

A FIELD GUIDE FOR MARKETING OPERATIONS

Map, Triage, Hold

Taking control of a marketing stack you did not build.



Sakshi Patel

VERSION 1 · 2026

Contents

The Framework	3
How the method works	4
The Stack Health Check	5
Map	8
Inherited Stack Audit Checklist	10
Data Flow Map	12
Triage	14
Sync and Data Hygiene Audit	16
Attribution Integrity Check	18
Lifecycle and Routing Audit	19
Hold	21
Intake One-Pager	23
Decision Rights and RACI	25
Vendor Evaluation Scorecard	27
Vendor Due Diligence Question Bank	29
AI Output QA Checklist	31
Where to Start	32

The Framework

Almost no one builds a marketing stack from scratch. You inherit it.

Every marketing operations job starts with the same hour. You open the CRM to see what you are now responsible for, and the picture does not resolve. There are workflows no one can explain. There are integrations that were half-built two years ago and never finished, still running. Two teams write into the same field with different rules, so the field means nothing. Reports pulled from the same systems return different numbers, and no one can say which is right. Somewhere in the settings, AI features have switched on inside tools the company already pays for, and nobody made that call. You are expected to look like you have this under control by the end of the month. You cannot even see most of it yet.

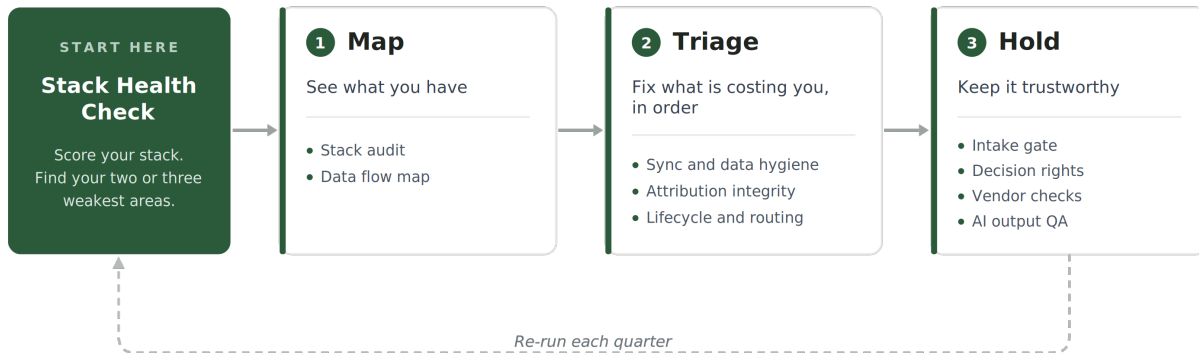
This is the job. Not the exception, the job. Almost no one builds a stack from scratch. You inherit it, with a decade of other people's shortcuts stacked on top of each other, and tools bought to solve problems that left when the people who bought them left. Then it keeps moving. A tool goes in, a team reorganizes, a platform ships another round of AI, and the map you finally drew is already wrong.

Most of the advice written for this moment is useless in it. Governance frameworks assume you are choosing to adopt a tool, so they start with an approval process for a decision that already happened without you. Best-practice guides assume a clean build. The AI advice treats AI as something you go shopping for, when the AI touching your data showed up in a product update you did not read. None of it starts where you are, which is inside a system someone else built, answerable for it today.

This framework starts there. It comes from the chair of the person who has to make an inherited, moving stack work, and who gets the question when the number on the board is wrong. The method has three steps and they run in one order. See what you have. Fix what is costing you. Keep it from sliding back. Map, Triage, Hold. It stays small because the situations it is for do not leave room for anything bigger.

THE FRAMEWORK AT A GLANCE

**Take control of a stack you did not build,
and keep it trustworthy as it changes.**



How the method works

Three steps, one order, and the order is not a suggestion. You map first because you cannot fix or defend what you cannot see. You triage second because you will never have time to fix everything, so you fix in the sequence that pays. You hold last because a stack you cleaned comes apart again unless something stands between it and the next well-meaning change.

Every step ships tools, and they are all in this book, built to fill in rather than admire. You will not use all of them. Score your stack with the Stack Health Check on the next page, take your two or three worst areas, and use the tools for those.

Run it on a cycle. Map once, when you inherit the stack or after something large changes. Re-run Triage and Hold on a schedule. Re-score each quarter and watch the number move. A stack does not sit still, so holding it cannot be a one-time project.

