MCP + Kubernetes:

Building a Self-Healing AI Platform (Not Just Pipelines)

MCP + Kubernetes:

Building a Self-Healing AI Platform (Not Just Pipelines)

Raghu Reddy

Staff Platform Engineer (Security) - Calix

Cloud Native Hyderabad - Organizer

Calix

CLOUD NATIVE HYDERABAD

MCP Dev Summit Bengaluru

Where the agentic stack is being built.

MCP + Kubernetes:

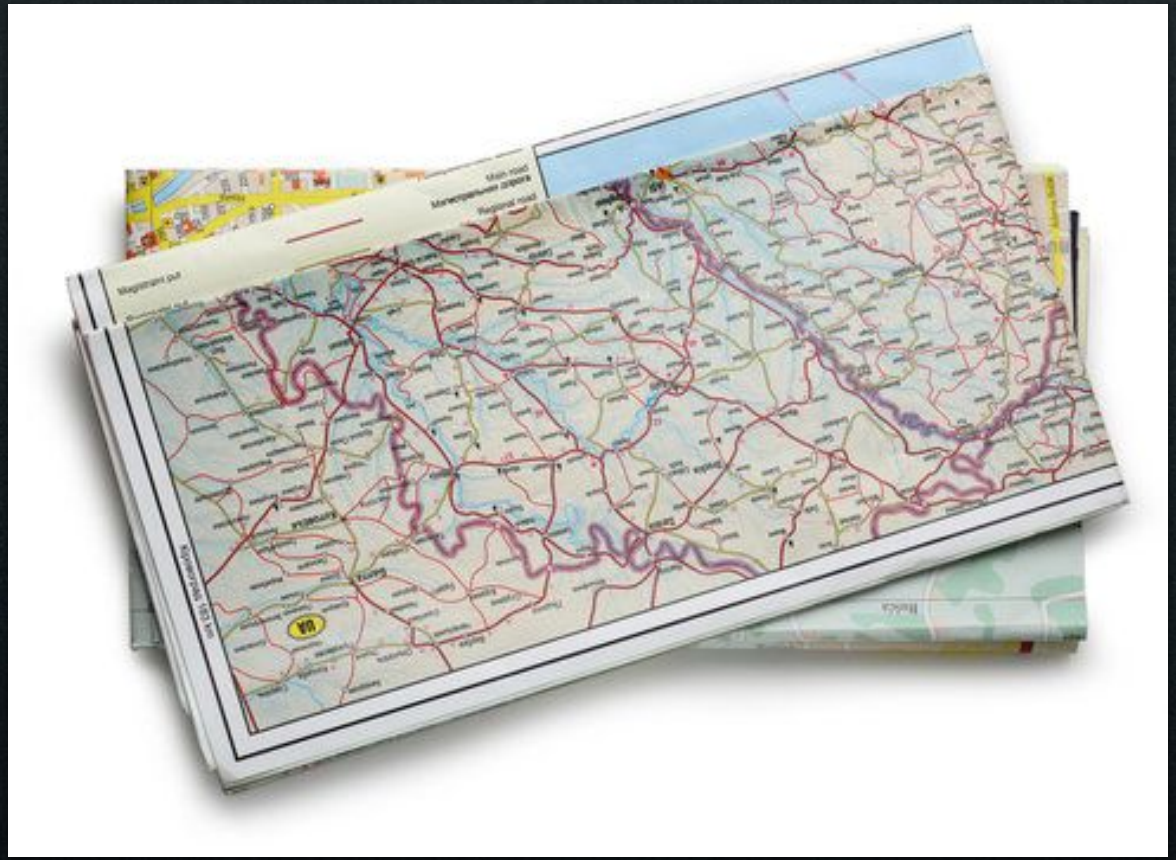
Building a Self-Healing AI Platform (Not Just Pipelines)

Esakki Raj

Senior AIOps Engineer @ Cisco

MCP Dev Summit Bengaluru

Where the agentic stack is being built.

