

## CHECKLIST

# Brokerage CRM Integration Planning Checklist

Plan the CRM connection for AI calling at a brokerage: owners, field mapping, dispositions, triggers, sync direction, webhooks, testing and data access.

### WHO IT IS FOR

Broker ops leads, CRM admins and retention managers connecting an AI voice agent to HubSpot, Salesforce, Pipedrive or a custom CRM.

### TIME TO COMPLETE

90 minutes across two sessions

This checklist takes you from "we want the AI agent to update the CRM" to a written plan your CRM admin, your ops lead and your calling platform can build against. Work through it before anyone opens HubSpot or Salesforce settings or writes a webhook handler. At the end you'll have every field mapped, every call outcome tied to a CRM status, a trigger list, a sync-direction rule per field and a test plan with pass criteria. The build itself gets short when the plan is done; most failed CRM integrations skipped this part.

## 1 Scope and owners

Decide what the integration is for before deciding how it works. One use case first, then extend.

- ☐ **Name the first campaign the integration serves: dormant trader reactivation, deposit follow-up, KYC reminder or registration follow-up.**

Each one needs different fields and different outcomes. Planning for all four at once produces a mapping nobody finishes.

- ☐ **Write down which CRM is the system of record for contacts, and which system holds trading data (MT4/MT5, back office).**

Most brokerages have both, and the calling platform needs a bit of each. Naming the owner of each fact stops two systems from disagreeing about the same trader.

- ☐ **Assign one owner on the CRM side and one on the calling side, with a decision-maker for conflicts.**

Field-name arguments stall integrations for weeks. A named person settles them in a day.

**CRM owner:** \_\_\_\_\_ **Calling owner:** \_\_\_\_\_

- ☐ **Confirm the integration path: native CRM connector, webhooks into your own middleware, the Integrations catalog of no-code flows, or direct REST API.**

Topcalls connects HubSpot, Salesforce and Pipedrive out of the box, fires webhooks on call events, and exposes a REST API for campaigns, contacts and call data. Pick the path your team can maintain, not the most powerful one.

- ☐ **List every system that must see call results beyond the CRM: dealing desk dashboard, retention team inbox, compliance archive, email tool.**

Each extra consumer is a webhook subscriber or a CRM workflow. Knowing the count now keeps the outcome payload complete.

- ☐ **Set the go-live definition: which campaign, how many contacts, and what "working" means in one sentence.**

Example: every completed call on the dormant list writes a disposition and a transcript link to the contact within a minute.

**Working means:** \_\_\_\_\_

## 2 Field mapping

Two lists: what the agent needs before the call, and what the CRM needs after it. Map each field once, in a shared sheet, with the CRM's internal field name rather than its label.

- ☐ **List the fields the agent reads before dialing: first name, phone in E.164, language, account status, last activity date, balance band, lead source, assigned account manager.**

The agent references these on the call. A missing language field means a call in the wrong language; a missing status means a generic script.

- ☐ **Record the CRM internal field name and type for each one, and flag any field that lives outside the CRM.**

"Last trade date" usually sits in MT4/MT5, not in HubSpot. Decide whether it syncs into the CRM first or the calling platform reads it from a separate export.

- ☐ **Pick the join key that ties a call back to a contact: CRM record ID, MT4/MT5 login or both.**

Never join on phone number or email alone. Shared family numbers and changed emails break the link, and outcomes land on the wrong trader.

- ☐ **List the fields the CRM receives after each call: disposition, call timestamp, duration, callback time, recording link, transcript link, summary, do-not-call flag.**

Create dedicated fields for these. Overwriting an existing status field with call outcomes destroys the sales pipeline view.

- ☐ **Decide which post-call fields are single-value (latest call wins) and which are appended (every call keeps a record).**

Disposition is usually latest-wins on the contact plus a full history as a call activity. Do-not-call is write-once and never reverts.

- ☐ **Define the allowed values for every picklist field on both sides, and confirm they match character for character.**

A disposition of "Callback" sent to a CRM field that expects "Call back" fails silently or creates a junk value.

- ☐ **Mark any field that holds personal data you don't want the calling platform to receive, and exclude it from the outbound mapping.**

The agent needs a first name and a status, not the trader's full address or document numbers. Send the minimum.

