

POINT OF VIEW

FOR LEADERS

Beyond the Use Case

How AI changes the way companies decide,
prove, build and work.

AI is not a category. Not a project. Not a buzzword to integrate.

It is a new **layer in the operating system of your company.**

It changes how decisions get prepared, how knowledge moves, how your people spend their day.

It is not a tool you bolt on top. It is how the company runs, and AI turns that from something you write down into something that runs.

Treat it as a toolkit and you get tools. Treat it as a layer and you change the company itself.

This is the turning point. And where this begins.

Doing the work was the advantage. Now anyone's agent can do it.

For decades, the advantage was executing better than rivals. Agents collapse that. What is left is [what an agent cannot hold](#): the judgment to choose what is worth doing, the relationships you are trusted with, and the connected knowledge that is yours alone. So the advantage moves up, to the layers most companies skip. And it moves now: the agents are ready, and the people who carry today's advantage are leaving faster than they are replaced.

LAYER 1 / 4 · WHO DECIDES

Most AI programs fail before they begin.

It rarely fails on technology or strategy. It fails because the company never agreed this would happen in a way that costs someone something if it doesn't.

The first move is not a use case. It is a decision, written down, with names on it.

This is layer one of the operating system: the decision itself. It stops being a charter that gets filed and becomes a call the company keeps making. Nothing above it holds until this one does.

01

A steering committee won't carry this. A standing position will.

Decision authority comes from proximity to the work.

WHERE MOST COMPANIES ARE

- AI owned by IT, sponsored by everyone, accountable to no one
- A committee that meets monthly and decides nothing
- "AI is a technology topic"
- **The works council and the skeptical department head are watching from the sidelines**

WHERE THIS GOES

- A small, named group with the authority, and the willingness to spend political capital
- Decisions made between the meetings, by people who answer for them
- "AI is how we run the business"
- **The works council and the skeptical department head are in the room, doing the work**

USE CASE

Copilot goes live in sales. Six months on, most reps barely touch it.

The tool was never the problem. No one close to the work decided how it fits the day: which data it may use, what good prep looks like, how skeptics get pulled in, how use even gets measured. A monthly steering committee sits too far from the visit to answer any of it. **A named group across sales, IT, data protection and the works council decides it between the meetings**, and adoption follows. The technology was ready in a week. The decision is what drifted.

A coalition needs what a committee never grants: the decision placed where the knowledge is.

01 The people who know are allowed to decide.

Not advise. Decide. If the call still has to climb to the most senior person in the room, you have a committee.

02 Rank backs the call, it does not make it.

Cover, not control: the senior people pre-authorize it, absorb the cost when it is unpopular, and defend it upward. Decoupled from the hierarchy, not against it.

03 Each member has something to lose.

Something operational to lose, not just reputation on the line. A reputation survives a failed program; an operational stake does not. With nothing at risk, no one holds the line when the call gets expensive.

04 The coalition holds a position.

One clear thesis on AI, stated the same in every room, without notes. If it changes with the audience, it is not a position yet.

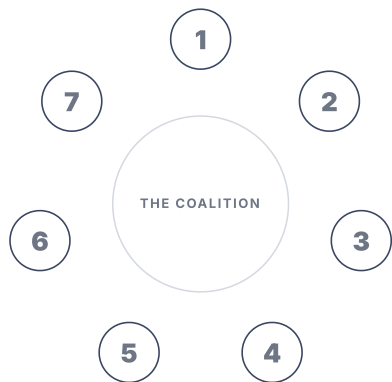
A committee asks who outranks the problem; a coalition asks who knows it, and backs them to decide.

Seat the table.

A coalition is five to seven people, named. Add each by name, then say what they are at the table: the owner who answers for it, someone who does the work, or someone who could stop it.

Name five to seven people. For each, mark what they are at the table.

Beside each seat, write the name and tick the role. **O** owner, **W** does the work, **S** could stop it.