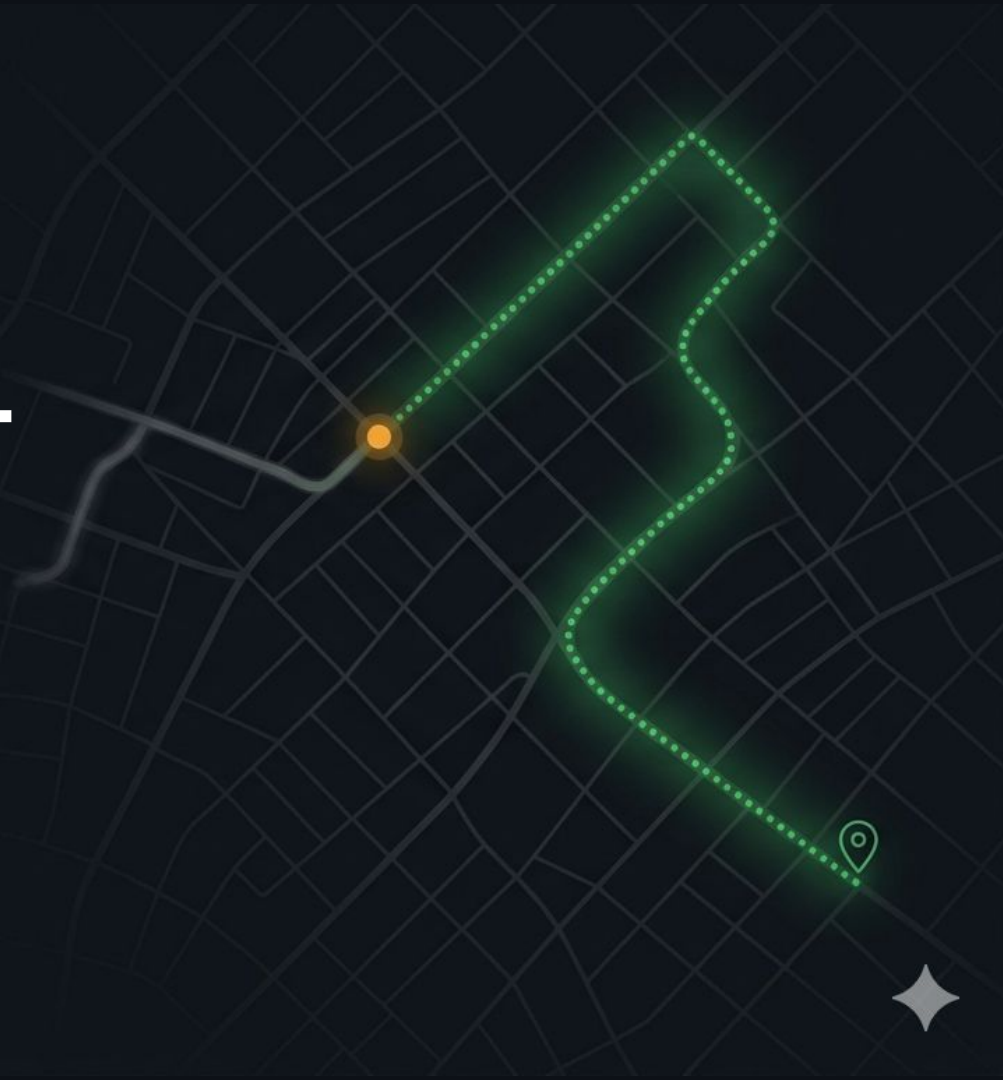



THIS WAY
THAT WAY

Recalculating…

Your map app reroutes itself.
Your AI just... stops

When your AI breaks, the failover plan has a name. It's yours.



Three things you’re getting today

01

The Story

02

The Solution

03

The Reality

Most MCP today is printed directions

🗺️ The Analogy

Paper directions can't recalculate. Miss one turn and you drive into the wrong neighborhood, confidently lost.

⚙️ The Technical Gap

Your pipeline hits a timeout or a garbage tool result, and just continues.

No retry. No fallback. No signal. The user gets the garbage.

