

LuAir Governance[•]

**The node,
installed.**

Seven capabilities. One working data operation.
Complete enough to do yourself.

Governance, installed.

THE SOLO PLAYBOOK • LUAIR.NET

START HERE

Solo

Solo installs one working governance operation — a node — inside your own team. Seven capabilities. A named owner on every rule. Proven on real cases in four to eight weeks. The method is free to read; the coaching is what you pay for; the node graduates — then it runs without us.

How this works

This playbook is the reference: one spread per capability. The Node Canvas is the companion: one page of blanks per capability, and its cover identifies the node — name, scope, version. Fill the canvas as you go — when it's complete, it is your node's specification. Read a spread, fill its canvas page, move on. Most capabilities take a conversation and a decision, not a project.

Yourself, or coached

Everything you need to do this yourself is here. The coached engagement adds judgment, pace, and certification: four to eight weeks at about four hours a week — roughly two to four weeks of setup, then two to four weeks of supervised live operation before graduation. Working alone, your sponsor plays the coach's part.

First node in your company? There is no orchestrator yet — your sponsor acts as one. Wherever this playbook says orchestrator, read: sponsor. In a small or first node, sponsor and orchestrator are often the same person — record the same name for both.

The seven capabilities

01 People & roles

When something goes wrong with your most important data, who handles it?

02 Intake

When someone discovers a data problem, where does it go?

03 Prioritization

When multiple data problems come up at the same time, how do you decide which one to fix first?

04 Escalation path

When a data issue is too big for whoever found it, what happens next?

05 Awareness

Do you know what your most important data is, what shape it's in, and what rules apply to it?

06 Definition of done

How do you know when a data problem is actually fixed?

07 Communication cadence

How does your team find out about data problems?

Five moves



Assess

Take the readiness assessment. Know your shape.



Plan

Pick the data that matters most. Secure a sponsor.



Build

Work the seven spreads. Fill the canvas.



Measure

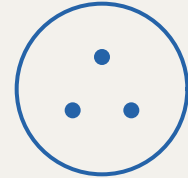
Run live, supervised, on real cases.



Graduate

Prove it. Then it's yours.

01



People & roles

When something goes wrong with your most important data, who handles it?

A node is the operation you install inside a boundary drawn on your existing org chart — around one role, one team, or a whole department — staffed by people or agents, not defined by them. Every node has someone it escalates to; that's the orchestrator, and you already know who yours is — it's whoever you'd escalate to today, whatever their title. Inside the node are the people doing the work: someone monitoring intake, people handling requests, and the people who own the rules about the data. Nobody owns a table; someone owns the rule that says what a valid record is.

WHY IT MATTERS

When something goes wrong, someone needs to own the fix — and you need to know who before the problem arrives. A role that lives in someone's head isn't a role; it becomes one when the name is written down.

CAPABILITY 01 • SET IT UP

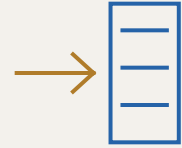
- 1** Draw your node around what you want to manage better — a team, a department, or just yourself. Start with the org structure you already have.
- 2** Write down who the node escalates to. That person (or agent) is your orchestrator — no new role needed, it's whoever the work already goes up to.
- 3** Choose an operator and a backup, and record who holds each role on the staffing record. No new hires required. Likely you have existing people who are playing this role already.
- 4** List the rules your team already enforces. Each one carries a rule number and names a decision owner.
- 5** No rule on data that matters? Write that down as a gap — a finding, not a failure.
- 6** When a question touches a rule, route it to the rule's owner. The map fills in through real work.

YOU'LL KNOW IT'S WORKING WHEN

- The node knows who it is: what's inside the circle, and who the orchestrator is
- Operator and backup are named on the staffing record — and know it
- Every listed rule carries exactly one owner's name
- A real question has been routed to a decision owner
- The backup has run the node for a week without the operator

For an agent: the roles page is its instructions — who it acts as, what it decides alone, and who it escalates to. The orchestrator can be a person or another agent; the pattern doesn't change.