None of this assumes you built the stack, and none of it names a vendor you must use. The examples use a CRM and a marketing automation platform, usually HubSpot and Salesforce, because that pairing is the common case. The method does not care which two you run.

The Stack Health Check

Score your stack before you touch it. Six areas, twenty-four questions, five to seven minutes. For each one, mark a single answer. You get a number out of 100, a band, and the two or three areas to start on.

Visibility: do you actually know what you have?

1. I have a current, written inventory of every tool in our marketing and sales stack, and what each one is for.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

2. I have a map of how customer and lead data moves between our core systems.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

3. I can list every place AI is currently active in our stack, including features built into tools we already pay for.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

4. A new hire could understand how our stack fits together from documentation, without shadowing me for a week.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

Data and sync integrity: can you trust the records?

5. Records stay consistent between our CRM and marketing platform without routine manual cleanup.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

6. We have a working approach to duplicates, and duplicates are the exception.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

7. Field mappings between systems are documented, and I know what breaks if one changes.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

8. When data looks wrong, I can trace where it came from rather than guessing.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

Lifecycle and routing: does flow work without leaks?

9. Every inbound lead has a defined path and a clear owner, from first touch to handoff.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

10. Our scoring or qualification logic reflects how the business defines a good lead today, not a year ago.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

11. Leads do not sit unrouted or get stuck between marketing and sales.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

12. I could explain, end to end, what happens to a lead from form fill to closed deal.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

Measurement and attribution: can you defend the numbers?

13. I could defend our attribution numbers if finance or leadership challenged them tomorrow.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

14. Marketing and sales agree on how core terms like a qualified lead or an opportunity source are defined.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

15. Our reporting reconciles across systems, rather than each tool telling a different story.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

16. I trust our numbers enough to make budget decisions on them.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

AI control: do you control the AI in your stack?

17. I can name every AI feature live in our stack and who is accountable for each one.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

18. I know what customer or prospect data each active AI feature can read, write, or send out.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

19. Something or someone reviews AI-generated output before it reaches a customer or a record of decision.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

20. New AI features get a deliberate yes or no before they are switched on, rather than arriving on by default.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

Governance and ownership: will it stay clean?

21. Nothing new gets added to the stack, a tool or a feature, without passing a basic intake step.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

22. Every core part of the stack has a named owner responsible for it.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

23. Key processes and configurations are documented well enough to survive someone leaving.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

24. We review the health of the stack on a regular cadence, not only when something breaks.

☐ Not true (0) ☐ Somewhat (1) ☐ Mostly (2) ☐ Fully (3)

Score it. Dimension score is the sum of its four answers divided by 12, times 100. Overall score is the average of the six. Bands apply to the overall and to each dimension, so one weak area shows even when the average looks fine.

Score	Band	What it means
0 to 40	Flying blind	High risk. Things are failing where you cannot see them.
41 to 60	Holding together	It works, but it is fragile, and drift is winning.
61 to 80	In control	Solid. The gaps are manageable.
81 to 100	Compounding	The stack is an asset, not a liability.

Start with your worst two or three. Each points to a step and the tool that fixes it.

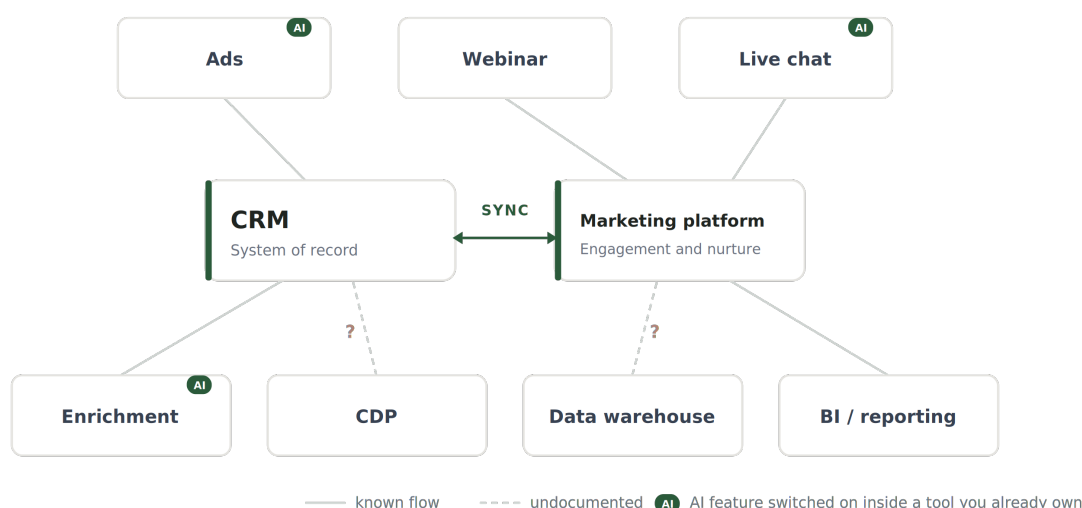
Weak area	Step	Tool to use
Visibility	Map	Inherited stack audit checklist, data flow map
Data and sync integrity	Triage	Sync and data hygiene audit
Lifecycle and routing	Triage	Lifecycle and routing audit
Measurement and attribution	Triage	Attribution integrity check
AI control	Map and Hold	AI surface census, intake one-pager, AI output QA checklist
Governance and ownership	Hold	Intake one-pager, decision rights and RACI, vendor scorecard, due diligence question bank