- ☐ **Fill the mapping table below for your own fields before the build starts.**

Fields mapped: \_\_\_\_\_ Fields left out: \_\_\_\_\_

| DATA POINT      | DIRECTION                | CRM FIELD (INTERNAL NAME) | CALLING PLATFORM FIELD | RULE                                        |
|-----------------|--------------------------|---------------------------|------------------------|---------------------------------------------|
| First name      | CRM to agent             | firstname                 | first_name             | Required; hold record if blank              |
| Phone           | CRM to agent             | phone                     | phone_number           | E.164 only; reject otherwise                |
| Language        | CRM to agent             | preferred_language        | language               | Fall back to country default                |
| Account status  | CRM to agent             | account_status            | custom.status          | Picklist; values matched exactly            |
| Last activity   | Platform export to agent | n/a (MT4/MT5)             | custom.last_activity   | Refreshed at list import                    |
| Disposition     | Agent to CRM             | ai_call_disposition       | outcome                | Latest wins on contact; history as activity |
| Callback time   | Agent to CRM             | ai_callback_at            | callback_at            | Creates a task for the account manager      |
| Do-not-call     | Agent to CRM             | do_not_call               | dnc_requested          | Write once; never revert                    |
| Transcript link | Agent to CRM             | ai_call_transcript_url    | transcript_url         | Appended per call                           |
| Your field      | _____                    | _____                     | _____                  | _____                                       |

### 3 Dispositions and CRM outcomes

A disposition is the one word the CRM stores about what happened on the call. Keep the list short, define each value, and decide what the CRM does with it.

- ☐ **Write the disposition list with no more than ten values, each with a one-line definition the retention team agrees on.**

Interested, callback requested, deposit intent, KYC will complete, not interested, wrong number, no answer, voicemail, do not call, handed to human. Add a value only when a team member would act differently on it.

- ☐ **Map each disposition to a CRM lifecycle stage or pipeline stage change, or explicitly to "no stage change".**

Interested moves the contact to a retention pipeline. No answer changes nothing except the attempt counter. Write both down.

- ☐ **Map each disposition to the follow-up it creates: task for the account manager, email sequence, SMS, next call attempt or nothing.**

Deposit intent without a task in the account manager's queue within minutes is a lost deposit. Put the automation on the CRM side where the team already works.

- ☐ **Decide what the agent captures beyond the disposition: reason for leaving, objection raised, deposit amount mentioned, preferred callback window.**

These land in a structured summary field, not in free text a workflow can't read. Agree the field names in the mapping sheet.

- ☐ **Set the rule for attempt counters and retry limits per contact, and where they live.**

If the CRM owns the counter, the calling platform must read it before dialing. If the calling platform owns it, the CRM only needs the final outcome.

- ☐ **Confirm who reviews unmapped or ambiguous outcomes each day and how they get corrected.**

An AI agent will produce a handful of calls that fit no disposition. Someone reads them, and the list gets a new value or a clearer definition.

| DISPOSITION           | CRM STAGE CHANGE                       | FOLLOW-UP CREATED                              | OWNER            |
|-----------------------|----------------------------------------|------------------------------------------------|------------------|
| Interested            | Move to retention pipeline             | Task for account manager, same day             | Retention        |
| Callback requested    | No change                              | Scheduled call task at requested time          | Retention        |
| Deposit intent        | Move to deposit pending                | Task plus deposit-link email                   | Retention        |
| KYC will complete     | Move to KYC pending                    | KYC reminder email; recheck in 48 hours        | Onboarding       |
| Not interested        | Mark reactivation declined             | Suppress from this campaign; review in 90 days | Ops              |
| No answer / voicemail | No change                              | Next attempt per cadence                       | Calling platform |
| Wrong number          | Flag phone invalid                     | Data fix task                                  | CRM admin        |
| Do not call           | Set do-not-call, remove from all lists | None; log the request                          | Compliance       |
| Handed to human       | No change                              | Live transfer or immediate callback task       | Retention        |

## 4 Triggers: what starts a call

Decide what event puts a contact into a campaign, and what takes them out. Both directions matter.

- ☐ **List every trigger that enqueues a contact: list import, CRM stage change, inactivity threshold reached, deposit page abandoned, KYC document rejected, form submitted.**

Batch triggers (a list) and event triggers (a stage change) usually run side by side. Name which campaign each one feeds.

