

# Optimizing Multi-MCP Workflows



Dan Barr | Sr. Technical Marketing Engineer | Stacklok

# MCP Works

But what comes next?



Anthropic side project → industry standard in < 18 months

Thousands of servers

Hundreds of millions of SDK downloads

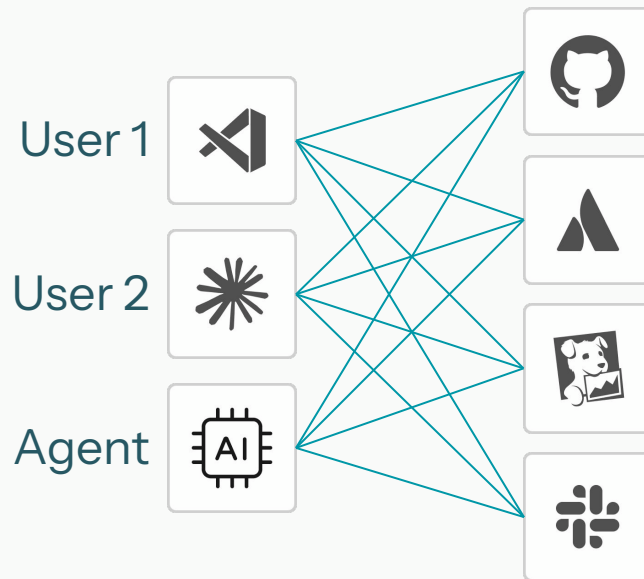
Adopted by OpenAI, Microsoft, Google, and more

Donated to the Agentic AI Foundation

# Multi-MCP challenges

## Configuration sprawl

$N \text{ servers} \times M \text{ developers} =$   
duplicated, drifting configs



# Multi-MCP challenges

## Configuration sprawl

N servers x M developers =  
duplicated, drifting configs



Leads to

## Tool overload

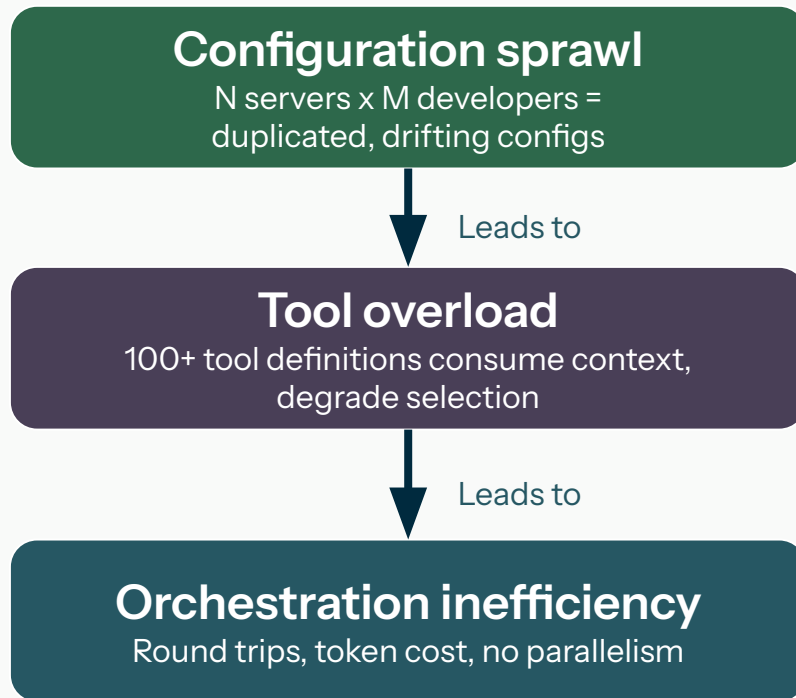
100+ tool definitions consume context,  
degrade selection

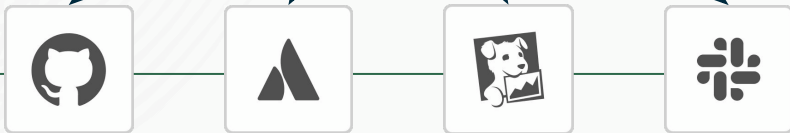
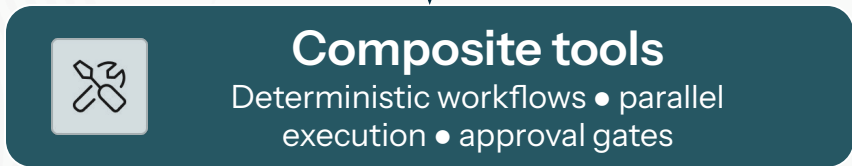
Hundreds of tool definitions