A pipeline can't heal itself : by design

❌No Destination

it never knows what "correct" looks like

❌No Recalculation

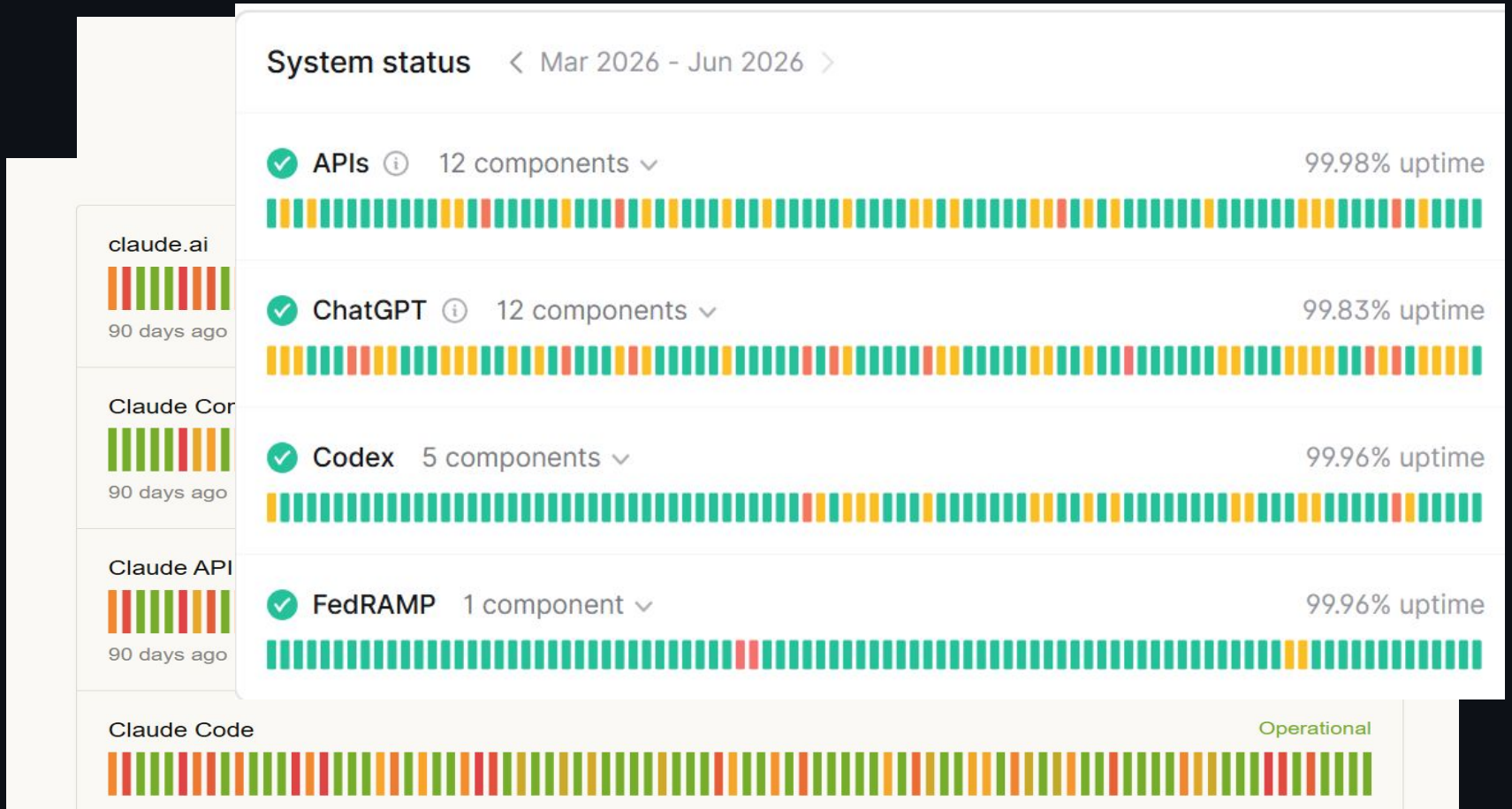