02



Intake

When someone discovers a data problem, where does it go?

One intake queue. Not a perfect one — the only one. A shared list everyone uses beats a polished system nobody does. Use whatever tool your team already has.

WHY IT MATTERS

Without a single place for work to enter, problems compete with whoever shouts loudest. A queue replaces noise with a list you can manage — and you can't prioritize what you never captured.

CAPABILITY 02 • SET IT UP

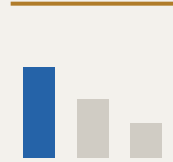
- 1** Pick the intake queue: a shared list, mailbox, or tracker your team already uses.
- 2** Log four things per item: date, who asked, one line on what, status.
- 3** Have the sponsor announce it: this is now the one place.
- 4** Sweep every known open item out of inboxes and memory into the list.
- 5** Work arrives outside the queue? Log it yourself, then point the requester to the queue.

YOU'LL KNOW IT'S WORKING WHEN

- Anyone on the team, asked where problems go, gives the same answer
- Everything open is on the list — nothing lives only in chat
- Work items from at least three different people show the queue is real

For an agent: the intake queue is where it receives work. Same queue, different operator.

03



Prioritization

When multiple data problems come up at the same time, how do you decide which one to fix first?

A rhythm, not a scoring model. On a schedule, the operator and sponsor look at the list, commit to a few items, and deliberately ignore the rest until next cycle.

WHY IT MATTERS

Without a rhythm for deciding, everything feels urgent and nothing gets finished. Committing to a few things — and saying no to the rest, out loud — is the discipline.

CAPABILITY 03 • SET IT UP

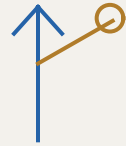
- 1** Set the meeting: who, what day, how often.
- 2** Set a ceiling: the most items you'll commit to per cycle. Keep it low.
- 3** At each review: commit, defer, done. Date the committed list.
- 4** Mid-cycle additions are the sponsor's call only — that's the exception rule.
- 5** Tell requesters what was committed and what was deferred. Silence breeds shadow queues.

YOU'LL KNOW IT'S WORKING WHEN

- The review happens on schedule, even when there's little to review
- Each cycle has a dated committed list, and it's short
- Deferred requesters heard the word deferred — from you, not from silence

For an agent: the rhythm is its scheduler; the ceiling and exception rule are its constraints.

04



Escalation path

When a data issue is too big for whoever found it, what happens next?

When something lands on the node and no rule says what to do with it, it goes up — to the orchestrator. That's the whole path. The node handles what its rules cover and escalates what they don't; the orchestrator decides — handle it, pass it on, make a new rule, or say 'we don't do this.' And every escalation is a signal: if it'll come up again, it becomes a new rule with an owner, and next time the node won't need to ask. Two nodes disagree? Same path: the dispute goes up to the first orchestrator they share, and the answer becomes a rule for both.

WHY IT MATTERS

Without a path up, anything the rules don't cover either gets guessed at or dies in the backlog. Rules answer the known; the orchestrator answers the new — and a node that escalates cleanly gets smarter with every answer.

CAPABILITY 04 • SET IT UP

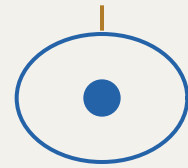
- 1** Confirm who your orchestrator is — you named them in People & roles. That's where escalations go.
- 2** Agree with the orchestrator how they want things sent up: a message, a tag on the item, a standing few minutes in the check-in.
- 3** Escalate the moment any of three things is true: no rule covers it, a rule covers it but the call is beyond the node's authority, or it has aged past its window. Don't guess, don't sit on it.
- 4** When the orchestrator answers, capture the answer on the item. If it will come up again, write it as a new rule and name its owner.
- 5** First real escalation: send it up, note the date. That's the proof.