1	_____	☐ o ☐ w ☐ s
2	_____	☐ o ☐ w ☐ s
3	_____	☐ o ☐ w ☐ s
4	_____	☐ o ☐ w ☐ s
5	_____	☐ o ☐ w ☐ s
6	_____	☐ o ☐ w ☐ s
7	_____	☐ o ☐ w ☐ s

HOW TO READ YOUR TABLE

- More than one **Owner**? That is a problem. If two people answer for it, no one really does.
- No mark under **Could stop it**? A problem. A table of supporters cannot carry a real decision; the blocker has to sit inside it, not wait outside.
- Fewer than **five** people? Too thin to hold the organization behind the call.
- More than **seven**? Too big to decide. It slides back into a committee.

A seated table is not a coalition yet. It becomes one when all of them say the same sentence. That is the next test.

~~"Central or distributed"~~ is the wrong question.

The real question underneath it is who holds which decision. Pick one pole and you have answered the wrong question. A central center of excellence produces frameworks nobody uses. Pure decentralization produces fifteen stacks and one shared regret. The coalition's job is to place each decision on purpose, on day one: which calls sit at the center, which sit with the domains, instead of choosing a side.

CENTRAL HOLDS	DISTRIBUTED HOLDS
governance & risk · guardrails · platforms & identity · security & model contracts · sovereignty posture · standards · knowledge	which use cases matter · which processes change · what "good" looks like here · acceptable trade-offs · operational ownership · domain judgment

Central owns how. Distributed owns what. Confuse the two and both fail.

The one-sentence test.

You have the table. Its first deliverable is a position: one sentence the CEO can say in any room, without notes, long before a roadmap exists. Five minutes will tell you whether you have one.

Write your company's AI thesis. One sentence. No notes.

Can't write it? Then you don't have a position yet, only a topic.

First, the table. Everyone you just seated writes theirs, apart. If the sentences do not match, you have a committee, not a coalition, and the work is here, not in the roadmap.

Then, past the table. Test it on the people who were not in the room: a wary CFO, a skeptical department head, the works council. If it still holds, that sentence is your mandate. Put names to it this week.

The hard part was never knowing what to do. It is getting the table to write the same sentence.

LAYER 2 / 4 • PROVE IT

The demo impresses. Nothing changes.

This is how the first wave usually ends. The technology worked, the room was impressed, and six months later no one can point to a number that moved. The cause is rarely capability. It is the choice of where to aim, made too late and never defended out loud.

Layer two is where it has to carry weight. Proof stops being a demo you watch and becomes a number you commit to: one that moves, gets used, and repeats. A pilot no one can attribute a result to has proven nothing.

02

A list of use cases is where programs stall.

WHERE MOST COMPANIES ARE

- A spreadsheet of a hundred use cases, ranked by no one
- Half of them never needed AI, just a fixed process
- Every owner means something different by “AI”
- No logic for what comes first, so nothing does
- **A shopping list waiting for budget**