Prompt

Agent context window

# Multi-MCP challenges





Layered problems,  
layered solutions



# ToolHive

by STACKLOK

The open source,  
enterprise-grade  
MCP platform

The screenshot displays the ToolHive MCP Servers interface. The main area shows a list of servers under the 'default' group. The servers listed are:

- canva: Stopped
- github: Running
- playwright-documentation: Running
- toolhive-doc-mcp-remote: (Status not visible)
- context7: Running
- mermaid: Running
- toolhive-do: Stopped
- vercel: (Status not visible)

The 'Manage Clients' panel on the right shows the following clients and their status:

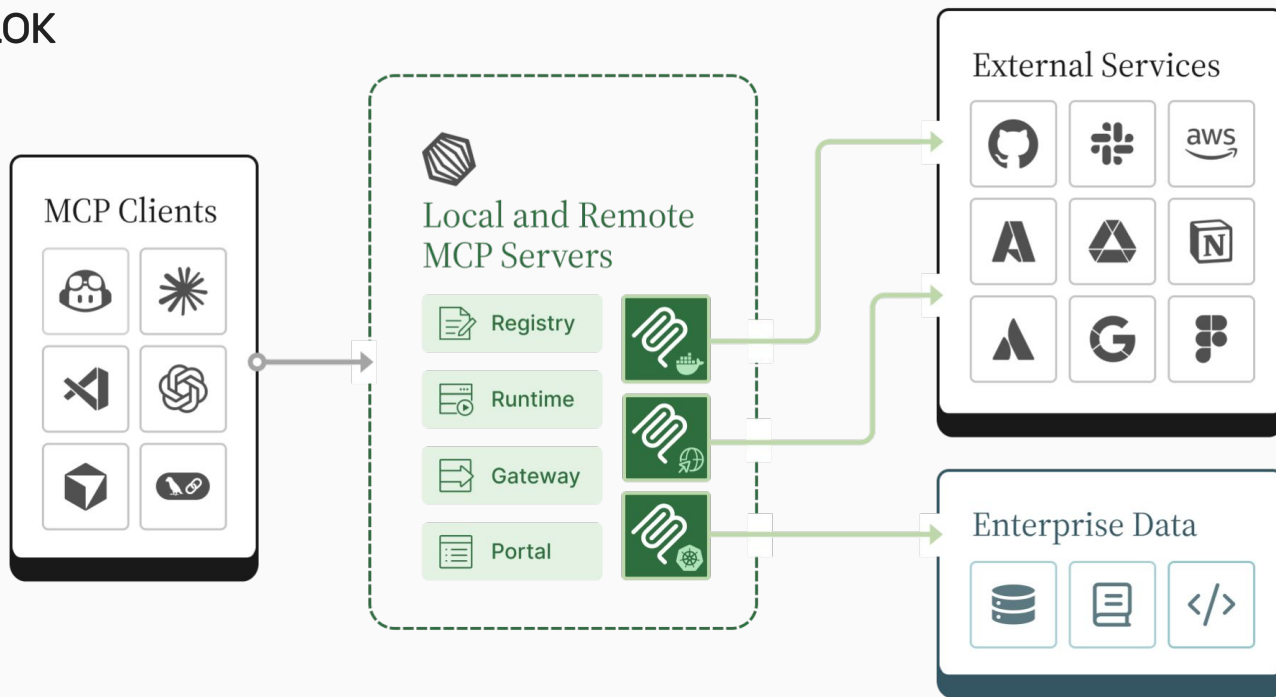
- roo-code: Running
- cline: Stopped
- vscode-insider: Stopped
- vscode: Running
- cursor: Running
- claude-code: Running
- amp-vscode: Stopped