YOU'LL KNOW IT'S WORKING WHEN

- Everyone in the node knows: no rule, no guessing — it goes to the orchestrator
- The orchestrator has agreed how escalations arrive
- A real escalation has gone up and come back answered
- At least one answer has become a new rule with a named owner

For an agent: escalation is its boundary — anything that trips a trigger (uncovered, beyond authority, or aging) gets handed up to the orchestrator, person or agent. An agent that escalates cleanly is one you can trust with everything its rules do cover.

05



Awareness

Do you know what your most important data is, what shape it's in, and what rules apply to it?

The node knows its own state — what's being worked, what's stuck and why, which data it protects and the condition that data is in. Not a report someone assembles: a property of how the node works. Awareness is also its voice: it's what the node tells its orchestrator.

WHY IT MATTERS

You cannot improve what you cannot see. Awareness means answering — at any moment, without preparing anything — what you worked on, what got resolved, what's stuck.

CAPABILITY 05 • SET IT UP

- 1** Make state visible where the work lives: every item shows its status on the list.
- 2** Mark stuck items with who they wait on. Stuck without a name isn't tracked, it's abandoned.
- 3** List your critical data: the attributes that matter, where they live, whose rules govern them.
- 4** Set a light checking rhythm for that data's condition, checked against its rules. Note what you find, and take the finding reaction.
- 5** Let the orchestrator see the same state you see — no special report, same list.

YOU'LL KNOW IT'S WORKING WHEN

- Anyone can answer 'what's stuck and why' by looking, not by asking around
- The critical data list exists, with owners against rules
- The sponsor has asked cold — and the answer came from the board, unprepared

For an agent: awareness is its self-reporting — state the agent maintains and speaks, not a dashboard bolted on.

06



Definition of done

How do you know when a data problem is actually fixed?

The node knows in advance what finished looks like for the kinds of work it supports. Those definitions of done are part of its rules — written once, owned like any rule, not invented per request. When work arrives, it gets matched to one. And if nothing matches — if nobody can say what done would look like — it isn't ready to be worked; it goes up to the orchestrator, like anything else the rules don't cover.

WHY IT MATTERS

Most governance work stalls because nobody agreed what finished looks like. When done is defined once for each kind of work, nobody negotiates completion item by item — the node just works, closes, and proves it.

- 1** List the kinds of work the node supports. A handful is plenty to start.
- 2** Write what done means for each kind. That's a rule — give it an owner.
- 3** When work arrives, match it to its definition before work starts. The item carries its completion criteria from day one.
- 4** Close only by confirming against the definition. Someone's name goes next to it.
- 5** No matching definition? It goes up to the orchestrator — don't invent done on the fly.
- 6** An item that comes back gets reopened, not re-created — and its definition gets sharpened.

YOU'LL KNOW IT'S WORKING WHEN

- Each kind of work the node supports has a written, owned definition of done
- Every open item carries its definition from the moment work starts
- Something has closed by confirmation, with a name next to it
- Something without a matching definition has gone up instead of being guessed at

For an agent: each kind of work it supports carries an Agent Work Method, and work that matches no definition triggers automatic escalation — the agent never invents its own completion criteria.

What a filled blank looks like

Every rule is one sentence with one name on it. These are for an ordinary business — orders, customers, inventory. Yours will read differently; the anatomy is what to copy.

THREE BUSINESS RULES (RULE — OWNER)

R1 A customer record is valid when it carries a billing email and one verified phone number.

OWNER: MARIA CHEN — BILLING LEAD

R2 An order may only close when its payment status matches the processor's record.

OWNER: SAM ORTIZ — OPS MANAGER

R3 Inventory counts reconcile to the warehouse system weekly; a variance over 2% opens a work item.

OWNER: DANA FIELDS — WAREHOUSE LEAD

TWO DEFINITIONS OF DONE (KIND — DEFINITION — OWNER)

Data correction request

Done = the record matches its source document, the requester has confirmed, and the fix is noted on the work item.

OWNER: SAM ORTIZ

New report request

Done = the report ran once on real data, the requester signed off on the numbers, and its refresh owner is named.