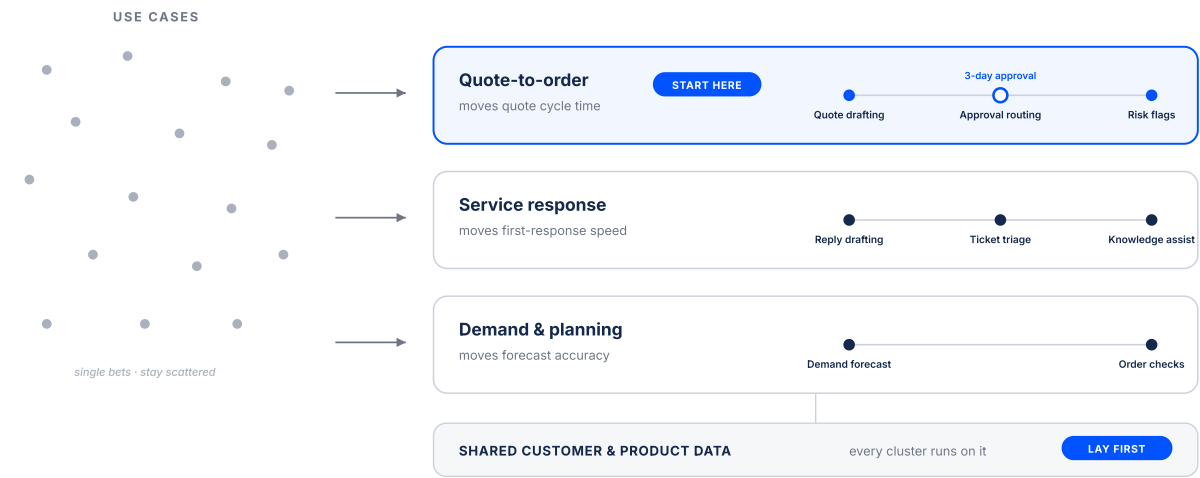
WHERE THIS GOES

- Only where AI is the difference, not where a rule would do
- Use cases grouped by the business outcome they share
- One owner per cluster, accountable for that outcome
- A sequence: what comes first, what compounds on it
- **A roadmap where each win pays for the next**

Cluster the use cases into value chains.

A flat list of use cases stalls. Look at how each value chain actually runs today, then group the use cases into the chain they move. A cluster moves a number; one use case rarely does. The real lever is often a bottleneck, like a three-day approval, not more AI.

REPRESENTATIVE · SCATTERED USE CASES → VALUE-CHAIN CLUSTERS



Many use cases. Clustered along value chains, a few move the number; alone, most don't.

Rank the clusters. Start where it's real and reachable.

Clusters set the direction. Read each on a few dimensions, and the order falls out: what is reachable now comes before the bold bet, however large.

REPRESENTATIVE · THE CHAIN'S CLUSTERS, RANKED

The bold bet ranks last.

Each cluster read on four dimensions, impact weighing heaviest. Order = impact × reachability, not impact alone.

			IMPACT	FEASIBILITY	DATA	SCALABILITY
FOUNDATION · shared customer & product data		every cluster runs on it				LAY FIRST
1	Cluster A · quote-to-order shortens the quote-to-order cycle	START HERE	● ● ●	● ● ○	● ● ●	● ● ○
2	Cluster B · service response lifts first-response speed		● ● ○	● ● ●	● ● ○	● ● ○
3	Cluster C · demand & planning high impact, scales well, but not yet reachable	BOLD BET	● ● ●	● ○ ○	● ○ ○	● ● ●

A first win is the one you can finish and point to. Save the bold bets for when you have proof.

Name the ambition. Then the number under it.

Every cluster needs two things behind it: an ambition, and a number one level down. The ambition is where this value chain could get to, in plain words. The number is the KPI that proves it, with a target and a date. Closing the gap between them is the work. This is the test the first wave usually fails.

First, the ambition for your value chain, in plain words.

One level down, the number that proves it: which KPI, from what to what, by when?

Nothing committed? Then it is a pilot, not a first win. Pilots impress. They do not count.

The number didn't move? Then stop it. Sunk cost is not proof.

Ask the three who see it differently to name theirs, apart: management, the business, IT. Different answers mean a list, not a cluster.

One KPI, signed and dated. That is the outcome the cluster serves. Everything else sequences behind it.

Name it before you build it. A KPI you commit to in advance is the only proof that survives the demo.

