Vamshi Jandhyala

The authority surface

✠

August 2026

Agent harness design, part II. What an agent may do alone is a product decision, not a setting.

When Claude Code is about to run a command on your machine, it stops and asks. Allow it once, allow it always, or reject it. Most of the time you glance and approve without much thought, and that casual yes is doing more than it looks. It is the whole answer to the question of what the agent is allowed to do. The agent does not decide. You do, in the half second before you click.

Move the same agent into production and that half second is gone. No one is sitting in front of it to approve the command, and it may be acting for thousands of people at once. The question has not disappeared. It has only lost the person who used to answer it. So what the agent may do on its own can no longer be settled in the moment. It has to be settled in advance, and not once for the whole agent but action by action.

Both ends of the dial are wrong

The obvious shape for that decision is a single dial. At one end the agent asks before it does anything; at the other it runs free. Both ends are wrong in the same way, because both pretend that all of an agent's actions carry the same risk.

Full autonomy treats reading a record and deleting one as the same act. The first time an agent acting alone touches the wrong account, you have not had a quality problem, you have had an incident. Asking before everything fails more quietly. A person required to approve every step soon stops reading the steps. Approval becomes a reflex, the reflex becomes rubber-stamping, and you are left with the look of oversight and none of its substance. The supervision did not get safer by being constant. It got worse, the way a smoke alarm that goes off every time you cook ends up taped over. That inversion, which Lisanne Bainbridge set out for industrial control rooms in 1983, is the subject of *Ironies of AI automation*, and it is why "just ask a human" is never free.

The axis is consequence, not difficulty

If a single dial is wrong, the real question is which actions sit where, and the instinct is to sort them by how hard they are. Difficulty is the wrong axis. What matters first is the cost of being wrong, and a useful minimum is three questions: how reversible the action is, how far its effects reach, and how bad the worst case is.

Take an agent that handles internal IT and access requests. Granting a developer read-only access to a test system holding only synthetic data, for the afternoon, is low on all three counts. It expires on its own, it touches nothing real, and the worst case is an afternoon of wasted curiosity. Granting that same developer standing administrator rights on a production database is high on all three. It is hard to walk back, it reaches everything behind that database, and the worst case is severe. For the agent these are almost the same call, one API away from each other and equally easy to perform. What separates them is only what happens if the agent is wrong, and that is what should decide whether it acts alone, not how confident the model happens to sound.

What falls out of those three questions is not a dial but a classification. Some actions the agent may take on its own. Some it may propose but not perform until a person approves. Some it may never take, whatever it concludes. Nor is the band always a property of the action alone: the same transfer can sit in different bands at fifty pounds and fifty thousand, or to a payee the customer has used for years and one seen for the first time. That map, every action the agent can reach with its band written against it, is the agent's authority surface. A surface rather than a setting, because it has shape: broad where the actions are cheap and reversible, pinched where they are not. Drawing that map, deciding which action lands in which band, is most of the work, and only someone who understands the domain can do it, because the cost of being wrong lives in the business and not in the model's weights. No default ships it. It is a product decision wearing the costume of a config file.

What the tools already ship

The tools people already use ship versions of this idea, which is a fair sign the gradient is real and not an enterprise nicety. Claude Code offers not a switch but a set of permission modes: one that prompts for anything beyond reading, one that runs only pre-approved tools and refuses the rest, one where a separate classifier model judges each action against standing lists of what is allowed and what is blocked, and one that stops asking almost entirely, reached through a flag whose name carries the word "dangerously".[1] When OpenAI introduced Operator, its computer-using agent, in January 2025, the agent would navigate a website on its own but pause for the user's approval before a consequential step such as placing an order or sending an email, and hand control back to the person to enter a password or a card number.[2] That is the same alone-or-approve split, drawn around a different set of actions. The flags and the product names will change with every release, and Operator's were folded into another product inside the year. The decision they encode, where the line between alone and not-alone falls, will not.

[1] Anthropic, "Choose a permission mode," *Claude Code documentation*, code.claude.com/docs/en/permission-modes (as of 5 August 2026). Sets out the permission modes, from prompting on everything but reads, through a mode that runs only pre-approved tools, through a classifier-reviewed mode with standing lists of what is blocked and allowed by default, to the bypass reached with `--dangerously-skip-permissions`. Even the bypass keeps a floor: explicit ask rules still prompt, as does a circuit breaker on deletions aimed at the filesystem root or home directory. The particular modes on offer had already changed between this essay's drafting and its publication.

[2] OpenAI, "Introducing Operator," 23 January 2025, openai.com/index/introducing-operator. Sets out Operator's confirmation step before consequential actions and its takeover mode for entering credentials. OpenAI folded Operator into ChatGPT's agent mode in July 2025.

How it asks is part of the design

Where the agent does stop to ask, how it asks matters as much as whether it asks. An approval request is a small interface under a hard constraint: the person has a few seconds and plenty else competing for them. "Approve this action?" over a yes and a no trains exactly the blind yes you were trying to avoid. A request that says what the action is, who it affects, why the agent wants to take it, and what will happen if it goes ahead gives the person something they can actually judge. The gap between those two requests is the gap between oversight and theatre, and it is purely a matter of design.

You can lower the cost of being wrong

The map is not fixed either. An action sits in the approve band because of its cost of being wrong, and that cost is something you can lower. Make a refund reversible within a window and you have moved it on the first count; cap how many the agent can issue in an hour and you have moved it on the second. A payment that needed a person can become one the agent handles alone, once the controls around it have done enough work to earn that. Reducing the stakes of an action is how you earn it more autonomy, rather than simply hoping the model is right, and it is most of what a later part of this series is about. Autonomy is engineered toward, not switched on because the demo went well.

The surface has to be predictable

The map needs one more property: that the people relying on it can predict it. An operator should be able to say, before a run, what the agent can and cannot do without them, and be right. The agent to worry about is not the one with narrow authority or the one with broad authority. It is the one whose authority is a surprise, that turns out to have been able to do something no one realised until it did. A gradient you cannot inspect is not a control, only a hope.

All of this is a decision you make before the agent has done anything to earn it: how much it may do alone while it is still a stranger. On your laptop you make that decision a hundred times a day without noticing, one click at a time. In production you make it once, on purpose, written down, for each kind of action, and then you live with it at volume. The agent that dazzles in the demo is the one where you were quietly the authority surface yourself. The agent that survives in production is the one where you remembered to build it.