01 Map

You cannot fix what you cannot see.

YOUR STACK

A CRM, a marketing platform, and everything bolted on around them



Every inherited stack looks something like this. A CRM and a marketing platform at the center, a dozen tools bolted on around them, connections nobody wrote down (the dashed lines), and AI features that switched on inside tools you already pay for (the green tags).

You are accountable for all of it and can see almost none of it. Mapping trades the guesswork for a picture you can trust, which is why this section comes first.

You cannot fix, defend, or govern a stack you cannot see, and the one thing every inherited stack has in common is that nobody holds the whole picture. The knowledge is spread across the people who built each piece, and most of them are gone. What is left lives in settings someone configured once and never opened again. Mapping pulls that picture into one place. It is boring. It is also the only honest starting point, because everything downstream is guesswork until it is done, and most people skip it and pay for that later.

Map runs in three passes.

Pass one is the tools. Go system by system and write down what each one does now, who owns it, who pays for it, and who actually uses it. Write what it does today, not what the person who bought

it hoped it would do. You will find two tools doing the same job, bought by different teams a year apart. You will find a tool no one has logged into since the person who set it up left. You will find one system doing a job you assumed another system owned. That is the point of the pass.

Pass two is the data. Trace how a record moves between systems and settle one question for each object: which system is the master. Contact, account, opportunity, activity. This is the pass that explains most of your reporting problems before you have opened a report, because most reporting problems are two systems disagreeing about who owns the record. When both believe they own it, the numbers never reconcile until you decide.

Pass three is the AI. Open the settings of every tool and write down each place an AI feature is switched on, including the ones baked into platforms you already pay for. There is no contract to follow for these, which is why this pass exists. AI arrives inside tools you already own, in an update, sometimes on by default. The only way to know what is running is to look, and the list is longer than you expect.

When the three passes are done you have a thing most teams never build: an accurate, current picture of their own stack. Keep it alive. The value is not the afternoon you spend building it. The value is having it the next time someone asks how a system works or why a number looks wrong, and pointing at the answer instead of guessing. Two tools do this work. The audit checklist covers the tools and the AI. The data flow map covers the data.

TOOL

Inherited Stack Audit Checklist

Build an honest inventory of what is in your stack. Two of the three passes live here, the tools and the AI. Fill it in. A cell you cannot fill is itself a finding.

Pass 1. Tool inventory. One row per system. What it does today, not what it was bought for.

Tool	Job today	Owner	Who pays	Who uses it	Status	Connects to	Risk

Filled example: Marketing platform · email, forms, nurture, lead capture · owner marketing ops · paid by marketing · used by marketing and SDRs · active · connects to CRM, ads, webinar tool · risk: two-way CRM sync, direction unclear.

Pass 2. AI surface census. Go through the settings of every tool. One row per live AI feature.

Where it lives	Feature	On by default?	Data it touches	Owner	Reviewed?	Risk

Filled example: CRM · AI lead scoring · on by default · all contact and activity data · owner: none · reviewed: no · risk: scores feed routing, unchecked.

Red flags. Check any you find.

- ☐ Two tools doing the same job, bought by different teams.
- ☐ A tool no one has logged into in ninety days.
- ☐ An integration no one can explain.
- ☐ The same field filled differently by two teams.
- ☐ An AI feature that is on and no one remembers enabling.
- ☐ Admin access far wider than the work requires.
- ☐ A record two systems both claim to own.
- ☐ Automations with no owner and no documentation, touching routing or scoring.

