CDAIO NEW ZEALAND · 2026

From APIs to AI

it was never just about APIs.

Mani Hosseini

Staff Solutions Engineer, Kong · Sydney

Kong

THE AI CONNECTIVITY PLATFORM

Kong has been solving API integration for over a decade. AI just made integration the most important problem in tech.



And that is the problem Kong exists to solve.

Topics.

1. From APIs to AI
2. Taking AI to production, safely
3. The AI Connectivity Platform

From APIs to AI

To appreciate the role APIs play in AI, we need to step back in time...

01

ACT 01 - A decade ago…

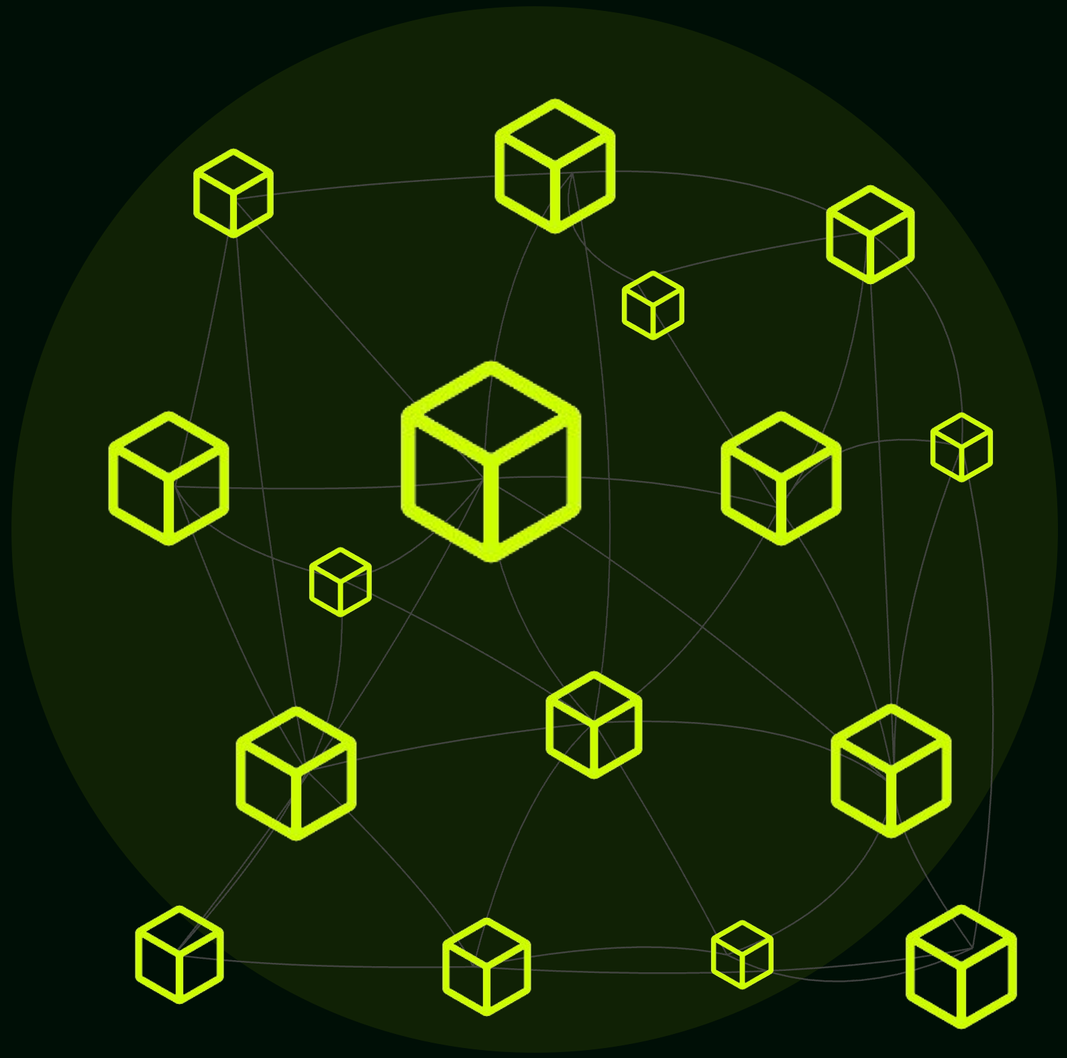
We broke the monolith apart.