OWNER: MARIA CHEN

No rule on data that matters? Write that down as a gap — a finding, not a failure. The map fills in through real work.

07



Communication cadence

How does your team find out about data problems?

A standing rhythm that happens whether or not there's news. A short check-in for the team; a short report to the orchestrator: worked on, resolved, stuck. Awareness speaking on a schedule.

WHY IT MATTERS

Without a regular beat, stakeholders assume nothing is happening and sponsors lose confidence. The rhythm keeps the node connected to the people who depend on it — silence is how working nodes get forgotten.

CAPABILITY 07 • SET IT UP

- 1** Set the beats: who hears from the node, how often, through what channel.
- 2** Keep the report to three lines: worked on, resolved, stuck (and who it waits on).
- 3** Send it on schedule even when the answer is 'quiet week'. Especially then.
- 4** The check-in is short, chaired, and noted — decisions leave a one-line trace.

YOU'LL KNOW IT'S WORKING WHEN

- Three consecutive cycles of the report have gone out on time
- The sponsor can say when they'll next hear from the node without asking

For an agent: the cadence is its heartbeat — scheduled state reports to named humans.

Five roles, whole views

Everything each person does, on one card. If it isn't on your card, it isn't your job.

Node Operator

A FEW HOURS A WEEK

You run the node. It's a role, not a job change.

- Monitor the intake queue; everything gets logged
- Chair the cycle review; keep the committed list short and dated
- Route rule questions to their decision owners
- Escalate anything no rule covers to the orchestrator — don't guess, don't sit
- Keep state visible — status, stuck, and who it waits on
- Confirm done against its definition before closing
- Send the three-line report on schedule

Backup Operator

MINUTES, UNTIL IT MATTERS

You are continuity. The node doesn't stop when the operator is out.

- Know the intake queue and the committed list well enough to run them
- Log and triage when the operator is away
- Run the node for one full proving week before graduation
- Chair the review when the operator can't

Node Orchestrator

MINUTES PER CYCLE — MORE ONLY WHEN THINGS COME UP

The node escalates to you. You already have this job — whatever your title.

- Receive what the node's rules don't cover; decide — handle it, pass it on, make a new rule, or say 'we don't do this'
- Get the node's three-line report on rhythm; ask cold sometimes — what's stuck?
- Turn repeat escalations into new rules with named owners, so they stop arriving
- Sign graduation
- A person or an agent; you may oversee several nodes, and you may have an orchestrator of your own

Decision Owner

MINUTES, WHEN YOUR RULE IS TOUCHED

You own rules, not tables. Your name means your call.

- Confirm the rules attributed to you — correct them if wrong
- Answer questions the node routes to your rule
- Own what done means for the kinds of work your rules govern
- Change the rule when the business changes; the node records it
- That's it. No meetings, no binders.

Node Sponsor

MINUTES A WEEK — AUTHORITY, NOT LABOR

You authorize the node and stand behind it.

- Announce the intake queue: this is now the one place
- Approve mid-cycle additions — nobody else can
- Ask cold sometimes — what's stuck? The answer must come from the queue
- Sign graduation when the proofs are real
- First node in the company? You also act as its orchestrator
- Working without a coach? You play the coach's part

If the operator is an agent

Five rules that exist only because the operator is software. Everything else in this playbook applies to an agent unchanged.

- 1** The agent keeps an action log it cannot edit. Every reply, every escalation, every close — written where the orchestrator can read it, not where the agent can rewrite it.
- 2** A wrong answer already sent gets a correction to the same person, and the mistake is logged as a work item — not erased.
- 3** The orchestrator can suspend the agent at any time. The backup operator takes the queue; the node never stops with the operator.
- 4** A change to the agent's model or instructions is a governed change: the node re-runs its supervised window before the agent runs alone again.
- 5** Every kind of work the agent supports has its steps written as an Agent Work Method; the agent works only from written methods.