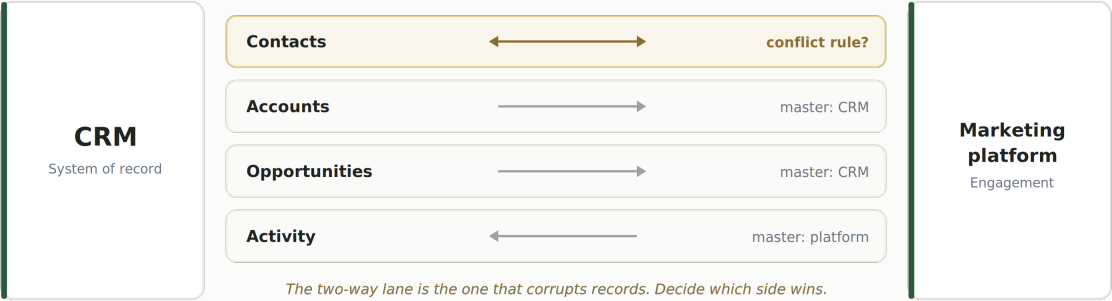
Done when.

- ☐ You can name every tool and the job it actually does.
- ☐ You can name every live AI feature and whether it was a choice or a default.
- ☐ A new hire could understand the stack from what you wrote.
- ☐ It lives somewhere it can be kept current.

TOOL

Data Flow Map

DATA FLOW
Which system owns each record



Document how data moves and settle the question behind most reporting problems: which system owns each record. This records the current state. The fixing happens in Triage.

The flows. One row per connection between two systems.

From	To	What data	Direction	Trigger	On conflict, who wins	Owner	Known issues

Filled example: marketing platform → CRM · contacts, form fills, engagement · two-way · real-time · on conflict: undefined · owner marketing ops · issue: lifecycle field overwritten.

The source of truth. One master per object. If two systems claim it, that is a finding.

Object	System of record	Why it is the master	Who else holds a copy
Lead or contact			
Account or company			
Opportunity or deal			

Object	System of record	Why it is the master	Who else holds a copy
--------	------------------	----------------------	-----------------------

Activity or
engagement

Done when.

- ☐ Every flow is written down with its direction and trigger.
- ☐ Every object has one named system of record.
- ☐ You can trace any number on a report back to where its data came from.
- ☐ Every two-way flow has a conflict rule, or is flagged because it does not.

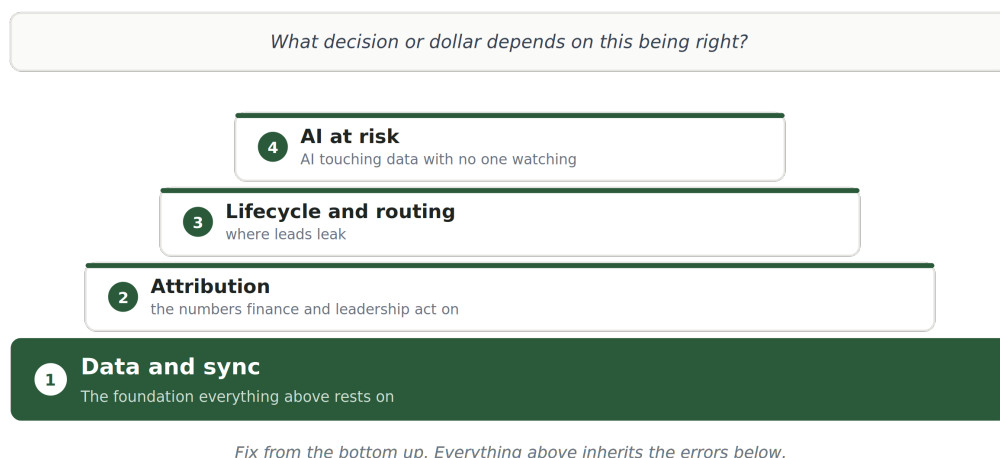
02

Triage

There is always more broken than you have hours for.

TRIAGE

Fix in order, because it stacks



Now you can see the whole stack, and the urge is to fix all of it. Kill that urge. There is always more broken than you have hours for, and the people who try to fix everything stay busy and stay behind. The skill is the order, and the order is not what annoys you most. It is what costs the business most.

