provisioned, an assistant configured with Claude Sonnet 5 as the model	Build Team
SendGrid	Account with sending domain authenticated (SPF/DKIM)	Build Team / Client domain
Twilio	A2P 10DLC campaign registered, phone number	Client (brand registration) + Build Team (setup)
Slack	Incoming webhook or bot token for the target channel	Client

Build steps

1. Extract & score

1. Schedule node: trigger daily at 9:00 AM.
2. HubSpot node: search contacts where last activity > 90 days AND has an associated deal.
3. Code node: score each lead 0–100 — weight deal value (higher value = higher score) and recency (more recently active within the dormant window = higher score). Example: score = (deal_value_percentile × 0.6) + (recency_percentile × 0.4) × 100.

2. Channel routing

1. Switch/Router node on the score field: ≥ 60 → Voice branch, 30–59 → Email branch, < 30 or explicit SMS preference → SMS branch.

3. Multi-channel outreach

1. Voice branch — HTTP Request node to Vapi's call API, passing dynamic variables (contact name, deal value, dormancy days, last topic) into the assistant's system prompt so the agent opens with real context.
2. Email branch — Claude node drafts a short, personalized email using the same variables; SendGrid node sends it.
3. SMS branch — Twilio node sends a short, human-sounding text from the registered number.

4. Log & notify

1. HubSpot node: write a timeline event / activity log for whichever channel fired.
2. IF node: when a reply or high-intent signal comes back (webhook from Vapi/SendGrid/Twilio), post to the Slack channel with the AI-generated summary and a CRM deep link.

Template — Vapi system prompt (voice branch)

```
You are Sarah, calling on behalf of {{company_name}}.
Context: {{contact_name}} discussed a ${{deal_value}} project {{dormancy_days}} days
ago
that went quiet. Do not read a script — reference the specific context above.
Goal: confirm whether it's still relevant and book a 10-minute catch-up if so.
If they say no or ask to be removed, thank them and end the call — do not push.
```

Template — Claude email-draft prompt (email branch)

```
Draft a short (under 80 words) reactivation email from {{sender_name}} to
{{contact_name}}.
Context: their ${{deal_value}} deal went quiet {{dormancy_days}} days ago.
Tone: warm, direct, no pressure. One question only. No generic filler.
```

QA checklist

- Run one batch against a small test segment (5–10 leads) before pointing at the full dormant list.
- Confirm all three channels log correctly to HubSpot.
- Confirm the Slack alert fires only on genuine high-intent replies, not every send.
- Have a teammate call/text/reply as a fake prospect to confirm the reply-handling path works end to end.

OFFER 01 & 07

AI Growth Engine + AI SDR (Acquisity)

Same underlying engine — built once, packaged twice.

Build this once

• Since both offers run on Acquisity's cold-email engine, build the sequence infrastructure once per client and simply set a different volume/goal (Growth Engine = full funnel to close; AI SDR = standalone 30-call target).

Access needed

Tool / Access	What you need	Who provides it
Acquisity	Workspace access for the client's account, sequence builder, reply-detection settings	Whoever administers Acquisity internally
Sending domain(s)	DNS access to publish SPF, DKIM, and DMARC records; ideally 2–3 warm-up domains separate from the client's primary domain	Client (DNS) + Build Team (records)
Target list	Cleaned list matching the client's ICP, or access to an enrichment source	Client or Outbound Pod
Calendar	Write access so the AI SDR can create events directly, not just send a link	Client
CRM	Where booked calls and replies get logged	Client

Build steps

1. Deliverability foundation

1. Publish SPF, DKIM, and DMARC records for every sending domain before any send goes out.
2. Warm up each domain on a ramping schedule (low volume for the first 1–2 weeks, increasing gradually) — confirm Acquisity's built-in warm-up feature is switched on if it has one; otherwise ramp manually.

Template — DNS records (example, replace with real values)

```
TXT  @               "v=spf1 include:_spf.acquisity.com ~all"
TXT  acq._domainkey  "v=DKIM1; k=rsa; p=<public key from Acquisity>"
TXT  _dmarc          "v=DMARC1; p=quarantine; rua=mailto:dmarc@{{client_domain}}"
```

2. List & sequencing

1. Load the target list into Acquisity, mapped to the ICP fields the sequence personalizes against (name, company, role, niche-specific detail).
2. Build the sequence: opening email → 1–2 follow-ups spaced several days apart — write copy oriented around the specific pain point for the niche, not a generic pitch.