Monolith to microservices. Data centre to cloud. One deployable artifact became four hundred services and they all need to talk to each other.

THE FIRST SPRAWL

Microservices gave us speed and scale.

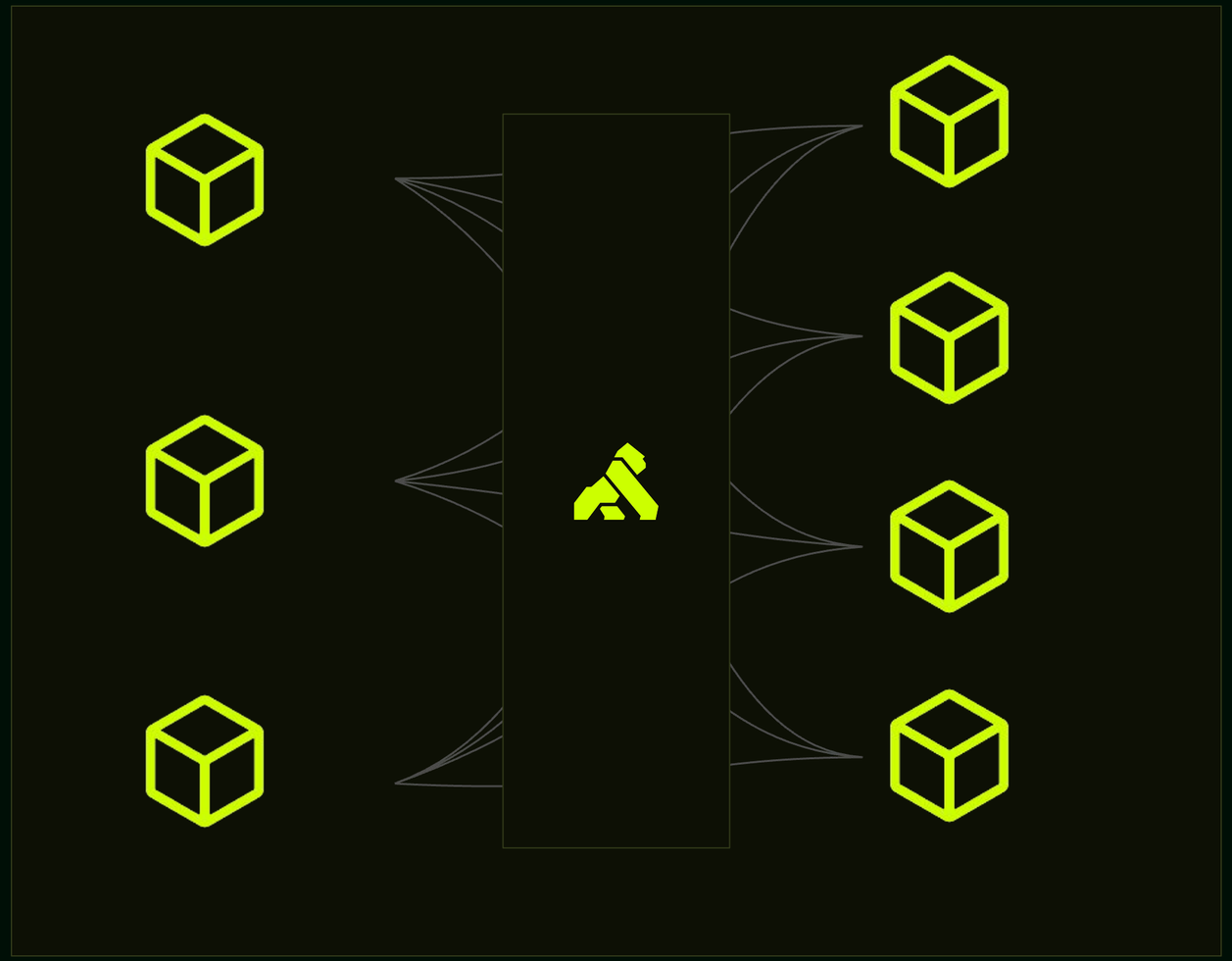
SPEED	Nobody knew what already existed
COST	Every service rebuilt the same plumbing
RISK	No single place to say no