# ToolHive

by STACKLOK

The open source,  
enterprise-grade  
MCP platform





 **vMCP Gateway**  
One endpoint • centralized auth • tool aggregation and filtering

 **MCP Optimizer**  
Just-in-time tool discovery • 60-85% token reduction

 **Composite tools**  
Deterministic workflows • parallel execution • approval gates

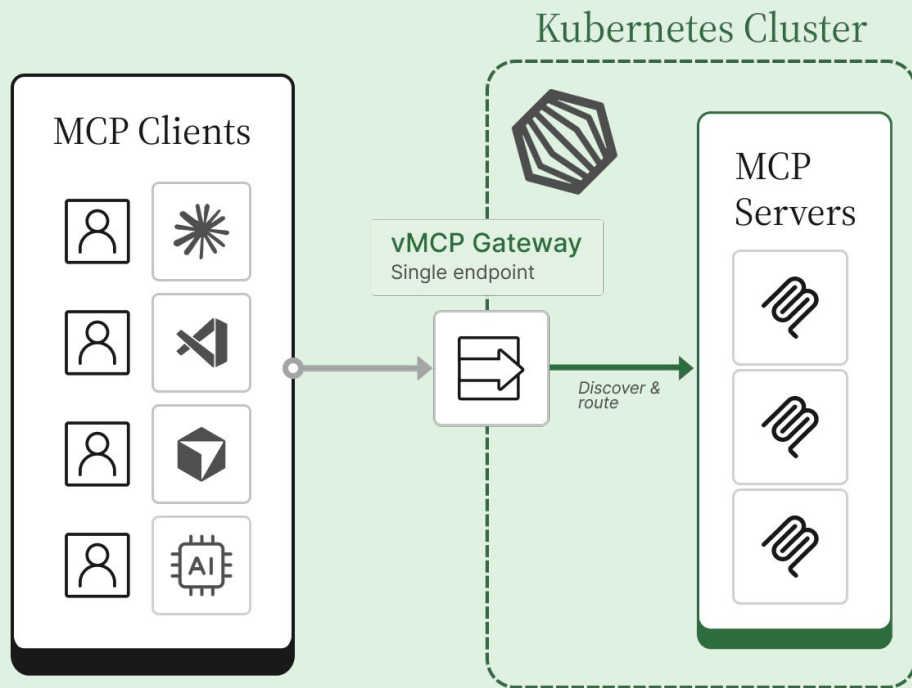


# Layered problems, layered solutions

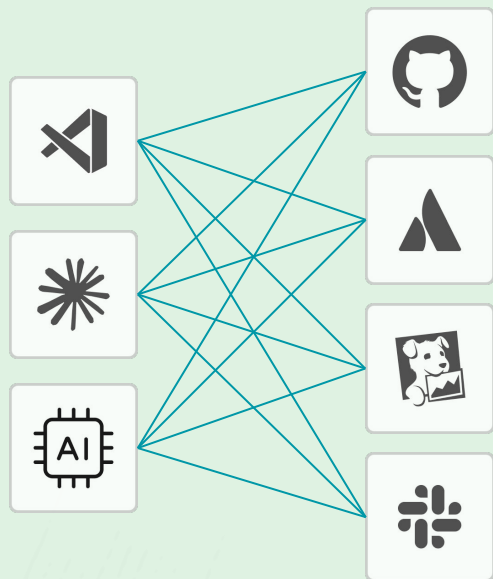
# vMCP: A more intelligent gateway

Centralized access to MCP servers & tools

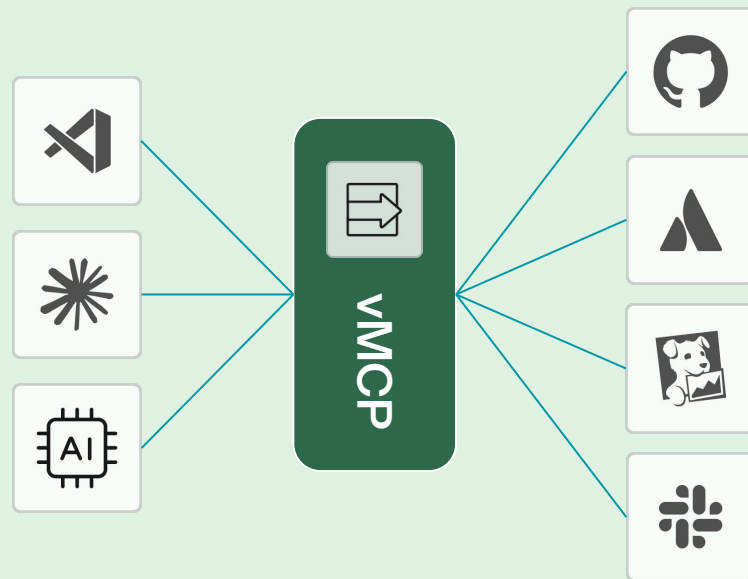
- Single endpoint
- AuthN & AuthZ
- Tool aggregation
- Tool filtering, overrides
- Dynamic discovery
- Kubernetes-native