Template — 3-touch sequence structure

Touch	Timing	Purpose
1	Day 0	Specific, low-pressure opener referencing the prospect's business
2	Day 3–4	Add a proof point (result, case study) — no “just checking in”
3	Day 7–8	Direct ask with an easy yes/no close (“worth 15 minutes?”)

3. Reply handling → auto-booking

1. Configure Acquisity's reply-detection (or connect it via webhook to n8n if Acquisity doesn't natively branch on reply sentiment) so a positive reply triggers the booking flow rather than just notifying a human.
2. Connect the calendar so the booking step creates the event directly on the client's calendar as “Discovery Call,” with the prospect's context in the event description.
3. Route the same reply event to the CRM (log) and Slack (notify) the same way the Reactivation build does.

4. Packaging split (01 vs. 07)

1. For AI Growth Engine: connect the output of this engine into the existing discovery → growth plan → close call sequence.
2. For AI SDR: track and report against the 30-calls-in-30-days target as a standalone metric, independent of the close-call pipeline.

QA checklist

- Send a small test batch (10–20 contacts) and confirm deliverability — check inbox placement, not just “sent” status.
- Send yourself a test reply and confirm it books correctly on the calendar with the right details.
- Confirm CRM logging and Slack notification both fire on a real reply.

OFFER 08

AI Receptionist

Same voice-agent stack as Reactivation, configured for inbound.

Access needed

Tool / Access	What you need	Who provides it
Business phone number	Porting to Vapi, or call-forwarding configured from the existing number	Client
Vapi	Assistant configured for inbound with a qualification + booking script	Build Team
Calendar	API/webhook access so the assistant can book directly	Client
CRM	Webhook or API access to log every call	Client

Build steps

1. Map the client's 5–10 most common inbound scenarios (emergency vs. routine, service types, service area) before writing the script — this is the input the assistant's prompt is built from.
2. Configure the Vapi assistant: greeting, qualifying questions, and a function-calling step that checks calendar availability and books the appointment.
3. Connect a webhook from Vapi (call-ended event) into n8n → CRM node to log every call, qualified or not.
4. Set up call forwarding or number porting so inbound calls route to the assistant.

Template — Vapi system prompt (inbound)

```
You are the receptionist for {{business_name}}. Answer warmly, in one sentence,
then ask what they need help with.
Qualify: service type, urgency (same-day vs. routine), and service area.
If urgent and within service area: check calendar availability via the booking
function and offer the next 2 open slots. Confirm the booking back to the caller.
If outside scope or service area: say so plainly and offer to take a message.
```

QA checklist

- Call the line yourself with 3–4 different scenarios (routine booking, emergency, question the assistant shouldn't answer) and confirm it handles each appropriately.
- Confirm a booked call actually appears on the calendar with correct time and details.
- Confirm the CRM log captures the call even when no booking happens.

OFFER 09

AI Agents

The workflow-mapping session determines the actual build list — this is the process for running it.

Access needed

Tool / Access	What you need	Who provides it
Client stakeholder	1–2 hours for a workflow-mapping session	Client
Per-agent system access	Varies by agent — CRM, inbox, Slack, scheduling tool, etc.	Client, confirmed per agent once scoped

Build steps

1. Workflow-mapping session

1. Walk the client through their day-to-day operations by function: sales, ops, support, back-office.
2. For each recurring task, capture: trigger, current manual steps, tools touched, and time spent — this becomes the shortlist.
3. Prioritize the 10 agents by a simple score: time saved × frequency, weighted down for anything needing access the client can't grant quickly.

Template — interview questions to run the session

- “Walk me through what happens the moment a new lead/ticket/invoice comes in.”
- “What do you do by hand today that you wish just... happened?”
- “Which of these tasks would you trust an AI to do without checking its work first?”
- “What's the cost of this going wrong once?” — this flags anything too high-stakes for an early agent.

2. Build in priority order

1. For each agent, reuse the shared architecture pattern (trigger → fetch/score → personalize via Claude → act → log → notify) rather than designing each from scratch.
2. Build shared components once — CRM connectors, Slack notifier, Claude prompt scaffolding — and reuse them across agents to cut build time.