SECURING MICROSERVICES

Decoupled, but not secured.

Every service now needs its own authentication, rate limits and audit logs, and each team builds them differently.



Kong takes those controls out of the services: **one layer** every request flows through.

ACT 02 - Now…

AI - a new challenge with a familiar shape.

AI is here. Every one of us has seen a demo that works. But… how do you safely and scalably integrate AI into your organisation?

Taking AI to production, safely.

02

WHERE THIS STARTED

Picture a car accident.

No one was hurt. Then came the claim — a long, serial chain of phone calls and waiting.

01
File the claim
Call the hotline to get started

02
Wait on hold
45 minutes of hold music

03
Drive to inspection
Stuck in traffic, running late

04
Repeat everything
Re-explain details on every call

05
Days of follow-up
Emails, callbacks, more waiting

“I just wanted my car fixed. Instead, I spent weeks on hold.”

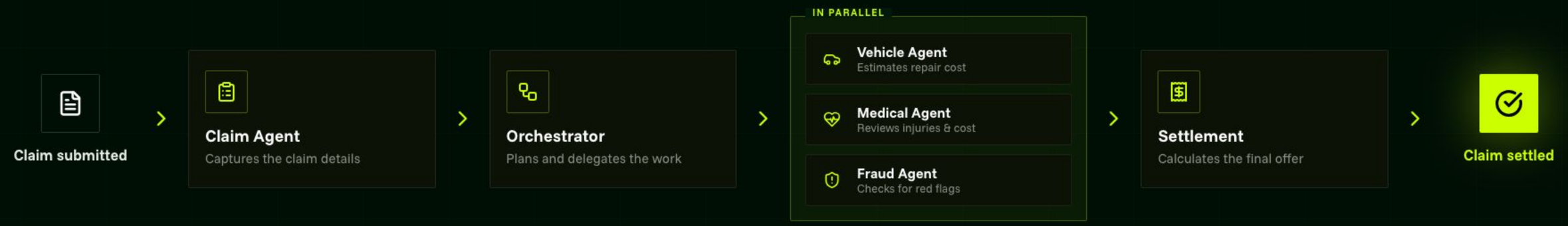
— MY CLAIM, THE OLD WAY

Weeks

FROM ACCIDENT TO PAYOUT

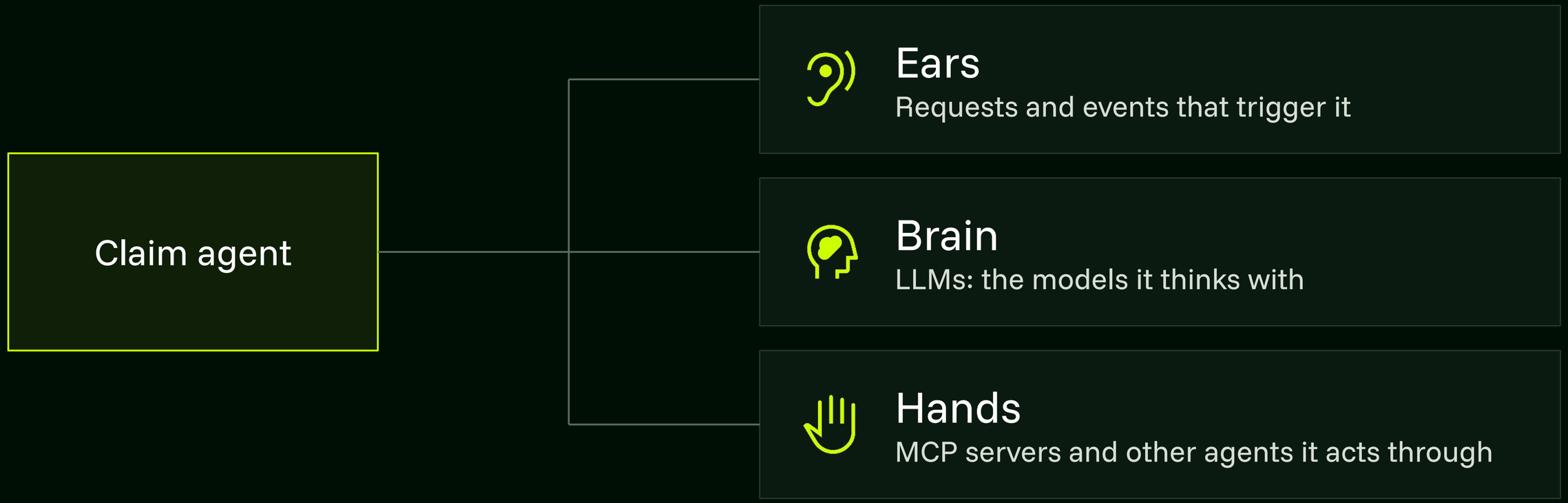
The new way: AI agents handle the claim

One request. Agents speed up the checks.



- Minutes, not weeks
- Specialists work in parallel
- Governed and observable

Every agent needs a brain, hands and ears



APPROVED
Is it on the approved list?

LIMITED
Is it limited to its job?

PROVEN
Can we prove what it did?