One question sorts the list: what decision or dollar depends on this being right. Ask it of every problem and the list sorts itself, and it sorts the same way on almost every stack, because the pieces sit on top of each other.

Data and sync go first. Everything rests on them. Wrong or out-of-sync records mean wrong reporting, wrong routing, and any AI you point at that data returns wrong answers with full confidence. Fixing anything above a broken data layer is repainting a room while the pipe behind the wall still leaks.

Attribution is next, because these are the numbers other people spend money on. Finance asks where pipeline came from. Leadership asks what to fund. Both answers come out of attribution, and a number you cannot defend is worse than no number, because people act on it anyway. Perfect

attribution is not the goal and does not exist in B2B. A number that reconciles and that you can stand behind is the goal.

Lifecycle and routing come third. This is where pipeline leaks with no alarm attached. A lead comes in, has no defined path and no owner, and sits. Nothing errors. Nothing pages anyone. Add up a quarter of that and you have lost real money to a gap no dashboard showed.

Last is the AI you found on the map. Now that every live feature is visible, look at the ones touching customer data or shaping a decision with no one checking the output, and fix or fence those first.

Triage is where the score comes from. Grade each area, find the worst, fix it this quarter, repeat next quarter. The list never hits zero. It is not supposed to. Three tools cover the first three priorities. The AI surfaces get handled in the last section.

TOOL

Sync and Data Hygiene Audit

The first fix, because the data is the floor. Work off your data flow map. This finds where reality and the map disagree, ranks it, and puts in the guardrail so it does not come back. Fixing a thing once without a guardrail means fixing it again next quarter.

Sync checks. Run on each flow. Check what holds.

- ☐ The sync behaves the way the map documents it, not some other way.
- ☐ A conflict rule exists and holds. The right side wins every time.
- ☐ No silent failures. Failed syncs are visible and handled, not piling up in a log no one reads.
- ☐ Nothing overwrites good data with blanks or defaults.
- ☐ Timing is understood. You know whether each flow is real-time or scheduled, and the lag is acceptable.

Data hygiene checks.

- ☐ Duplicates are the exception, and you know the merge rules.
- ☐ The fields routing, scoring, and reporting depend on are populated.
- ☐ The same field means the same thing everywhere and is filled the same way by every team.
- ☐ Old records are aged out or marked.
- ☐ Validation exists on the fields that matter.

Rank what you find.

Severity	Meaning	When to fix
Breaks reporting	Numbers leadership or finance act on are wrong	First
Breaks routing or scoring	Leads go to the wrong place or wrong priority	Next
Cosmetic	Real but low stakes	When you reach it

Fix, then prevent. For each problem, install the guardrail.

Problem	Guardrail
A field gets overwritten	Set and enforce the conflict rule; the source of truth wins
Duplicates	Dedup rules plus a merge policy, not a one-time cleanup
Missing required fields	Validation or required-field rules at entry
The same field means two things	One documented definition, and picklists instead of free text
Silent sync failures	An alert or standing check on the sync error log

Done when every flow behaves as the map says, two-way flows have an enforced conflict rule, duplicates are the exception, the important fields are populated and mean one thing, and you would put these numbers in front of finance.

TOOL

Attribution Integrity Check

The second fix, once the data underneath is clean. The goal is not perfect attribution. It does not exist in B2B. The goal is a number you can defend and that sales does not dispute. Most attribution fights are definition fights wearing a costume.

Definitions worksheet. Write yours down. Then get sales to sign the last column.

Term	Your definition	Sales agrees?
Qualified lead (and at what point)		☐
Source (first touch, last touch, multi-touch)		☐
Sourced vs influenced		☐
Attribution model in use		☐

If sales would define any of these differently, that gap is your first finding.

Model fit.

- ☐ It reflects real buying paths, including long cycles and many touches.
- ☐ You know what it misses: offline, self-reported source, dark social, referral.
- ☐ It is not crediting things that no longer matter while ignoring things that do.

Reconciliation. Pick numbers you report and tie them out.

Number	CRM says	Platform says	Reconciles?
			☐
			☐

Defensibility test. Take any number you report and check.