- ☐ **Define the delay between the trigger and the call for each event trigger.**

A registration follow-up may call within minutes. A dormancy trigger may wait for a compliance window. Put the number in the plan, not in someone's head.

Trigger: \_\_\_\_\_ Delay: \_\_\_\_\_ min

- ☐ **Define the exit conditions that pull a contact out of a campaign before the next attempt: deposit made, KYC passed, opt-out received, account closed, complaint opened.**

The CRM or back office knows these first. Without an exit sync, the agent congratulates a trader on reactivating after they already deposited, or worse, calls someone who opted out yesterday.

- ☐ **Confirm the calling platform checks the do-not-call flag and the exit conditions at dial time as well as at list import.**

A list imported Monday and dialed Thursday is three days stale. Dial-time checks close the gap.

- ☐ **Set the maximum number of campaigns a contact can be active in at once, and which one wins.**

A trader on the KYC reminder list and the dormant list gets one call. The plan says which.

- ☐ **Restrict triggers to contacts with a recorded consent basis and a permitted calling window for their country.**

Review the consent field and the window rules with your compliance officer before any automated trigger goes live. The trigger should refuse, not warn.

- ☐ **Decide whether campaign triggers can be fired from the CRM workflow engine or only from the calling platform, and document the credentials each side holds.**

Two systems able to start calls means two places to audit. One is easier to explain to a regulator.

## 5 Sync direction and conflict rules

Every mapped field gets a direction and a winner. Two-way sync without a rule is how a trader's status flips back and forth overnight.

- ☐ **Mark each field as CRM to platform, platform to CRM, or two-way, and cut the two-way list to the minimum.**

Contact details flow in. Outcomes flow out. Do-not-call is the one field that must be honored in both directions.

- ☐ **Set the winner for each two-way field when both sides change it between syncs.**  
Most recent timestamp wins for phone and language. Most restrictive value wins for consent and do-not-call, always.
- ☐ **Decide the sync frequency for CRM-to-platform data: real time via webhook, hourly batch, or at list import only.**  
Contact details rarely change mid-campaign, so import-time sync plus a dial-time do-not-call check often covers it. Outcomes should flow out in real time.
- ☐ **Define how a deleted or merged CRM contact is handled by the calling platform.**  
A merged duplicate must not survive as a second dial. The join key and the merge rule from the mapping section decide this.
- ☐ **Decide whether the calling platform writes to the contact, to an activity record, or both, and who can edit each.**  
Account managers edit contacts. Nobody should edit the call activity except through a correction process.
- ☐ **Write the reconciliation check: a daily count of calls completed versus outcomes written, with an owner who investigates the difference.**  
Sync failures are silent. A count that doesn't match is the only alarm you'll get.  
**Daily check owner:** \_\_\_\_\_

## 6 Webhooks and API handling

If any part of the integration runs through webhooks or your own middleware, these items are the difference between a demo and a system that survives a bad week.

- ☐ **List the call events you subscribe to and what each one updates: started, completed, qualified, appointment or callback booked.**  
Topcalls fires webhooks on these events. Most brokerages only act on completed and qualified; subscribing to everything creates noise in the CRM activity feed.
- ☐ **Verify webhook authenticity on receipt: shared secret or signature check, and reject anything that fails.**  
A webhook endpoint that writes to the CRM without verification is an open door into your contact data.
- ☐ **Make every handler idempotent using the call ID, so a retried delivery doesn't create a second activity or double a counter.**  
Retries happen. Your handler either checks whether it has seen the call ID before, or your CRM fills up with duplicate outcomes.
- ☐ **Define the retry and dead-letter behavior when the CRM API is down or rate-limited.**  
HubSpot and Salesforce both throttle. Queue the update and retry with a delay instead of dropping it. Log what was dropped, if anything, so the reconciliation check finds it.

- ☐ **Store recording and transcript links, not the files, in the CRM, and check who can open them.**

Links keep the CRM light and let access control live in one place. Confirm the retention period for the underlying files with compliance.

- ☐ **Record where API credentials for each side are stored, who can rotate them, and the rotation date.**

A key in a shared spreadsheet is a finding waiting for the next audit.