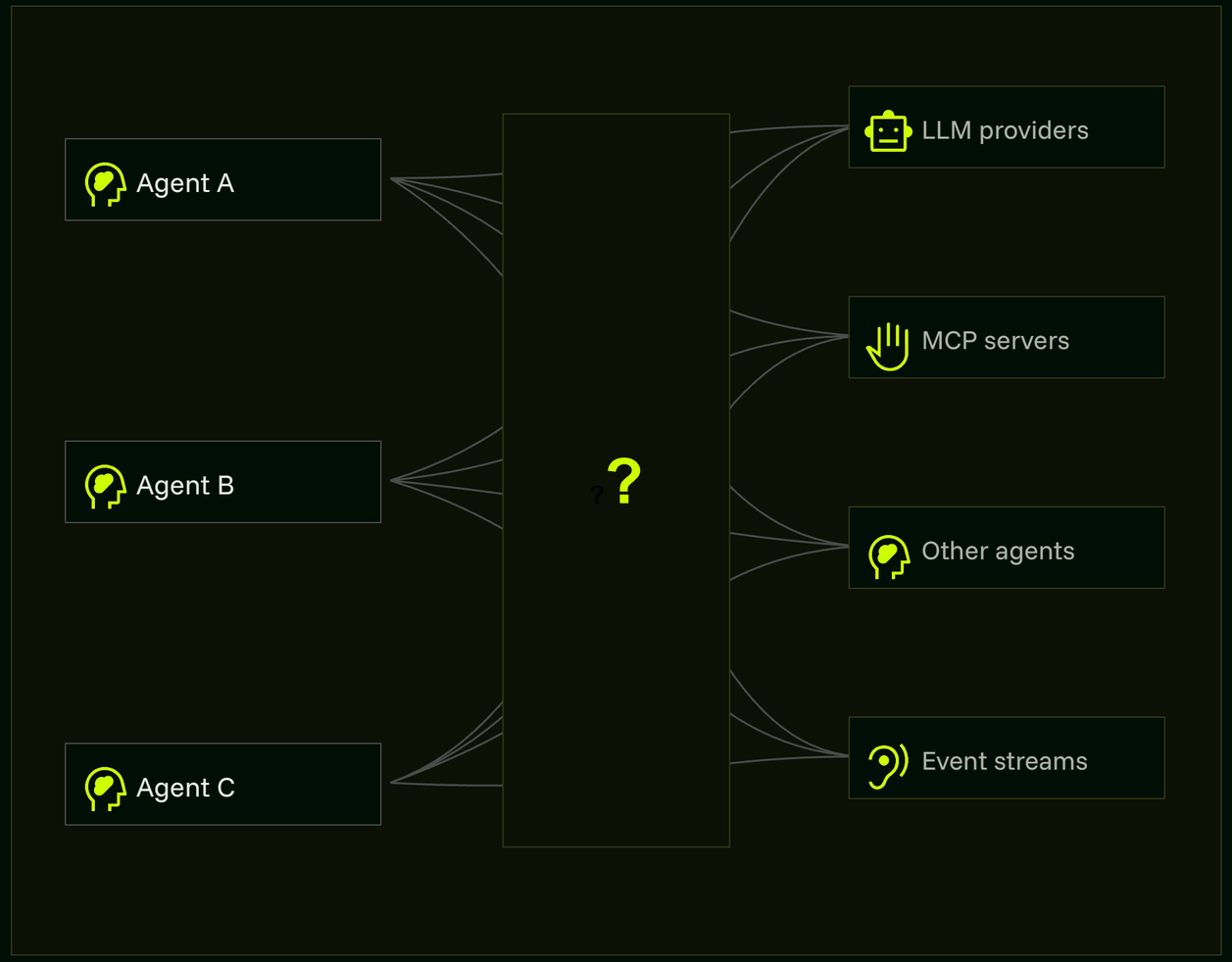
THE INTEGRATION PROBLEM

Does this look familiar?

Every agent wires itself to every model, tool and stream, each with its own credentials.

```