3. QA & handover per agent

1. Test each agent against 3–5 real scenarios pulled from the client's own data before calling it done.
2. Write a one-page handover doc per agent: what it does, what triggers it, how to pause it, who to contact if it misfires.

QA checklist

- Each agent tested against real (not hypothetical) scenarios before going live.
- A named approver signs off on each agent before it starts acting on real data.

OFFER 03

Looking to Exit

An advisory methodology, not a software build.

What you need

- Recent P&L / financial statements (12–24 months if available)
- Cap table or ownership structure
- Advisory Lead assigned for this engagement

Method

1. Unit-economics pass

Calculate these four numbers from the financials provided — they're the backbone of everything downstream:

```
Contribution margin  = Revenue − Variable costs (per job/unit/client)
CAC                  = Total sales + marketing spend ÷ new customers acquired
LTV                  = (Avg. monthly revenue per customer × Gross margin %) ÷ monthly
churn rate
Payback period       = CAC ÷ (Avg. monthly revenue per customer × Gross margin %)
```

2. EBITDA bridge

Identify 3–5 levers and quantify each in a simple bridge table so the client can see exactly where margin comes from:

Lever	Current	Target	$ EBITDA impact
Pricing	e.g. avg. job $420	e.g. $460 (+9.5%)	+$X
Cost of delivery	e.g. 62% COGS	e.g. 56% COGS	+$X
Overhead	e.g. 18% of revenue	e.g. 15% of revenue	+$X

3. Valuation benchmarking

Pull comparable EV/EBITDA multiples for the client's sector and size band from recent transaction data (industry M&A reports, sector-specific brokers). Treat any multiple you don't have a real comp for as illustrative only — don't let a placeholder multiple end up in a client-facing report.

4. Exit-readiness report

Structure the written deliverable in this order so it reads as a decision document, not a data dump:

- Executive summary — current valuation range and the single biggest lever
- Current-state financials — the four unit-economics numbers above
- EBITDA bridge — the lever table, with a 90-day and 12-month view
- Valuation range — current vs. post-improvement, with the comps used shown
- 90-day action plan — the first 3 moves, owned and dated

QA

- Have a second reviewer sanity-check the EBITDA assumptions before the client sees them — these numbers carry real weight in a sale conversation.
- Never present a valuation multiple that isn't backed by an actual comparable — flag it as a range with a stated source instead.

OFFER 04

Fractional Sales

A training engagement, not a staffing one — the client already has a VA; we teach them to run it.

What you need

- CRM/pipeline export and current close-rate baseline
- Call recordings routed through Acquisity's Sales Call Analyser
- The client's VA's time for training sessions and the first few co-piloted calls

Method

1. Pipeline audit

Pull win/loss data by stage to find where deals actually die — a stage-by-stage drop-off table, not just an overall close rate.

2. Call review — objection taxonomy (auto-tagged by the Sales Call Analyser)

Acquisity's Sales Call Analyser auto-transcribes and tags every recorded call into one of five buckets, so the playbook targets what's actually happening, not assumed objections:

- Price — “too expensive,” “need to check budget”
- Timing — “not right now,” “check back next quarter”
- Authority — “need to run this by my partner”
- Trust — “how do I know this will work,” unfamiliarity with the company
- Competitor — actively comparing or already using an alternative

3. Call-scoring rubric (also produced by the Analyser)

The Analyser scores every call 1–5 on each dimension automatically — this is what the client's VA is taught to read and act on:

Dimension	1 (weak)	5 (strong)
Opening	Generic, no context	References the prospect's specific situation
Discovery	Pitches immediately	Uncovers the real problem before pitching
Objection handling	Argues or concedes	Reframes and re-asks for the close
Close ask	Vague or absent	Direct, specific next step with a date

4. Build the playbook, then teach it

Build the objection-handling playbook and close script directly from the Analyser's tagged patterns, not a generic template. Then run training sessions with the client's VA: walk the playbook, role-play the top objections, show them how to read their own scorecard. Co-pilot 2–3 live calls, coaching in real time, before stepping back fully.