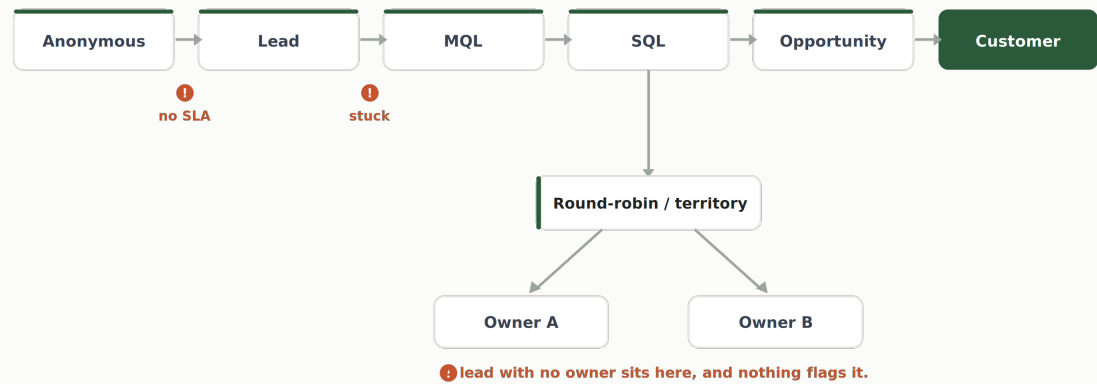
- ☐ You can trace it to the records it came from.
- ☐ You can explain the logic in plain language.
- ☐ It would survive finance pushing on it.

If any of these is no for a number you report regularly, that is what to fix first.

TOOL

Lifecycle and Routing Audit

LIFECYCLE AND ROUTING
Find where leads leak



The third fix, and the one that recovers pipeline nobody logged as lost. A lead with no path and no owner sits, and nothing tells you. Walk the real path, not the diagram on the wall.

Walk the path. One row per lead type. Write the stages, the owner at each, and where it leaks.

Lead type	Path (stages)	Owner at each stage	Leak point
Inbound			
Demo request			
Content			
Event			
Partner			

Find the leaks.

- ☐ Every lead type has a defined path and a named owner at every stage. No orphans.
- ☐ There is an SLA on follow-up, and it is met.
- ☐ Nothing sits unrouted or unworked.
- ☐ Leads do not fall out between marketing and sales.

Routing and scoring.

- ☐ Assignment rules are current. Leads reach a real owner, not someone who left.
- ☐ Scoring reflects how the business defines a good lead today, re-derived from how sales qualifies now.
- ☐ Good leads are not buried, and weak leads are not pushed as hot.

Done when every lead type has a path and an owner, nothing sits unrouted or falls out, routing matches how the team is organized, scoring is current, and you could explain end to end what happens from form fill to closed deal.

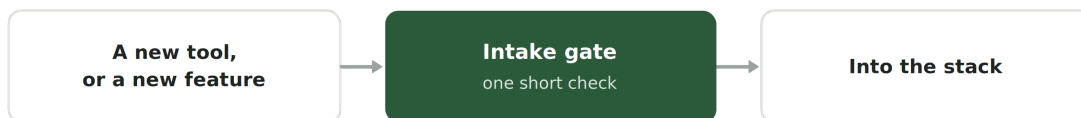
03

Hold

Stacks decay.

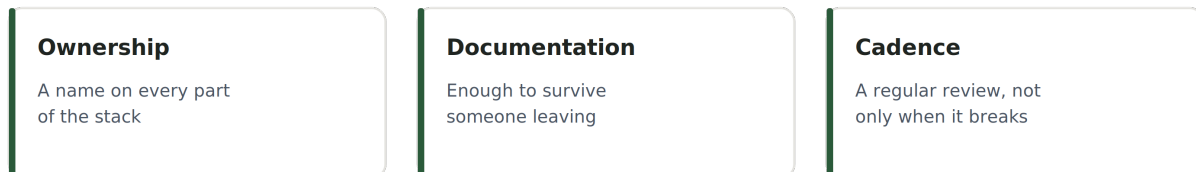
HOLD

Keep it trustworthy as it changes



The gate covers features too, not just tools, because AI arrives without a purchase.

Then keep it clean:



INSTRUMENTS Intake one-pager · Decision rights and RACI · Vendor scorecard · Due diligence question bank · AI output QA

Triage is not a state you get to keep. Stacks decay. Six months after a cleanup, someone has bought a tool on a personal card, a new manager has switched on a feature they saw in a webinar, and the field mapping you corrected has been reverted by an admin who never knew why it was set that way. This is not a discipline problem, and treating it as one is the mistake most teams make. It is structural. Nothing sits between a person's good intention and the production system, so every good intention lands straight in production. Hold is the layer you put in that gap.