nothing re-checks reality

❌No Recovery

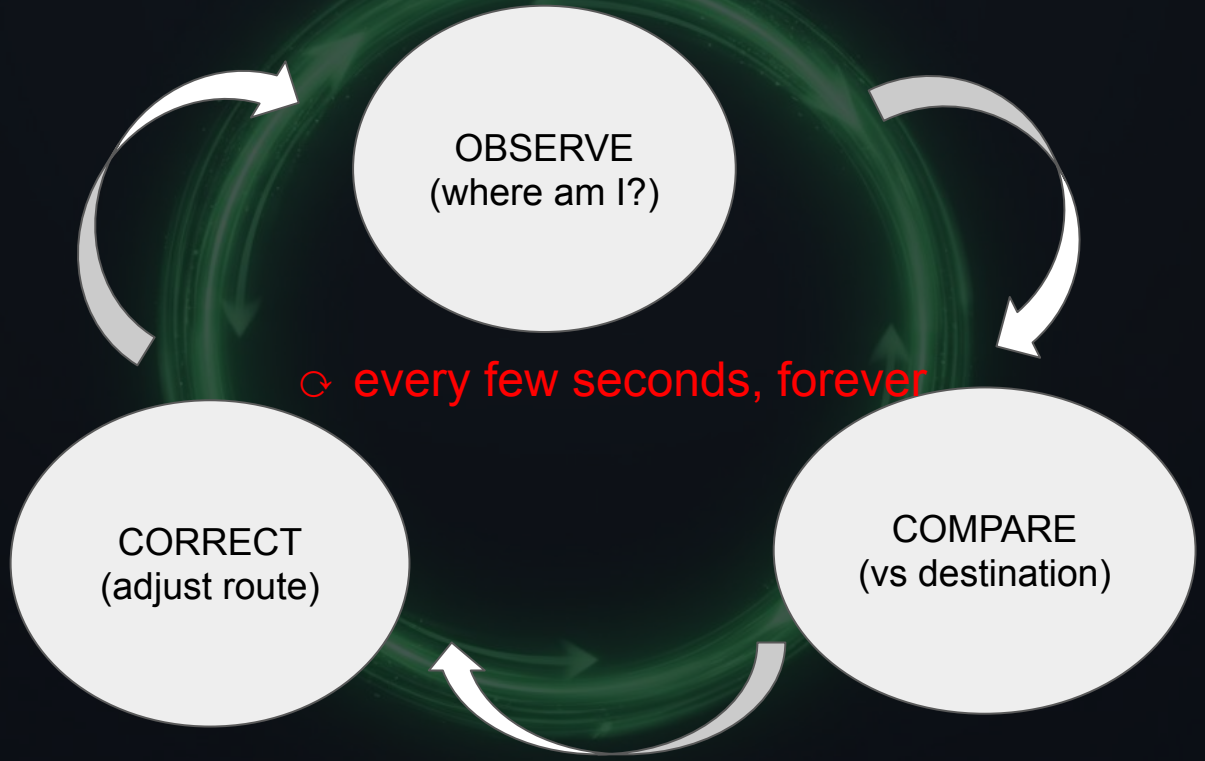
when it breaks, it just… stops

Three things every GPS has. Zero in a pipeline.