## Before



## After

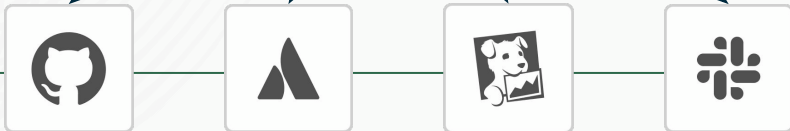
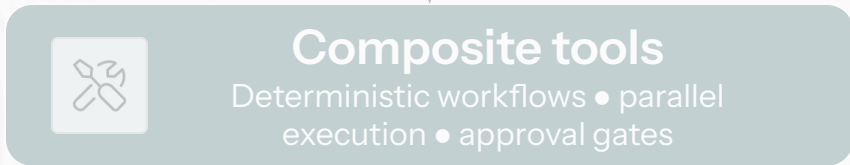


### What changes?

Centralized auth  
Backend token exchange

Conflict resolution  
github\_create\_issue  
jira\_create\_issue

Tool filtering, overrides  
Per-team tool views



Layered problems,  
layered solutions

# Tool optimization

## Without MCP Optimizer

Hundreds of tool definitions

## With MCP Optimizer

find\_tool  
call\_tool

**60-85% token reduction**

Better selection • faster response • lower cost

## Selection accuracy

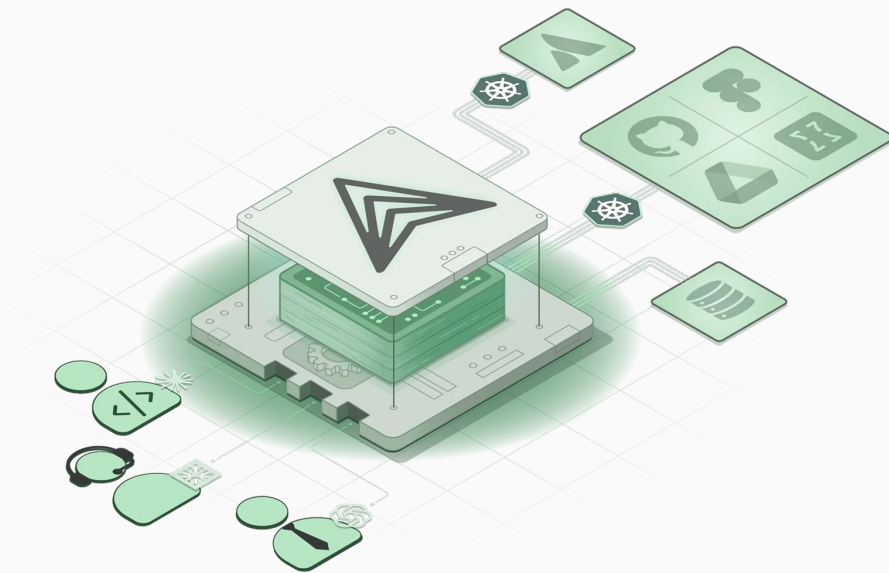
Model	Before	After
Gemini 2.5 Flash	83.2%	92.4%
gpt-oss-20B	38%	69.4%