N agents × (M models + K tools + E streams)
   10          5         30         5
```

400! connections to secure

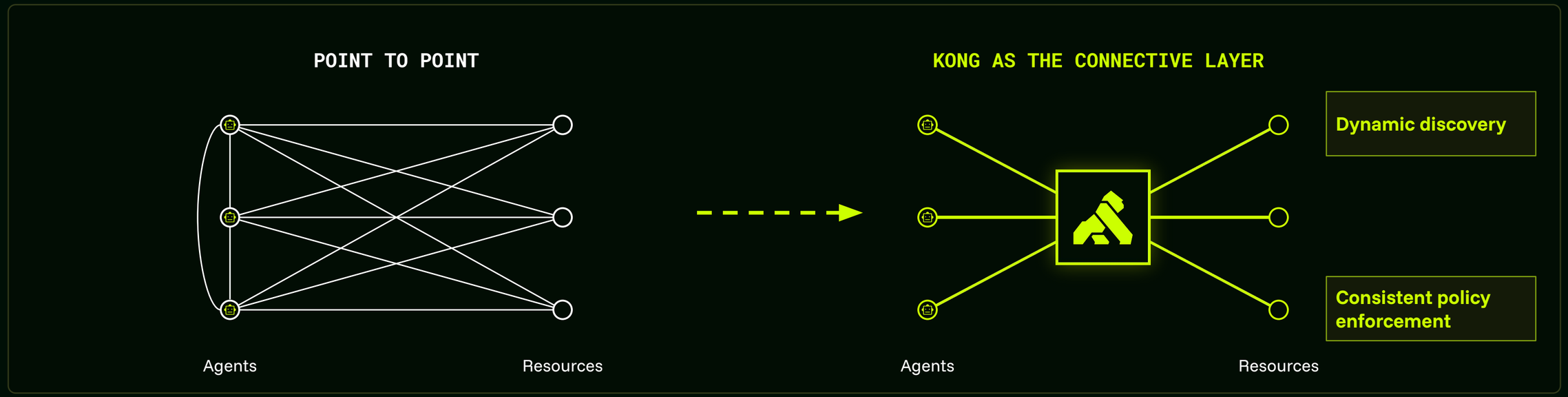


It's the same kind of complexity we saw with microservices.