Outage

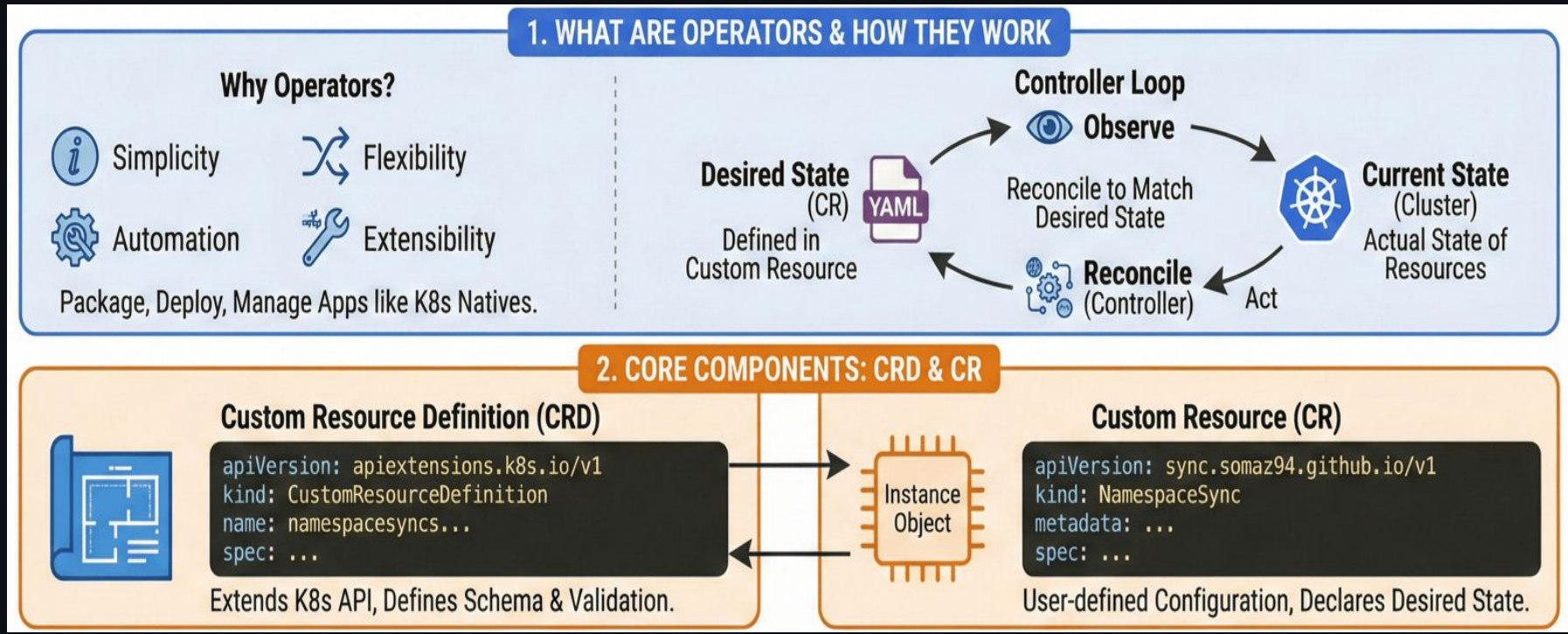

System status
Mar 2026 - Jun 2026
APIs
12 components
99.98% uptime
ChatGPT
12 components
99.83% uptime
Codex
5 components
99.96% uptime
FedRAMP
1 component
99.96% uptime
claude.ai
90 days ago
Claude Cor
90 days ago
Claude API
90 days ago
Claude Code
Operational

This was solved a decade ago : not by AI people



Kubernetes control loop

How Kubernetes keeps a promise: The Operator



How Kubernetes keeps a promise: The Operator

CRD = the kind of place.

Operator = the GPS

CR = your Destination

Loop = recalculating

Treat your AI workflow like a destination

🗺️ The Analogy

You don't give GPS turn-by-turn instructions. You give it a destination. It owns the route.

⚙️ The Technical Gap

You don't script if model_down: fallback(); retry(3).

You declare -
primary: opus, fallbacks: [sonnet].

The operator owns the recovery: forever, not once.

MCP = the car & tools. Kubernetes = the GPS

"You just declare the destination"

MCP Workflow: Custom Resource

mcp_summit > k8s-selfheal-poc > manifests > ! 02-workflow.yaml

```yaml
# The actual workflow instance.
# This is the GitOps-managed, declarative object.
apiVersion: ai.example.com/v1alpha1
kind: MCPWorkflow
metadata:
  name: triage-bot
spec:
  models:
    primary: claude-opus-4-8
    fallbacks:
      - claude-sonnet-4-6
```

An operator re-checks this every 3 seconds, forever

DEMO

GPS gets you to the address — not the right restaurant

🗺️ The Analogy

GPS nails the address. It can't tell you the restaurant is any good.

⚙️ The Technical Gap

A readiness probe says 200 OK. It can't catch hallucinated JSON.
For that, wire an eval / guardrail verdict into the same loop.

The operator doesn't reconcile on pods. It reconciles on a health signal.

MCP Dev Summit Bengaluru

Thank You

MCP + Kubernetes: Building a Self-Healing AI Platform (Not Just Pipelines)