# Demo

Build a Virtual MCP Server in K8s

Publish to registry

Connect to tools with a few clicks

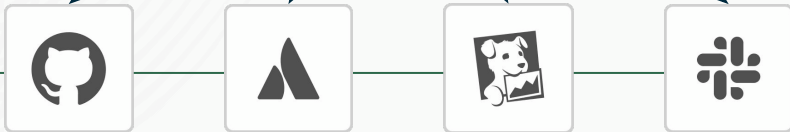




**vMCP Gateway**  
One endpoint • centralized auth • tool aggregation and filtering

**MCP Optimizer**  
Just-in-time tool discovery • 60-85% token reduction

**Composite tools**  
Deterministic workflows • parallel execution • approval gates



# Layered problems, layered solutions

# Declarative workflows

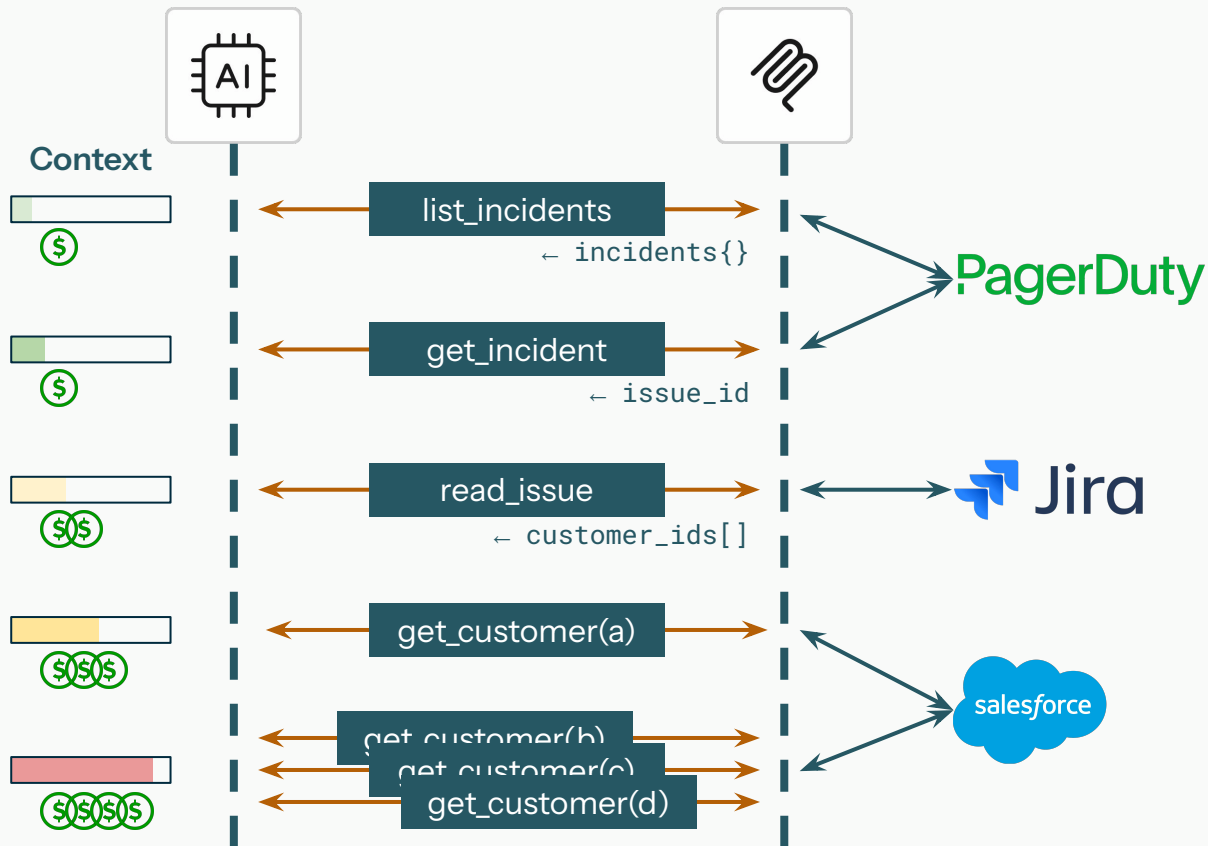
Task-level tools for function-level servers

- Fewer round trips through the LLM – lower cost, lower latency
- Parallel execution where steps are independent
- Structured data passing between steps
- Hide upstream API granularity from the model
- Version-controlled, testable, consistent across model upgrades

# LLM as an orchestrator



“How much revenue is today’s outage impacting?”