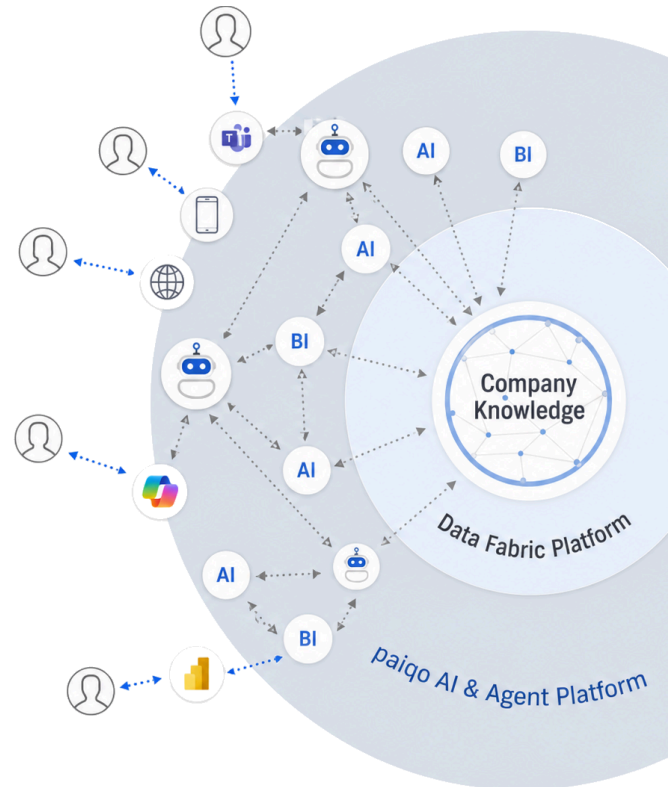
The foundation is the part no one demos.

First wins run on borrowed scaffolding. A dataset someone cleaned by hand, an API key, a quiet exception from IT. It works once. Then you try to scale, and the borrowed parts are gone. What AI runs on is broader than data: the data it can reach, the operating model that runs and watches it, the governance and privacy it lives inside, the identity that decides what it may touch, and the new roles that own all of it. None of it demos well, which is why it is the first thing a committee cuts. You build it for the use cases that move a business KPI, not for its own sake. Get those right, and the foundation under them moves the same number.

This is layer three: what everything above it runs on. Skip it, and the agents you were promised never arrive.

Agents reach the source directly. The foundation still decides.

- 01 The old rule.** For two decades the move was to centralize: pull data into one platform, away from operations, so people could see across the business. Build the warehouse first, report on top. Companies waited, and shipped little for years.
- 02 What changed.** Agentic flips that. The agent acts inside operations: it reaches the source systems directly and reasons on top. It does not wait for a finished warehouse. What it needs at the center is not a warehouse to read but a knowledge layer to act on, so it grasps the whole context across systems. *To an agent, raw data is noise. Connected knowledge is power.*
- 03 What did not change.** Clean data, governance, security, a process that holds, these still decide whether it works. Skip the connecting layer and every team wires its own agent to its own corner: fifteen agents, no control over any of them.



The fallacy is that the foundation is a prerequisite. It is a requirement you build alongside the agents, not a tower you finish before them.

The wall you see coming. Invite it in.

Part of the foundation is technical. The rest is organizational, and it starts with the part everyone braces for: the wall. Legal, regulation, the works council, data protection, security. Each can stop the whole thing. Each is right to. The wall almost always arrives late, around the month the pilot first wants real data and real users, and late, it reads as an obstacle. Brought in on day one, the same people turn a constraint into a design input: a boundary named early is something you build *with*, not around.

A blocker you planned for is a constraint. A blocker that surprises you is a wall.

Most operating models are written down. The ones that hold are wired in.

The classic operating model is a document: governance, controls, who approves what, the standards everyone signed. Written once, filed, ignored, because nothing makes it bind. Agents move faster than any binder is read or any committee convenes, so a written model cannot govern them at all. What governs them is not the old model automated but a different kind of thing: it stops being something you read and starts being something that runs.

WHERE MOST COMPANIES ARE	WHERE THIS GOES
• Policy lives in a binder. • A rule holds if someone remembers it. • Compliance is a person checking, after the fact. • Governance is a quarterly gate.	• Policy is encoded: data, structures, guardrails. • The rule is enforced the moment the agent acts, not audited later. • Compliance becomes a property of the system. • Governance runs continuously, by itself.

CONCRETELY

An export rule stops being a PDF audited each quarter. It is encoded, so the agent cannot ship a restricted item at all: the boundary lives in the system, enforced the moment it acts, every attempt logged.

When the operating model enforces itself, the routine stops needing a person to push it. The people move up to the judgment.

Audit the ground under your best use case.

Take the use case you are proudest of. Eight things hold it up. Borrowed parts work once, then vanish when you scale. For each, be honest: do you own it, or are you borrowing it? Click the ones you are borrowing.

