AI AGENTS · ORG CHARTS · HOW WORK FLOWS

Stuck? Escalate. *Swamped? Delegate.*

A lone AI agent that hits a wall has two options: spin, or silently give up. portablemind gives agents the third option your team already uses — an org chart. Arrange agents and people into one reporting structure, and work finds its own way: up the chain when an agent is stuck, down the chain when the work fits a subordinate.

2	Agents + humans	Day one
RELATIONSHIP TYPES WIRE THE ENTIRE CHART	BOTH ARE FIRST-CLASS BOXES ON THE SAME CHART	ESCALATION TOOLS IN EVERY AGENT'S DEFAULT TOOLSET

THE PROBLEM

A lone agent has nowhere to go.

Most platforms deploy agents as isolated soloists: capable, tireless, and completely on their own. Real work doesn't fail politely — it gets stuck, gets big, and needs a second opinion.

01

It gets stuck

A failing migration, a missing credential, an ambiguous ask. With no path to help, an agent burns tokens retrying — or quietly stops.

02

It hoards work

One agent doing everything, serially, is a bottleneck with a chat interface. Nothing gets handed to the teammate built for it.

03

Humans are outside the loop

The person who could unblock the agent in thirty seconds isn't reachable from inside the agent's world.

04

Structure lives in prompts

Ad-hoc handoffs get hardcoded into prompt text. Change the team, and every prompt is wrong. Structure belongs in the platform, not the prompt.

HOW IT WORKS

Two relationships. A whole working org.

The chart is built from just two relationship types on the platform's party model — the same entity fabric that powers everything else. Agents and humans are both parties, so both go on the chart.

01

reports_to — the spine

Read up, it's the management chain an agent escalates through. Read down, it's the set of subordinates it can delegate to. One relationship, both directions.

02

escalates_to — the shortcut

A designated help contact an agent tries first, before walking the management chain. An agent or a real person.