# Function-level tools

1:1 API mapping – how servers are built

`get_pull_request`

`get_pull_request_diff`

`get_issue`

`get_issue_comments`

`get_reviews`

`get_check_runs`

`create_review`

# Task-level tool

Composite tool – how assistants think

`prepare_pr_review`

One call, one result

## Composite tool:

pr + diff + issue

parallel

comments + reviews

parallel

checks

parallel

approval gate

human

create review

gated

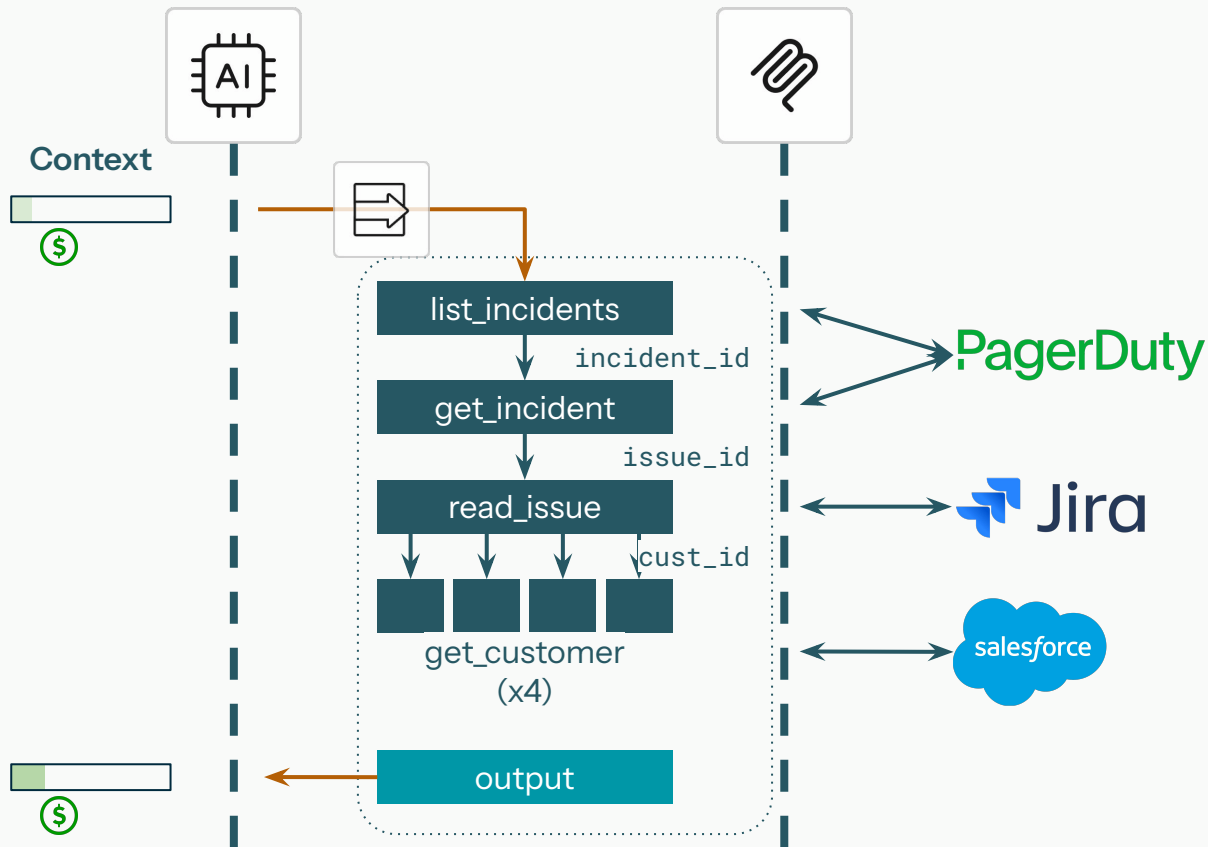
Per-step error handling

Customized output

# vMCP as the orchestrator



“How much revenue is today’s outage impacting?”