For each, mark whether you own it or are borrowing it.

☐ Own	☐ Borrow	The data it uses, clean and repeatable
☐ Own	☐ Borrow	Access to the source systems
☐ Own	☐ Borrow	Permission to run on real data
☐ Own	☐ Borrow	A named owner, not just the person who built it
☐ Own	☐ Borrow	Governance signed off for scale
☐ Own	☐ Borrow	A way to fix it when it breaks
☐ Own	☐ Borrow	Its rules enforced in the system, not in a document
☐ Own	☐ Borrow	A way to watch and stop it in production

HOW TO READ YOUR SCORE

- Count what you marked **Borrow**. The goal is to own all eight before you scale.
- 1-3 borrowed?** Close. Name who owns each borrowed part, and when it becomes permanent.
- 4-5?** A pilot dressed as a system; the foundation is not there yet.
- 6 or more?** The demo will not survive the second team.

The demo borrows. The system owns. The foundation is the distance between them.

The work gets done by agents. The hard part is the organization, not the technology.

This is the layer everyone points to and almost no one reaches. What stops them is the slowest thing in any business to change: who does the work, how teams are shaped, what a role is even for. The agents have crossed from answering to planning and acting. They are ready long before the organization is.

This is layer four: how work happens once AI runs underneath everything. It stands on the first three holding under it. This is where the layers stop being a program you run and become how the company works.

Most companies automate the task. Few rethink the work.

WHERE MOST STOP

- AI as a faster typewriter. Same tasks, same roles, less time each.
- A copilot in every seat. The org chart does not move.
- Productivity ticks up, then plateaus.
- **Faster work, same work.**

WHERE THIS GOES

- Agents run whole processes. People set the intent and own the outcome.
- Roles move from doing to directing: the org chart grows a second column of agents under each person, to steer, review and correct. Even a purely operational role becomes a steering one.
- The organization forms around outcomes, not functions. Not every role survives the redraw.
- **The work is run, not done.**

USE CASE

The same process, automated, then rethought.

The difference is **who does the work**. **Automate the task:** the person still does it, only faster. Everyone gets a copilot and clears their own pile of documents in less time; the process and the roles stay the same, so once everyone has the tool the gain is banked and the curve flattens. **Rethink the work:** the agent does it, and the person does a different job. The agent clears the whole queue end to end, and people stop processing documents and start judging the exceptions and setting the rules it follows. The work changed, not only its speed, so the curve keeps climbing.

From a tool to an operating model. The cost climbs where the organization has to change.

01

Prompt-by-prompt

Tools in people's hands, nothing structural behind them.

Costs almost nothing, which is why most companies quietly stop here and call it transformation.

02

An assistant in every seat

A faster version of the company you already are. Same shape, more speed, gains that tick up, then plateau.

Costs licenses and adoption, not courage. The trap: paying for tools and naming it change.

03

Agents run the work

Defined work runs without anyone pushing it. You manage the system, not the task.

This is the jump, the first rung that asks the organization to change: the foundation, new roles, governance as a standing function. Not just to buy.

04

Run on agents

You operate the company by describing what you want. You design the intent, agents handle the process.

The deepest rewrite, the highest cost. Right for some businesses, wrong for others.

BUYING TOOLS

CHANGING THE COMPANY

The cost jumps once buying tools turns into changing the company. The far right is a choice to make on purpose, not a finish line, and the real failure is drifting into an answer instead of choosing one.

When agents do the work, judgment becomes the job.

The fear is that AI takes jobs. In many companies the harder truth runs the other way: roles sit unfilled, and experience retires faster than it can be replaced. Agents take on what is already going uncovered. What they cannot take on is judgment: choosing the frame, not optimizing inside it. The frame is what is worth doing, whether it is right, and what good means here:

01 **Decide what to pursue, and whether it is worth doing**

The agent optimizes toward a target. A person chooses the target, and whether it is the right one at all. A value call, not a computation.

