THE HANDOFF

THE RIGHT ALTITUDE · FLIGHT MANUAL SERIES

Delegation to Reese, my deputy.

Prepared for Jordan Hale

June 1, 2026

This delegation plan is based on the leader's self-assessment and observations. It is a preparation tool, not a performance evaluation.

THE READ

This isn't a readiness problem — it's an identity problem, and you already know it. You've told me the real reason: Atlas has been your proof that you still deliver. That's the diagnosis. The hard part of this handoff isn't Reese's capability; you've rated them ready, high-trust, high-autonomy, owns_it. The hard part is that letting go of Atlas means letting go of the evidence you point to when you ask yourself whether you're still valuable. So watch for this: every instinct to 'just check one thing' will dress itself up as quality control or stakes-management, when it's actually you reaching back for the proof.

Reese's readiness looks like someone who can run the final phase cleanly if you stop being the gravity well. Your own pattern is the risk, not theirs. You've named it precisely — your control default is takes_back, when work is different you redo it, and the hardest part for you is watching. That triad is lethal in combination. Reese does it differently (your stated 'what typically goes wrong'), you experience the difference as wrongness, you redo or reclaim, and Reese learns that ownership was never real. With a high-autonomy report, that signal lands hard and fast — they'll stop bringing you their best thinking and start managing you instead of the project.

What's at risk if you do this badly: not the public wobble you're afraid of, but the quieter failure where Atlas ships fine but Reese never actually owned it because you hovered through the final phase. You'd get the result and lose the leader you're trying to build — and you'd have taught your strongest deputy that your delegation is conditional. The client-facing stakes are real, but they're more protected by a clean handoff than by a shadow co-pilot. A confused dual-ownership in the final phase is exactly what produces the public wobble you dread.

HOW TO FRAME IT

Have this conversation once, deliberately, and make it unambiguous. Start with why them, and be specific: 'You've watched Atlas from the inside long enough to make the calls I'd make — and in some cases better ones. I'm handing you the final client-facing phase and everything after, with full ownership.' For a high-autonomy report, the worst framing is a conditional one. Don't say 'let's run it together for a bit.' Say the word 'yours' and mean it. Reese will calibrate their entire posture to whether you sound like you're transferring or lending.

Then address the thing they need most — air cover — directly and unprompted, because it's the currency that makes high-stakes ownership possible. Say it plainly: 'This touches a client release and my name is on the eighteen months behind it. That means if something goes sideways publicly, I'm the one who stands in front of it — not you. You make the calls; I absorb the blast. Your job is to run it well, not to protect me.' That sentence does two things: it frees Reese to make bold calls, and it tells them you won't quietly disown them if it gets hard.

Finally, name why now in a way that elevates rather than diminishes. Your stated reason is that the work is below your level now — say a version of that without making Atlas sound like a hand-me-down. Try: 'I've been running this past the point where I should — partly because I'm proud of it. Handing it to you now, at the highest-stakes phase, is me betting on you when it actually counts.' That's honest about your own attachment, which builds trust, and it frames the timing as confidence, not offloading.

WHAT AUTHORITY TO TRANSFER

Transfer the decision rights explicitly, item by item, because ambiguity here is what your takes_back default will exploit. Reese can make, without checking you: all phasing and sequencing calls for the final release, all internal team direction, all trade-off decisions on scope-versus-timeline within the committed release window, and all the day-to-day client-facing communication. Say the words: 'You don't run these by me. You decide, you tell me what you decided if it's notable, and we move on.' Write that list down and send it to them — a documented authority transfer is the antidote to a leader who redoes work he experiences as 'different.'

Be explicit that 'different from how I'd do it' is now in-bounds, not a trigger for review. This is your single biggest risk given that 'different_approach' is your stated failure mode. Tell Reese directly: 'You'll make calls I wouldn't have made. As long as it ships well and the client is served, your judgment stands — even when it's not my judgment.' Then hold yourself to it. The moment you start substituting your approach for theirs on a decision that's now legitimately theirs, you've revoked the authority you just granted, and a high-autonomy report will feel it instantly.

What stays with you is narrow and should be named as such so it doesn't expand: you stay the owner of the external blast radius (the air cover), and you stay a consulted party — not an approver — on anything that changes the client's contractual commitments or moves the release date publicly. 'Consulted, not approver' is the precise distinction. Reese brings you in for those because of the institutional and reputational weight, not because they need permission. Everything else is theirs.