# Demo

Defining a composite tool



# What's next for composite tools?

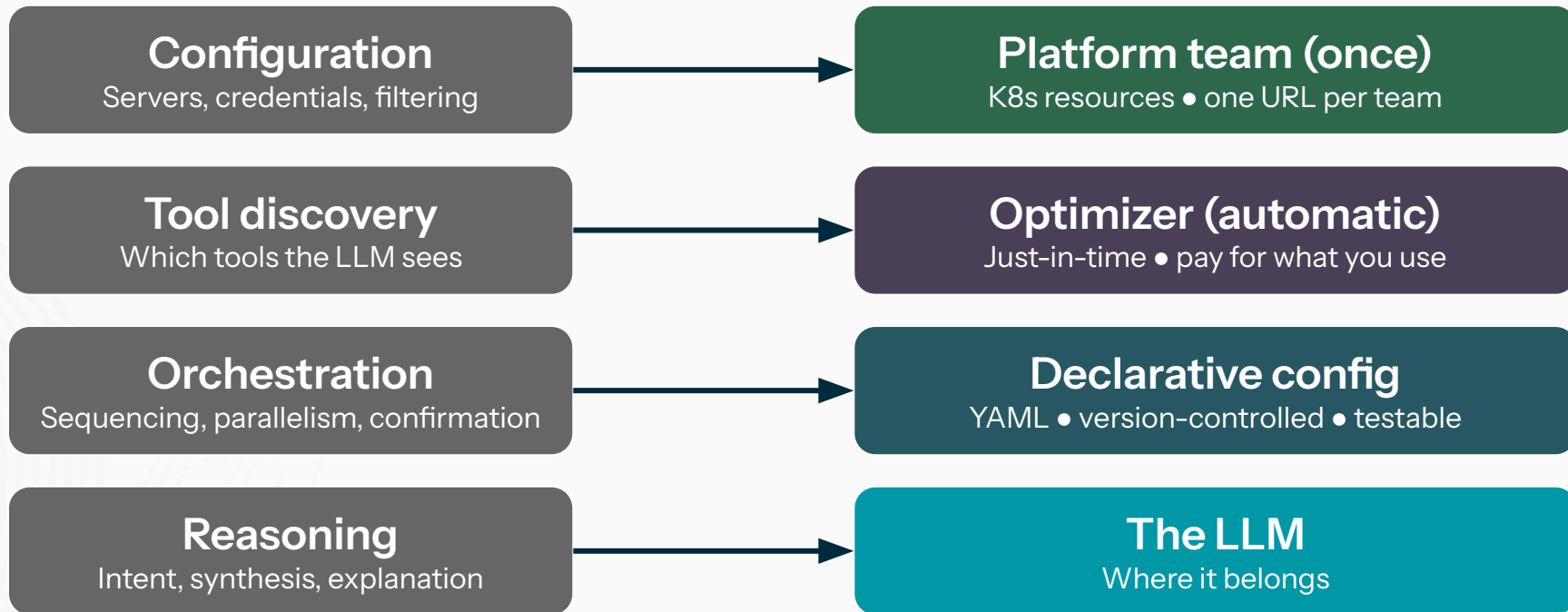
## MCP sampling

Inference at strategic times during the composite workflow

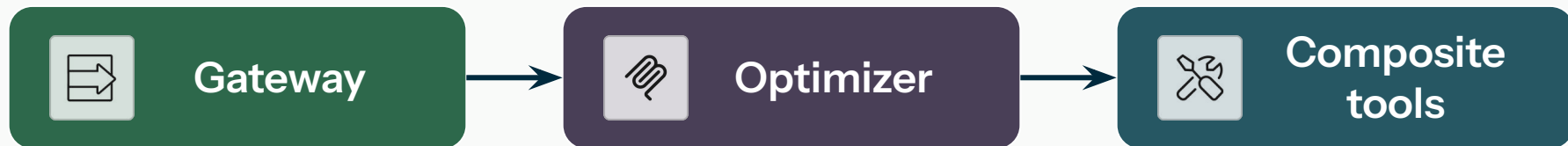
## Workflow engine

Replace DAG+Go templates with a more powerful scripting engine

# What moves where?



# Adoption ramp



Start here to solve...

Configuration complexity  
Credential sprawl  
Tool conflicts

—  
Immediate value, lowest effort

Add if...

Token costs climbing  
LLM picks wrong tools  
5+ servers or 20+ tools

—  
Flip a switch on the gateway

Finally...




Multi-step, cross-system  
Approval gates needed  
Poor MCP tool design

—  
Highest leverage, most design effort

# Get started



**Thank you!** Download ToolHive today, and let's continue the conversation

-  [github.com/stacklok](https://github.com/stacklok)
-  [stacklok.com/download](https://stacklok.com/download)
-  [discord.gg/stacklok](https://discord.gg/stacklok)