5. Weekly tracking

Track every week against the pre-engagement baseline using the same four fields so 30-day lift is a clean comparison:

Week	Calls taken	Close rate	Top objection this week
Baseline	—	18%	—
Week 1			
Week 2			
Week 3			
Week 4			

QA

- Confirm the client's VA can read their own scorecard unassisted before calling training complete.
- Confirm the baseline close-rate number is calculated the same way before and after (same pipeline stages counted) so the +20% comparison is real.

OFFER 05

DFY SEO

Technical + content build, long-lead by nature.

Access needed

Tool / Access	What you need	Who provides it
Website / CMS	Editor or admin access	Client
Google Search Console	Owner or full access, verified property	Client
Google Analytics	View/edit access	Client

Build steps

1. Technical audit — specific checks

- Indexing: every priority page returns a valid 200, is in the sitemap, and isn't accidentally no-indexed.
- Core Web Vitals: LCP under 2.5s, CLS under 0.1, INP under 200ms — fix whatever's failing before content work starts.
- Meta titles under 60 characters, meta descriptions under 155 characters, one H1 per page.
- Schema markup matching the business type — LocalBusiness + Service at minimum; add FAQPage schema on pages with genuine Q&A content.

2. Keyword & content — brief template

Use one brief per page so writers (human or AI-assisted) aren't guessing at scope:

Field	Example
Target keyword	“emergency plumber [city]”
Search intent	Local, high-urgency, transactional
Word count target	900–1,200
Required sections	Service overview, service area, pricing signal, trust signals, CTA
Internal links	Link to 2–3 related service pages

3. Authority building

- NAP (name/address/phone) consistency across the top 20 local directories before anything else — inconsistent NAP undercuts local rankings regardless of content quality.
- Target a steady monthly citation/backlink cadence rather than a single burst — local citations first for local-service clients, broader link building after.

4. Tracking

- Set up rank tracking against the priority keyword list (Search Console API or a third-party rank tracker) and report monthly.

QA

- Confirm Search Console shows the target pages as indexed within 1–2 weeks of publishing.
- Set client expectations explicitly at 90–180 days before meaningful rank movement — do not imply a 30-day result.

OFFER 06

Creative Ad Agency

30 tested ads in 30 days — a production + testing pipeline.

Access needed

Tool / Access	What you need	Who provides it
Meta Ads Manager	Access to the live client ad account	Client
Brand assets	Logo, brand guidelines, any existing footage/photos	Client

Build steps

1. Baseline + hook/angle matrix

Pull the current CPL by existing creative, then build a matrix of angles × formats so production isn't ad-libbed:

Angle	Format	Example hook
Pain point	Static	“Still chasing quotes by hand?”
Before/after	UGC clip	Job transformation, 15–30s
Testimonial	UGC clip	Owner or customer, unscripted tone
Urgency	Static	“Booked out 3 weeks — here's why”

2. Copy formula

Use a consistent structure per piece so the test isolates the hook, not the whole ad's coherence: Hook (pattern interrupt) → Problem → Agitate → Solution → CTA.

3. Test budget

Spend at least 2–3× the target CPL per creative before judging it — e.g. a $30 target CPL needs $60–90 in spend, or roughly 50+ clicks / 3–5 leads, before comparing performance. Judging on day one is comparing noise.

4. Production + launch

- Produce the first batch (8–10 pieces) and get brand/compliance sign-off before spend.
- Launch in small test batches inside Meta Ads Manager with a fixed test budget per creative so results are comparable.
- Apply the kill-criteria rule above and reallocate budget to winners.
- Repeat production in batches until 30 variations have been tested across the 30-day window.

QA

- Every piece of creative is checked against brand guidelines and ad-platform policy before it goes live.
- No creative gets killed before it reaches the minimum spend/lead threshold above.

Master access checklist

Pulled together from every offer above — useful as a single list to send the client before kickoff.

- HubSpot (or client CRM) — private app / API token
- n8n workspace — provisioned with credentials for each tool below
- Vapi — account + phone number(s) for outbound and/or inbound
- SendGrid — account + authenticated sending domain
- Twilio — A2P 10DLC registered number
- Slack — webhook or bot token for the target channel
- Acquisity — workspace access, sending domain(s), DNS access for SPF/DKIM/DMARC
- Calendar — write access for automated booking (not just a shareable link)
- Meta Ads Manager — access to the live ad account
- Google Search Console + Analytics — verified property access
- Business phone number — porting or forwarding access for the AI Receptionist