**Credential owner:** \_\_\_\_\_ **Rotation date:** \_\_\_\_\_

- ☐ **Decide what in-call actions the agent may take through function calling: check a balance, book a callback, update a status, none.**

Each in-call action is a live API call under a trader's ear. Start with read-only lookups and add writes after the test plan passes.

## 7 Testing before go-live

Test with a sandbox CRM or a test segment tagged as such. Never test against live dormant traders.

- ☐ **Create ten test contacts in the CRM covering every account status and every disposition you expect, with your own team's phone numbers.**

Include one contact with a missing language, one with a badly formatted number and one flagged do-not-call, to prove the holds work.

- ☐ **Run one call per test contact and check that every mapped field lands in the right CRM field with the right value.**

Read the CRM record after each call. A field that's close but wrong ("Callback" vs "Call back") only shows up when a human looks.

- ☐ **Trigger a do-not-call request during a test call and confirm the contact is suppressed across every list within the agreed time.**

This is the test compliance will ask about. Screenshot the result and keep it with the sign-off.

- ☐ **Fire a duplicate webhook delivery for the same call and confirm the CRM shows one outcome, not two.**

Most platforms let you resend an event from a log. If yours doesn't, replay it from your own logs.

- ☐ **Take the CRM offline or revoke the API key for ten minutes mid-test, then restore it, and confirm queued outcomes arrive with nothing lost.**

If outcomes vanish during the outage, the retry design from the webhooks section isn't done.

- ☐ **Check the trader-facing side: the account manager sees the task, the email sequence starts, the callback appears at the requested time in the right time zone.**

The integration is done when the retention team acts on it without asking where the data is.

- ☐ **Run the daily reconciliation check once against the test batch and confirm the counts match.**

**Calls:** \_\_\_\_\_ **Outcomes written:** \_\_\_\_\_

- ☐ **Get written sign-off from the CRM owner, the calling owner and compliance before the first live batch.**

Keep the sign-off with the mapping sheet version. When the mapping changes, the sign-off is redone.

## **8 Data protection and access**

The integration moves personal data between systems. These items say what to review and who signs off; they don't replace your compliance officer's judgment.

- ☐ **Document which personal data fields leave the CRM, where they go, and how long the calling platform keeps them.**

Under GDPR this is what a data processing record and a processor agreement describe. Review the list with your compliance officer or counsel.

- ☐ **Confirm the data processing agreement with the calling platform is signed and covers the fields in the mapping sheet.**

If the mapping adds a field later, the agreement may need a look too.

- ☐ **Set the retention period for recordings, transcripts and call summaries on both sides, and who deletes them when it ends.**

Recording rules and retention expectations differ by jurisdiction. Confirm the period per country you call with compliance.

- ☐ **Define how a trader's access or deletion request is handled across the CRM and the calling platform.**

One request, two systems. Name the person who runs it and the steps they follow.

- ☐ **Restrict who in the CRM can see recording and transcript links to the roles that need them.**

Retention and compliance need them. The whole sales floor does not.

- ☐ **Log every outbound call and every CRM write with a timestamp and the system that made it, and keep the log for as long as compliance says.**

When a trader complains, the log answers who called, when, what was said and what changed on their record.

- ☐ **Review the whole plan with your compliance officer or counsel and record the date and the version they approved.**

**Approved by:** \_\_\_\_\_ **Date:** \_\_\_\_\_ **Version:** \_\_\_\_\_

### How to use this checklist

1. Print or copy the checklist and fill in the scope section with your CRM admin and your retention lead in the same room. Thirty minutes is enough.
2. Build the field mapping table in a shared sheet using the columns in the field-mapping section, with the CRM's internal field names, not labels.
3. Agree the disposition list with the retention team before anyone configures the agent. The agent's outcomes are only useful if the team acts on them.
4. Hand the completed sections to whoever builds the connection: your CRM admin, your developer, or the Topcalls team during onboarding.
5. Run the testing section against a sandbox or test segment, collect the sign-offs, and only then import the first live list.
6. Rerun the testing section whenever the mapping sheet changes. A new field is a new integration.

### Book a 30-minute call with a Topcalls co-founder

We'll map your CRM fields and dispositions into a Topcalls campaign, pick the connection path that fits your stack and scope a first test batch.

**[topcalls.ai/book](https://topcalls.ai/book)**