02 **Handle what no rule covers**

Every process has the case the spec never saw. The agent fails literally there. A person has to notice, and decide.

03 **Own the result**

When it goes wrong, someone answers. An agent can be made reliable, observable, governed, never accountable. It has nothing to lose. That stays a person's, always.

None of this is the leftover; it is the most durable part of the work. You can own a decision, or be trusted with a relationship, only when you can be held responsible for it. And it is no longer only the manager's: whoever now steers agents carries the judgment too.

Someone always has to answer. An agent never can.

An agent can do the work. It cannot be the relationship.

AI can speed the process, surface the information, carry the load. But trust forms between people, not with a process. The relationship, and the person someone reaches when it matters, stay human. As the people who hold those relationships leave, an agent cannot inherit them; only a person earns them back.

You can make an agent reliable, observable, governed, consistent. Trust is none of those. It is extended to people, never to a process. The agent only borrows it: it spends what someone else built, every time it acts, and gives none back. Someone has to stand behind the output, and be who the customer reaches when it matters.

An agent can spend trust. Only a person can earn it.

Name the hand-off, and the part a person keeps.

Drafting the options is the agent's job. Deciding which customer is worth bending the rule for is not.

The process you would hand to agents

The one judgment a person keeps

HOW TO READ IT

- Both lines must be filled. An empty second line means automation, not a rewrite.
- First line: the work an **agent runs end to end**.
- Second line: the call a **person still owns**: the exception, the risk, the judgment.

If agents do the work and a person keeps the judgment, that is the rewrite. If the second line is empty, it is just automation.

You can only build as high as your **weakest layer** holds.

Four layers, one system. Skip one and everything above it wobbles,
which is why most programs stall, not for lack of technology.

01

Build the Coalition

Who decides

02

Ambition & First Wins

Prove it

03

AI-Ready Foundation

What it runs on

04

Agentic Organization

How work happens

Each layer makes the same move: from written down to running.

So do not start at the top, but at the layer holding you back. Build them as living layers, not finished floors, and AI stops stalling in pilots and becomes how the company runs.

A self-check across all four layers follows on the next pages.
Work through it to find the layer to start with.

THE AUTHORS

Who wrote this.



Carena Barkow

HEAD OF TECH ADVISORY

Carena leads Cloudflight's Tech Advisory practice. She works with executive teams on the hard part of AI: the decisions, the operating model, and the organization that has to live with them.

[LinkedIn](#) ↗



Felix Schubert

PARTNER · TECH ADVISORY

Felix is a partner in Tech Advisory. He helps companies move from AI ambition to results that hold up, the wins that earn the right to go further.

[LinkedIn](#) ↗



Florian Zoller

PARTNER · TECH ADVISORY

Florian is a partner in Tech Advisory. He works where strategy meets delivery, building the layer alongside the people who have to run it.

[LinkedIn](#) ↗

Where are you, and what is holding you back?

Read all eight reads, then your ambition, and tick the line that fits you today. There is no score to chase: the lowest layer is the one to start with. Then map where you stand against where you want to go.

LAYER 1 · WHO DECIDES

Build the Coalition

Who is accountable for AI across the company?

- ☐ 1 No one in particular.
- ☐ 2 A committee that meets now and then.
- ☐ 3 A named owner with a mandate.
- ☐ 4 A standing coalition with budget and authority.

When AI needs a hard decision, what happens?

- ☐ 1 It stalls or gets avoided.
 - ☐ 2 It escalates and waits.
 - ☐ 3 A clear owner decides.
 - ☐ 4 The coalition decides and the business follows.
-

LAYER 2 · PROVE IT

Ambition & First Wins

What has AI actually changed so far?

- ☐ 1 Demos and pilots, nothing in production.
- ☐ 2 A few tools people sometimes use.
- ☐ 3 One real process, measurably better.
- ☐ 4 Several value chains with proven numbers.

Can you point to a business KPI that moved?

- ☐ 1 No.
- ☐ 2 We think so, but cannot show it.
- ☐ 3 Yes, for one case.
- ☐ 4 Yes, tracked and repeatable.