The gap has a name in most companies: there is no gate. Anyone can add anything at any time, so over time everyone does. The one control that changes this is a small intake step. Before a tool goes in or a feature gets switched on, it clears a short, known check. The check does not need weight. It needs to exist, so that adding something is a decision someone made instead of a thing that happened.

Here is the part almost every governance model misses. The gate has to cover features, not just tools. The old reflex was to scrutinize anything with a purchase order. The AI landing in your stack now has no purchase order. It came in an update, inside a tool you already owned, sometimes

switched on before you knew. An intake step that only fires when money moves never fires for any of it. So the gate watches for capabilities going live, not just contracts getting signed.

Three more things keep a stack honest over time, and none of them is dramatic. Every core part has a named owner, so “who owns this” is never a shrug. The important configurations are written down well enough that the stack survives the day someone leaves, and in this field people leave. And the whole thing gets reviewed on a schedule, so drift shows up while it is small instead of during an incident. That is the section. Five tools put it in place.

TOOL

Intake One-Pager

This is a form. Fill it in before a tool goes in or a feature gets switched on. One page, a few minutes, then a decision. A gate people route around is not a gate, so keep it this short.

Requested by	Proposed owner	Date
--------------	----------------	------

#	Question	Answer
1	What is it, in one line?	
2	What job does it do that we cannot already do today?	
3	Does anything in the stack already do this?	☐ No ☐ Yes →
4	What data does it touch (read, write, send out)?	
5	Does it use AI, and does it train on our data?	☐ No AI ☐ AI, no training ☐ AI, trains → can we opt out? ☐ Yes ☐ No
6	How is it arriving?	☐ Buying it ☐ Feature in a tool we own ☐ On by default (keep or kill)
7	If it goes wrong, what breaks, and who is affected?	
8	Cost, and whose budget?	

On question 2, if the honest answer is “what tool X already does,” stop and use X. On question 8, a feature often costs nothing, and questions 4 to 7 still decide it.

Decision	☐ Approve	☐ Approve with conditions	☐ Reject
-----------------	----------------------------------	--	---------------------------------

Decided by

Decision

☐ Approve

☐ Approve with conditions

☐ Reject

Date

TOOL

Decision Rights and RACI

Two failures come from nobody owning things. A system breaks and has no owner, and you get a shrug. A cross-team decision has no owner, and it stalls or gets made by whoever is in the room. Keep this light. Most of the stack needs one owner. Only the decisions that cross teams need the grid.

Own the parts. One accountable owner per area.

Area	Owner
CRM	
Marketing platform	
Integrations and sync	
Attribution and reporting	
Lead routing	
AI surfaces	
Data quality and governance	

If an area has no owner, that is the first gap to close. If it has three, decide who is actually accountable.

The cross-team decisions. Accountable is the one person who decides. Responsible does the work. Consulted has input first. Informed is told after.

Decision	Accountable	Responsible	Consulted	Informed
Add a tool or feature				
Turn on or keep an AI feature				
Change the data model or source of truth				
Change attribution definitions				
Grant admin access				
Sign a vendor or data agreement				

Filled example: turn on or keep an AI feature. Accountable, marketing ops lead. Responsible, marketing ops. Consulted, IT security and legal. Informed, marketing leadership.

One accountable per decision. If two people are both accountable, no one is.

TOOL

Vendor Evaluation Scorecard

A scorecard that produces a feeling instead of a number changes nothing. This one produces a number, and it makes you clear the dealbreakers before you are allowed to like the demo.

Step 1. Dealbreaker screen. If any is true, stop. Do not score it.

- ☐ We cannot export our own data in a usable format without the vendor's help.
- ☐ It trains on our data and there is no opt-out.
- ☐ No one on our team will own it.
- ☐ It duplicates a tool we already run, with no material gain.
- ☐ It cannot meet a security or compliance requirement we already hold.

Step 2. Score each 0 to 3. 0 fails or unknown, 1 weak, 2 adequate, 3 strong.