Graduation

Graduation is one page. Each capability answers one question, live, on real work — then the node is yours.

- 01 A question found its owner without a meeting
- 02 Work arrived in one queue, from more than one person
- 03 A cycle committed little and finished it
- 04 Something escalated and landed with a name
- 05 'What's stuck?' was answered by looking
- 06 Something closed against its written exit condition
- 07 Three reports went out on rhythm, one of them quiet

Supervised weeks complete. Real cases handled. Sponsor's signature — or your coach's certification. The node is live. That's Solo: one node, proven, yours.

Coached install, or questions: contact@luair.net · luair.net

Where this leads

Know your shape first

The readiness assessment — seven questions, a few minutes: luair.net/assessment.html

The worked use case

Node Zero: a complete example of a node staffed by an AI agent — and new use cases and method updates as they're published: luair.net/node-zero.html

The companion canvas

The blanks that matter — free at luair.net/solo.html

Questions, or a coached install

contact@luair.net · luair.net

Governance, installed.

Glossary

The vocabulary of the method in one place. These definitions are the reference for every page that uses the terms.

Agent Upgrade

the graduation of a Role node's operator role to an agent that runs under the same rules and a named Node Orchestrator.

Agent Work Method

a Work Method written for a kind of work an agent performs, stated in three parts: the role that acts, the goal of the work, and the steps in order.

Backbone Capability

one of the seven capabilities every node installs.

Backup Operator

the continuity role: runs the node when the operator is out, and for one full proving week before graduation.

Business Rule

a written statement, with one owner, that governs how the node's data or work is handled.

Cycle Review

the recurring review where open work is examined and the next cycle's commitments are chosen; mid-cycle additions are the sponsor's call.

Decision Owner

the role that owns a business rule; their name means their call on that rule.

Definition of Done

the written completion criteria for a kind of work, agreed before the work starts.

Employee File

the written specification of one role, kept independent of whoever holds it; in this method, the filled Node Canvas.

GLOSSARY

Escalation Path

the defined route from the node to its orchestrator for anything no rule covers, beyond the node's authority, or aged past its window.

Finding Reaction

the action, written in advance with one owner, that the node takes when a check shows its critical data does not meet its business rule.

Intake Queue

the single place where work is submitted to the node and logged as work items.

Node

the operation you install inside a boundary drawn on your existing org chart — around one role, one team, or a whole department — staffed by people or agents, not defined by them.

Node Authority

what the node handles on its own; anything beyond it escalates.

Node Awareness

the node's ability to state its own condition — open, resolved, and stuck — by looking, without preparing a report.

Node Canvas

the form that documents one node: seven capability pages plus cover, graduation, and the staffing record.

Node Graduation

the point where the seven proofs are witnessed on real work and the node runs without coaching.

Node Operator

the role that runs the node: monitors the intake queue, keeps state visible, escalates what no rule covers, and sends the report.

Node Orchestrator

the role the node escalates to; decides what the rules do not cover, and turns repeat escalations into new rules.

GLOSSARY

Node Sponsor

the role that authorizes the node, announces the intake queue, approves mid-cycle additions, and signs graduation.

Qualification

a written requirement a candidate, a person or an agent, must meet to hold a role in the node; for an agent, the model and the effort level it runs at.

Readiness Shape

the assessment result: each capability rated absent, emerging, or functioning.

Rule Number

the short unique name a business rule carries within its node, given once and never reused.

Rule Wording

the requirement that every business rule is one declarative sentence built from defined terms.

Shape Report

the written form of a Readiness Shape, produced for the person who completed the assessment.

Staffing Record

the page of the Node Canvas that states for each role in the node who holds the role now, since what date, and the qualifications met.

Three-Line Report

the recurring report: worked on, resolved, stuck — with who each stuck item waits on.

Work Item

one logged unit of work: date, who asked, one line on what, and status.

Work Method

the written steps for performing a kind of work; a person is trained on them, and an agent executes them as its instructions.