LAYER 3 · WHAT IT RUNS ON

AI-Ready Foundation

How ready is your data and access for AI?

- ☐ 1 Scattered and hard to reach.
- ☐ 2 Cleaned for specific projects only.
- ☐ 3 Key sources reachable and governed.
- ☐ 4 A maintained foundation agents can build on.

Where do governance, security and roles sit?

- ☐ 1 Handled case by case, late.
- ☐ 2 Raised when something breaks.
- ☐ 3 Defined for the main use cases.
- ☐ 4 A standing function, in from day one.

LAYER 4 · HOW WORK HAPPENS

Agentic Organization

How does AI change how the work is done?

- ☐ 1 It does not, yet.
- ☐ 2 People work faster, same roles.
- ☐ 3 Some processes run with agents.
- ☐ 4 Work is organized around what agents run.

What are your people moving toward?

- ☐ 1 The same tasks as before.
- ☐ 2 The same tasks, with assistants.
- ☐ 3 Directing and reviewing agents.
- ☐ 4 Setting outcomes and the judgment calls.

YOUR AMBITION

How far do you want to take this?

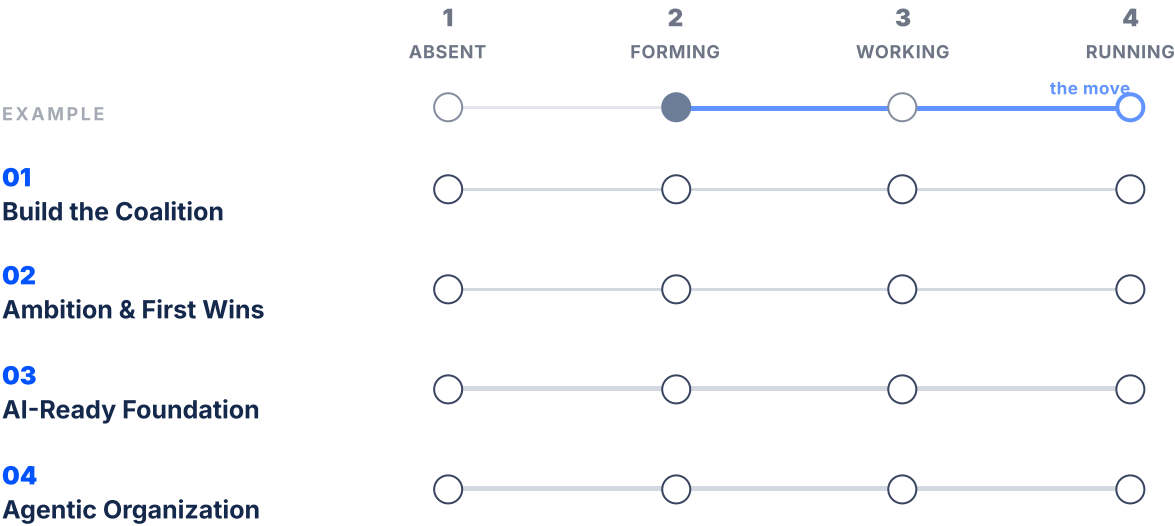
How far do you want to take this?

- ☐ 1 AI in people's hands when they need it.
- ☐ 2 An assistant in every seat.
- ☐ 3 Agents running defined work.
- ☐ 4 Run the business on agents.

Where you stand, where you want to go

Turn your answers into a position. For each layer, **average your two answer numbers (1 to 4)**; that score is a stop on its lane. Fill the dot for **today**, circle the dot for your **goal**, then draw the arrow between them. Read it two ways: your **lowest today** is the layer to start with, and the lane with the **longest arrow** is where the move matters most.

● Today · fill the dot ○ Goal · circle the dot → draw the move



My ambition, in one line:

ONE LAST THING

Which question are you carrying?

Not the answer. The question itself, the one still on your mind.
Write it down before it slips.

cloudflight | paiqo TECH ADVISORY

We live where transformations die.

We do not stop at advice. We build the layer with you.

© 2026 Cloudflight

 AI co-created