WUNDERCORP RESEARCH

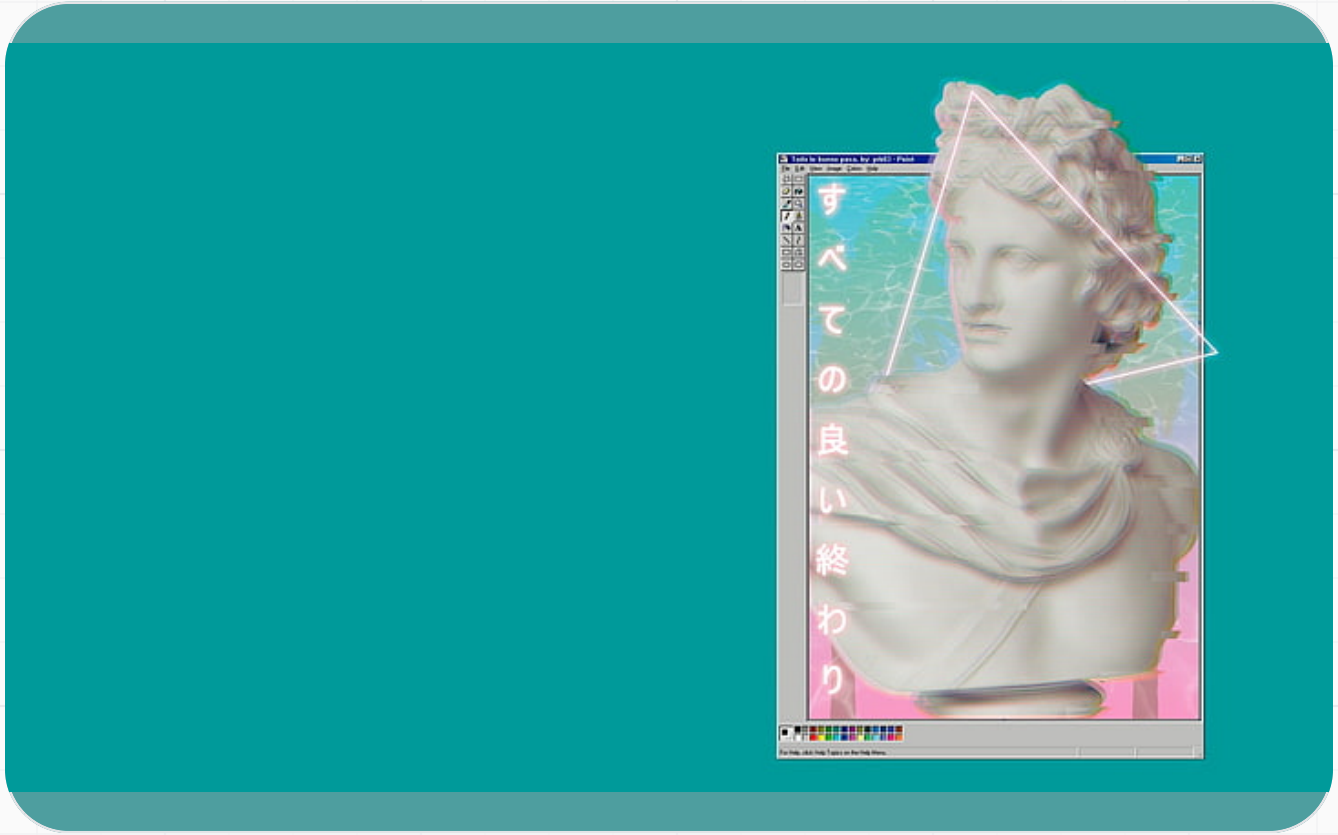
Agentic Swarming

Turn one ambitious goal into a coordinated team of AI specialists.

BuilderStudio plans the work, asks for approval, and runs each lane in contained Hermes environments so teams can move faster without giving up control.

WHY IT MATTERS

Instead of forcing one assistant to do everything, the swarm splits complex work into focused lanes across product, marketing, outreach, ops, QA, and security.



From one goal to many parallel deliverables, with visible outputs and a clear approval step before execution.

What agentic swarming means

Start with one outcome. The system plans the lanes, runs them in parallel, and brings back evidence you can review.

01

One goal

The user starts with a real outcome, like launching a product, auditing a workspace, or preparing a campaign.

02

Many lanes

The system breaks that outcome into specialist lanes, each with a role, a job to do, and a place to put its work.

03

One place to review

Instead of one blended answer, you get lane-by-lane outputs, commands, diagnostics, transcripts, and changed files.

SIMPLE PITCH

Agentic swarming helps teams tackle cross-functional work without juggling separate prompts or stitching together disconnected outputs by hand.

Why this matters

Complex work rarely lives in one lane. Launches, audits, and operational projects usually span multiple teams.

Faster end-to-end execution

Multiple workstreams move at the same time, so teams spend less time waiting for one long run to finish.

Better specialization

Each lane can stay focused on its own job, whether that means building, messaging, testing, operational planning, or security review.

Clearer accountability

Outputs stay separated by lane, which makes it easier to inspect what happened, what changed, and what still needs review.

MENTAL MODEL

Traditional AI assistant

One prompt -> one agent -> one long answer or one workspace run.

->

Agentic swarm

One goal -> planned scenario -> many contained lanes -> reviewable outputs.

Audience benefit

Teams can move across product, marketing, ops, QA, and security in parallel without losing the approval checkpoint that keeps high-risk work under control.

This is coordinated AI work you can inspect, approve, and act on, not a black box you have to trust blindly.

The user journey

From slash command to parallel output, the workflow stays simple, visible, and approval-led.