THE GUARDRAILS

Keep the hard lines few and truly hard, because every fake guardrail you add is a leash that signals distrust to a high-autonomy report. There are really only three. One: no public commitment that changes the client's expectations on date or scope without a heads-up to you first — not for approval, but so you're never blindsided in front of someone whose call you might catch. Two: the quality bar for anything client-facing stays at the standard Atlas has set for eighteen months — Reese knows what that bar is because they've lived it; you're affirming it, not redefining it. Three: if something breaks in a way that could become a public incident, you find out from Reese before you find out from anyone else.

Communicate these as the boundaries of a sandbox, not as conditions of probation. The framing matters: 'These three are the only things I need to know early on. Everything inside them is yours to run.' That tells Reese the guardrails are small and the freedom is large — which is exactly the truth and exactly what frees up ownership. If you find yourself wanting to add a fourth and fifth guardrail, stop and ask whether it's about the project or about your own discomfort with watching.

The trap with guardrails for someone with your defaults is converting them into checkpoints. A guardrail is a line Reese must not cross without flagging; it is not a reason for you to inspect everything in the vicinity of that line. Don't let 'don't move the release date without telling me' quietly become 'show me all your timeline planning.' The first protects the stakes; the second reclaims the work. Notice the slide when it starts.

CHECKPOINT PLAN

Set a light, predictable rhythm and let Reese drive its content — high-autonomy, asks_well, owns_it people thrive on a cadence they control. A biweekly thirty-minute check-in through the final phase is right: frequent enough that you're not anxious, infrequent enough that you're not hovering. Make it Reese's agenda, not yours. The opening question is always theirs to answer first: 'What do you want me to know, and where, if anywhere, do you want me?' That structure respects their autonomy and keeps you out of inspection mode.

Your questions in those sessions should be about decisions and risks, never about methods. Ask: 'What are the calls you've made since we last talked?' and 'What's the one thing that could turn into a public problem, and how are you holding it?' Do not ask 'walk me through how you're doing X' — that's the redo instinct in reconnaissance form. Because Reese asks well when stuck, you can trust that real questions will surface; you don't need to fish for them. Between checkpoints, default to silence unless they pull you in or you hit one of the three guardrails.

Build in an explicit taper, because your success metric is that six months out you've forgotten the daily detail. Say it on day one: 'We'll do biweekly through ship, then I'm stepping out of the cadence entirely — you'll bring me in if you want me, and otherwise it's just yours.' Naming the taper protects you from your own takes_back gravity and tells Reese the off-ramp is real, not theoretical. If you're still in a standing Atlas meeting six months from now, the delegation failed regardless of how the release went.

WHAT NOT TO DO

Don't redo or quietly correct work that's merely different. This is your named failure mode and it will arrive disguised as standards. When you see Reese make a call you wouldn't have made, your hand will move toward fixing it — and the instant you do, you've told a high-autonomy deputy that their ownership was a loan. The signal to Reese isn't 'Jordan has high standards'; it's 'Jordan doesn't actually trust me, and I should run things the way he'd run them.' That kills exactly the independent judgment you said success requires.

Don't insert yourself as the escalation path — your success definition explicitly says you must not be it. Your takes_back default will tempt you to be 'available' to the client or the team for the hard calls, framing it as support. It isn't support; it's a back door to reclaiming. When someone routes a hard decision to you, route it back: 'That's Reese's call now — go to them.' Every time you take one of those, you teach the whole system that you're still the real owner and Reese is the front.

And don't hover, even silently. You've told me watching is the hardest part for you, which means your discomfort will look for outlets — extra 'how's it going' pings, lurking in channels, asking the team for color. A high-autonomy report reads ambient surveillance instantly and starts performing for you instead of running the project. If watching is unbearable, that's information about you to work on privately, not a license to monitor. Sit with the discomfort; don't export it onto Reese.

IF IT COMES BACK TO YOU

Reese asks well, so when they bring you something, treat it as a real signal — not as an opening to take the work back. First, diagnose what kind of return it is. Are they bringing you a guardrail item that legitimately needs your air cover or your consult? Then engage fully on that narrow thing and only that thing. Are they bringing you a decision that's actually theirs, looking for permission or for you to make it? Then send it back, warmly and clearly: 'You don't need me for this — what's your read?' Make them say their answer out loud, then back it.