#	Criterion	A 3 looks like	Weight	Score
1	Need and non-overlap	A real unmet job; nothing in the stack does it	×2	
2	Integration	Native, documented, two-way with our CRM and platform	×2	
3	Data control and portability	We know what it touches, where data sits, and can export it	×2	
4	Security posture	Current SOC 2 Type II or equivalent, documented	×1	
5	AI handling	No training on our data; output is inspectable (score 3 if no AI)	×1	

#	Criterion	A 3 looks like	Weight	Score
6	Adoption fit	The team will use it; low admin burden	×1	
7	Exit	We could remove it and get our data back without pain	×1	

Step 3. Total it, out of 30. The three doubled criteria are doubled on purpose. A tool that scores well everywhere else but fails on need, integration, or data control makes more work than it saves.

Step 4. Read it.

24 to 30: strong, proceed.

16 to 23: viable with real gaps you fix or accept knowingly.

Below 16: a demo talked you into it. Walk away.

TOOL

Vendor Due Diligence Question Bank

The questions that get the answers the scorecard needs, and the ones that surface the problem before you sign. Make them answer in writing. A verbal yes is not a commitment. Record each answer next to the question.

Data

1. Where is our data stored and hosted? _____
2. Who on your side can access it, and under what controls? _____
3. Can we export all of it, any time, in a usable format, without your help? _____
4. How long do you retain it, and how is it deleted when we leave? _____
5. Who are your sub-processors? _____
6. Is it encrypted in transit and at rest? _____

Security

1. What certifications do you hold, and can we see the report? _____
2. When were you last penetration tested, and by whom? _____
3. What is your breach history, and how fast do you notify us? _____

AI

1. Do you use our data to train or improve your models? Can we opt out, and is opt-out the default? _____
2. Is our data ever used to improve models other customers benefit from? _____
3. What model sits behind the AI, and where does it process our data? _____
4. Can we see how a given output was produced? _____
5. What happens when the AI is wrong, and what human oversight exists? _____

Integration and contract

1. How does it connect to our stack, and is the sync real-time or batch? _____
2. What happens to our integration when you ship an update? _____
3. What drives the price up, and is there a cap at renewal? _____
4. What are the termination terms, and how do we get our data back? _____

Red flags. A bad answer sounds like this.

"We may use your data to improve our services." That usually means they train on it. Pin it down.

"SOC 2 in progress." Not the same as holding it.

No clean way to export your own data without them.

Vague answers on where the AI processes your data.

Anything they will not put in writing.

Treat an evasive answer on data or AI as a finding, and often a dealbreaker.

TOOL

AI Output QA Checklist

The common failure with AI in a stack is not bad output. It is output nobody checked, going straight to a customer or into a record that drives a decision. One control matters most: a human looks before it ships or takes effect. You cannot hand-check everything, so match the scrutiny to the stakes.

Every AI output.

- ☐ Accurate. No invented specifics, numbers, or names.
- ☐ On brand and appropriate.
- ☐ Safe. No leaked data, no compliance problem, nothing biased.
- ☐ Reviewed by a person before it went out or took effect.
- ☐ Traceable. You can tell it was AI-generated and audit it later.

By output type.

- ☐ Content (emails, copy): claims accurate, brand voice holds, personalization resolves, a person reviews before send.
- ☐ Lead scores: spot-checked against leads you know are good or bad, not silently misrouting, owned by someone.
- ☐ Data enrichment: writes correct values, does not overwrite good data, source noted, writes sample-checked.
- ☐ Chat and agent replies: accurate and on brand, escalates to a person when it should, does not promise what it should not.
- ☐ Summaries feeding decisions: accurate, does not overstate confidence, the people deciding know it came from AI.

Match review to stakes.

Output	Review level
Customer-facing or decision-driving	A person reviews every time
Low-stakes internal	Spot-check, not every item

When in doubt, review it. A bad output in front of a customer costs more than a check.

Where to Start

None of this needs a big budget, a clean build, or authority over IT. One person can start this week. Map the stack once, triage in order, hold the line, and run it again when the stack has moved enough to need it. Most stacks need it sooner than their owners want to admit.

The fastest way to know where you stand is the six-area score at the front of this book. Can you see it. Can you trust the data. Do leads move without leaking. Can you defend the numbers. Do you control the AI in it. Will any of it stay in order. Your worst two or three are where to start. That is the method. See what you have, fix what is costing you, hold the line.

One honest limit. This will not turn a bad stack into a good one in a single quarter, and it will not survive a team that refuses to own anything. What it does is give one practitioner a way to take a stack that runs their day and make it run on purpose instead.