1 Choose a swarm

Type /swarm, /swarm demo, /swarm list, or a scenario command such as /swarm security-audit.

2 Review the plan

BuilderStudio loads scenarios, selects the goal, and estimates the model route, token use, and expected spend.

3 Approve execution

The user approves workspace and high-risk executor permissions before any swarm starts.

4 Watch lanes run

One terminal pane opens per lane so the user can see progress, status, and heartbeat while the daemon runs.

5 Inspect results

Lane outputs return as changed files, commands, diagnostics, transcript tails, and runtime URLs.

WHAT TO REMEMBER

Users do not have to coordinate six separate agents by hand. The scenario defines the lanes, BuilderStudio prepares the run, and Hermes returns evidence that makes review straightforward.

How the system is structured

The product plans and visualizes the swarm. The daemon handles execution and isolation.

01

Command UI

Slash commands and scenario picker

02

Planner

Goal, scenario, lanes, model route

03

Approval

Cost, permissions, executor consent

04

Daemon API

GET /swarm/scenarios, POST /swarm/run

05

Hermes lanes

Contained parallel agent jobs

UI

Frontend responsibilities

Recognize /swarm commands
Fetch and list scenarios
Select goal and lane count
Build the model and cost estimate
Request workspace and executor approval
Open one terminal pane per lane
Render returned diagnostics and artifacts

RUNTIME

Daemon responsibilities

Enforce the swarm contract
Create session and lane folders
Run Hermes container-only agents
Block host home, SSH, shell, and installers
Return lane summaries and evidence

OUTPUT

Response contract

The response returns the swarm session directory, lane status, changed files, terminal commands, diagnostics, transcript tails, and runtime URLs when available.

Safety posture

The swarm is designed to be useful without being reckless.

ALLOWED

- Contained Hermes lanes
- Target-workspace-only mounts
- Explicit user approval before execution
- Per-lane evidence and diagnostics
- Scenario-defined roles and objectives
- Explicit-env-only secrets mode

BLOCKED BY DEFAULT

- x Host shell access
- x Host installs
- x Remote shell installers
- x Host home directory mounts
- x Host SSH directory mounts
- x Implicit secret discovery

The user stays in control of the decision to run, and the runtime stays strict about what each lane can reach.

Example: the launch-stack swarm

The default demo shows how one product-launch goal becomes parallel specialist workstreams.

User goal

Prepare a complete launch package for a BuilderStudio-style product: website, messaging, outreach, launch operations, QA, and security readiness.

Typical lane outputs

- **Product site**
  Landing page, positioning, conversion copy
- **Marketing**
  Messaging brief, launch copy, social assets
- **Outreach**
  Email templates, partner notes, sales angles
- **Launch ops**
  Checklist, timeline, owner-ready tasks
- **QA**
  Release plan, regression notes, risk list
- **Security**
  Compliance readiness and hardening review

What viewers see

```
$ /swarm demo
Planning: launch-stack scenario
Estimate: 6 lanes, model route selected
Approval required: contained Hermes swarm
Lane 001 product-site ........ running
Lane 002 marketing ........... running
Lane 003 outreach ............ running
Lane 004 launch-ops .......... running
Lane 005 qa-release .......... running
Lane 006 security-readiness .. running
Results: changed files, diagnostics, transcript tails
```

The demo makes the value concrete: one command kicks off a temporary launch team and brings back work you can actually review.

Where swarms fit

The best use cases share one trait: they naturally break into multiple workstreams.

01

Product launch

Website, messaging, outreach, launch operations, QA release planning, and security review.

02

SaaS MVP

Parallel lanes for app skeleton, onboarding, pricing, docs, tests, and deployment readiness.

03

Marketing launch

Positioning, campaigns, creative briefs, email sequences, social posts, and measurement plans.

04

Security audit

Dependency review, auth checks, privacy readiness, compliance notes, and hardening recommendations.

05

Growth experiment

Hypotheses, landing variants, outreach tests, analytics setup, and success criteria.

06

Internal ops

Checklists, SOPs, integration plans, stakeholder updates, and launch-room coordination.

BEST-FIT PATTERN

Agentic swarming is strongest when one outcome needs several kinds of work and the team benefits from reviewing evidence lane by lane.

How to explain it in conversation

Lead with the outcome, then explain the operating model in plain language.

30-SECOND DESCRIPTION

Agentic Swarming lets a user launch a coordinated set of AI specialists from one command. BuilderStudio plans the lanes, estimates the run, asks for approval, and sends the work into contained Hermes environments. The result is a set of reviewable outputs, not one blended answer.

NON-TECHNICAL

For broad audiences

Think of it as spinning up a temporary project team that can draft, build, test, and audit at the same time, while the user still approves the run and reviews the results.

TECHNICAL

For technical audiences

It is a slash-command orchestration layer that maps a goal to scenario-defined lanes, posts a strict /swarm/run payload to the daemon, and expects isolated Hermes container jobs with per-lane outputs.

FAQ for audiences

Answers to the questions people tend to ask first.

Is this just a group chat of agents?	No. The important part is the operating structure: predefined lanes, approval before execution, cost planning, contained runtime boundaries, and lane-by-lane outputs.
Can users control cost?	Yes. The UI can show the selected model route, lane count, estimated tokens, and estimated cost before the user approves the run.
How is this safer than many agents running freely?	Each lane runs through the Hermes container-only posture. Host shell access, host home and SSH mounts, host installs, and remote shell installers are blocked by design.
What makes the demo compelling?	The launch-stack demo turns one product-launch goal into parallel website, marketing, outreach, launch ops, QA, and security deliverables.

Bottom line: Agentic Swarming helps teams move complex work forward faster while keeping planning, approvals, and review in the open.