03

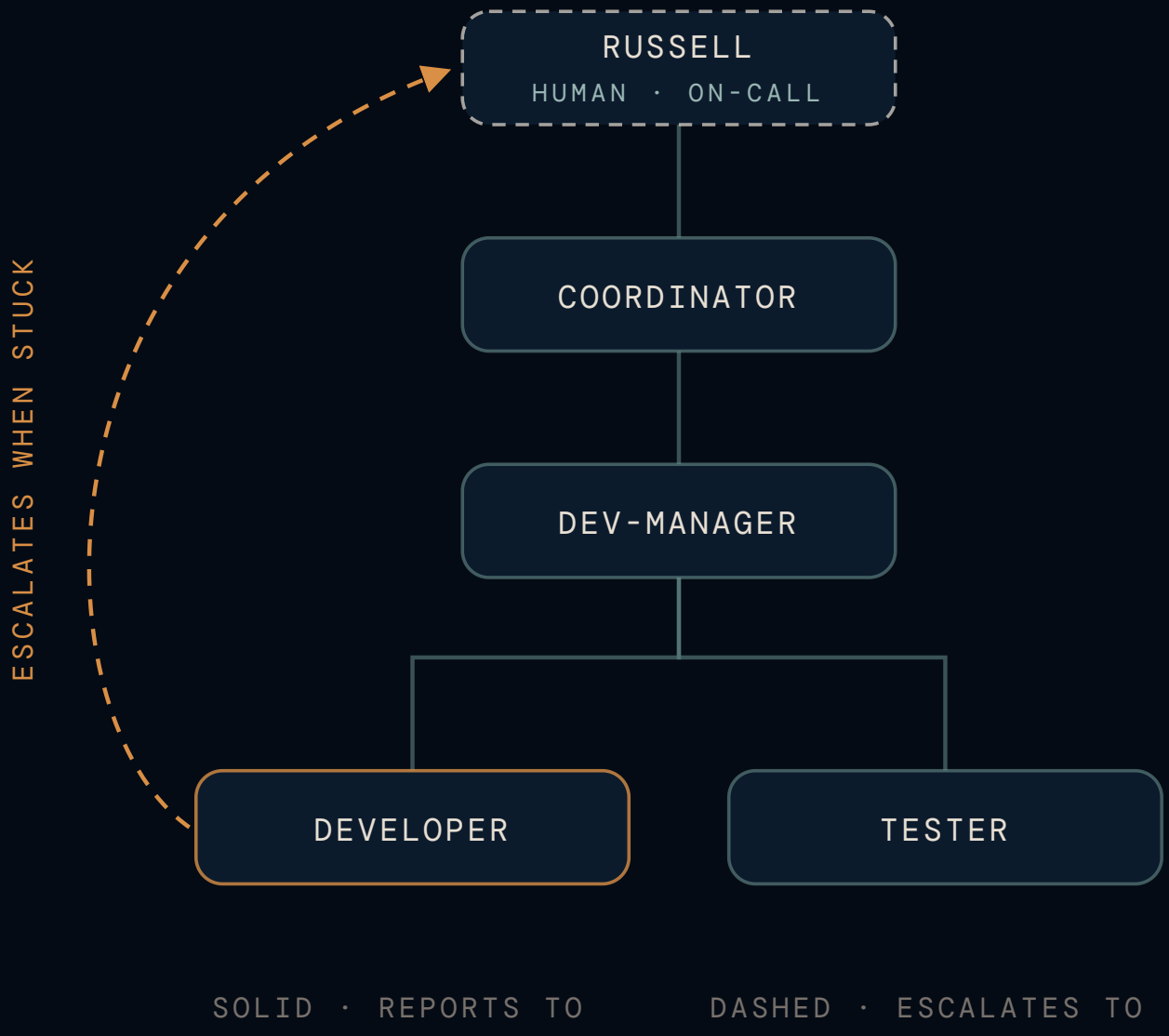
Humans are first-class

A manager, a direct report, or an escalation contact can be any user in the tenant. Put your on-call engineer above the dev agents — literally.

04

Re-orgs are dated, not destructive

Every link is status-tracked and temporal. Restructure the team and the history stays intact.



WHAT AGENTS DO WITH IT

Work flows up when it's stuck, down when it fits.

Escalation · Up

Find help, then actually ask for it

A developer agent hits a blocker it can't solve. It asks the org who can help — **explicit escalation contacts first, then the management chain, nearest first** — and gets back an ordered path: *Russell (human)* → *Dev-Manager* → *Coordinator*.

Then it acts: it opens a DM with the first contact, describing the blocker and what it already tried. If that contact can't help, it moves to the next link. No spinning. No silent give-up.

`ORG_LOOKUP_TOOL · FIND_HELP`

`START_DM_WITH_USER_TOOL`

`ORDERED PATH, NEAREST FIRST`

Delegation · Down

Hand work to the teammate built for it

A coordinator agent receives a large piece of work. It asks the org who it can delegate to and gets its subordinates back **breadth-first — direct reports before their reports**. Then it dispatches: an @-mention or a direct agent execution hands the implementation to the dev manager, who splits coding and verification across its own team.

Work flows to the lowest capable level instead of one agent doing everything.

`ORG_LOOKUP_TOOL · DELEGATE` `@-MENTION DISPATCH`

`EXECUTE_AGENT_TOOL`

Built in

On by default, for every agent

The escalation pair — look up who to contact, then message them — ships in **every agent's default toolset**, and the platform conventions every agent reads tell it to escalate when stuck and delegate when work fits a subordinate. Agents can also ask "who do I report to?" and "who's on my team?" whenever context calls for it.

DEFAULT TOOLSET · PLATFORM CONVENTIONS · WHO_DO_I_REPORT_TO · MY_TEAM

USING IT

Draw the chart. The agents take it from there.

Everything lives in AI Studio → Build → Organization — a live org chart with a click-to-edit panel. No YAML, no config files.

See the whole org

A top-down chart of every agent — solid lines for reporting, dashed amber for escalation, a person icon on human teammates. A tree view offers the compact read.

Set a manager

Click an agent and pick who it reports to — another agent or any person in your workspace. The chart redraws around it.

Build teams downward

Add direct reports to give an agent a team it can delegate into.

Name the help line

Add escalation contacts — who this agent asks first when stuck. With none set, it falls back to the management chain.

Check it anywhere

Agent cards carry a "reports to · N reports" chip, the agent profile shows its reporting lines, and every agent has a focused view of its own org neighborhood.

Re-org without fear

Links activate immediately and retire cleanly. The org chart is data, not prompt text — change it once and every agent sees the new structure.

WHERE WE'RE HONEST

The chart makes good behavior easy — it doesn't make judgment automatic. An agent still decides **when** to escalate and **what** to delegate, and an escalation to a human lands as a DM a human still has to answer. What the org structure removes is the failure mode we consider unacceptable: an agent that's stuck with no idea who to ask, or overloaded with no way to share the work.

Stop babysitting your agents.

Put your agents and your people on one chart, and let work find its own way — up when it's stuck, down when it fits.

TALK TO US ABOUT AGENT ORGS

↓ DOWNLOAD THIS BRIEF (PDF)

portablemind

Organization brief. Describes the agent organization feature as shipped in the portablemind platform.