When you send it back, give them something to own the next time, not just a deflection. After they state their call, say 'That's right, go,' or if you truly disagree, ask 'what would change your mind?' rather than substituting your decision. The goal is to strengthen their judgment, not to outsource yours one decision at a time. Because they ask well, the questions they bring will usually be real forks — honor that by coaching the thinking, not supplying the answer.

The one thing you must not do is solve it cleanly because solving it feels good and proves you've still got it. That's the identity trap operating in real time — Atlas as evidence. Notice the pull, name it to yourself, and redirect the energy into making Reese sharper. A returned problem is an opportunity to build the owner; a problem you reclaim is just you feeding your own need to be needed.

IF THEY STRUGGLE

Reese owns_it and asks well, so real struggle will more likely show up as over-functioning quietly than as a dropped ball — someone who owns it tends to grind before they reach out. Watch for the tells: longer gaps before responses, a notably thinner check-in, hedged language about a part of the project they used to speak about confidently, or a client signal that doesn't match what Reese is telling you. Those are the real-trouble markers. A messy-looking middle, decisions you'd have made differently, a bumpy week — that's the normal learning curve of someone taking the wheel for the first time. Don't confuse the two.

If you see real-trouble tells, intervene at the level of support, not rescue. The move is to offer a door, not to walk through it for them: 'I'm noticing the client thread went quiet — is that a you-want-to-talk-it-through thing, or a you've-got-it thing?' That gives Reese a clean way to pull you in without you having seized anything. Because they ask well, naming what you observe usually frees up the real ask. Resist the urge to leap to the fix even when you can see it — name the observation, then wait.

The line between support and rescue is who holds the pen afterward. Support means you help Reese think, and they make the call and execute it. Rescue means you take the call, the comms, or the client relationship. If you ever find yourself drafting the client message or getting on the call instead of Reese, you've crossed into rescue and you're reclaiming under the cover of help. The only legitimate full intervention is the real public-incident scenario your guardrails describe — and even then, you lead the air cover while Reese stays the operator.

WHERE TO LET GO

Here's the hard truth: you have to let go of Atlas as evidence that you still deliver. That's the actual thing — not the project, the proof. As long as Atlas is your proof, you cannot fully hand it over, because handing it over feels like surrendering the answer to a question you're still asking yourself. So separate them deliberately. Your value as a leader at this stage is no longer measured by what you personally ship; it's measured by whether the people you develop can ship without you. Reese running Atlas cleanly is your new evidence — better evidence, because it scales and the old kind doesn't.

You also have to let go of your version of the quality standard as the only valid one. You've named that Reese will do it differently and that you tend to experience different as wrong and redo it. But 'the way Jordan would have done it' is not the standard anymore — 'shipped well and the client served' is the standard, and there are many roads to it. Holding your specific approach as the bar isn't rigor; it's control wearing rigor's clothes. The discipline is to judge outcomes, not methods.

Understand why holding on is more damaging than an imperfect result. An imperfect Atlas under Reese's ownership teaches them to own, builds a leader who outlasts this project, and frees you for work only you can do. A perfect Atlas that you secretly co-piloted teaches Reese that your trust is conditional, keeps you anchored to work below your level, and leaves your real fear — that you only matter when you're carrying something — completely untouched. The wobble you're afraid of is far more likely to come from a confused dual-ownership than from a clean transfer to someone who's ready. Letting go is the safer bet, not the brave one.

THE ONE SENTENCE

Say this: 'Atlas is yours now — you make the calls, you ship it, and if it goes sideways in public, I'm the one who stands in front of it, not you.'

It matters because it transfers all three things at once: the ownership ('yours now'), the authority ('you make the calls'), and the air cover ('I stand in front of it') — and that third clause is the one that makes the first two real. Reese needs air cover most, and a high-autonomy owner can only run boldly through a high-stakes client phase if they know you'll absorb the blast rather than disown them when it's hard. The sentence also quietly handcuffs your own worst instinct: by publicly making yourself the shield rather than the operator, you've defined your role as the thing that lets Reese run free — which is the opposite of the takes_back, redo, hover pattern you'd otherwise slide into. Once you've said it, your job is to never contradict it with your behavior.

> *Delegation is not about getting things off your plate. It's about building the plate someone else can carry. Let go of the method. Own the outcome. Trust the person.*

— Don