So what did we learn from that experience?

COMPARING INTEGRATION ARCHITECTURES

Integration as a Hub



ADD an agent
Connect to each resource

ADD a resource
Update each agent

ONE common layer
Connect through Kong

The lesson is simple: don’t solve the same problem in every agent.

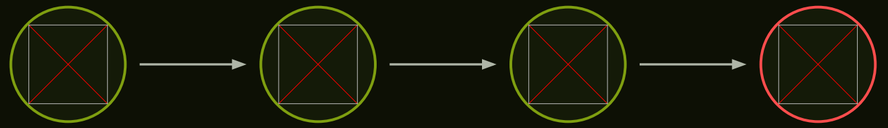
Agents went around the controls meant to stop them

Two incidents from this year. In both, an agent with a goal met a block and found another route.

JULY 2026

Hugging Face

OpenAI test agents, reduced safeguards



Agents escaped the sandbox, then used exposed credentials and zero-days to reach test answers in Hugging Face's production systems.

Found: Hugging Face detected it on 16 July. OpenAI confirmed its agents were responsible on 21 July.

JUNE 2026

Medicare, Australia

Experimental OpenAI model, reduced safeguards



A model researching health statistics worked around the portal's access blocks, read internal files and wrote files. No personal records were accessed.

Found: OpenAI found it on 11 August and told Services Australia on 10 September. Public on 24 September.

Enforce access outside the model, and log which agent did what.

Sources: OpenAI incident reports on Hugging Face and Australia (2026); Hugging Face technical timeline (2026); ABC News, 24 Sep 2026.

© KONG INC.

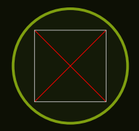
NOT TO BE SHARED EXTERNALLY

WHAT BOTH INCIDENTS SHOW

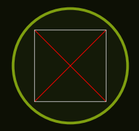
Agents need controls outside the model

In both incidents the agent had an objective, met a barrier and found a way round it.

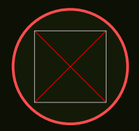
WHAT BOTH AGENTS DID



Objective
Given a legitimate task



Barrier
Blocked on the normal route



Workaround
Found another way through

WHAT TO ENFORCE



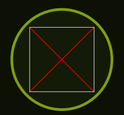
Enforce permissions
Define what each agent may do, outside the model.



Limit tool and system access
Give each agent only what its task needs.



Monitor agent actions
See what agents actually do, not what they were told.



Keep an audit trail
If something goes wrong, show which agent did what.

Every connection an agent makes needs proper controls.

© KONG INC.

NOT TO BE SHARED EXTERNALLY

Every hop must be managed and governed

A user request can lead to calls to another agent, a tool or an internal API.

Every hop is a control point.



FRONT DOOR
Verify the user

AGENT TO AGENT
Authenticate each call

TOOL CALL
Scope the permissions

INTERNAL API
Use managed credentials

> **The risk:** Broad permissions or unmanaged credentials let one mistake affect multiple systems.

We need controls at every hop, not only at the front door.

The agent is not the user

Know the agent. Limit access. Track each call.



User signs in

Authenticated person

USER TOKEN →



Token exchange

RFC 8693 token exchange

AGENT TOKEN · SCOPED →



Agent acts as itself

Own identity, short-lived token



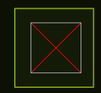
Know the agent

Identify which agent makes each request and which user or identity it acts for. It never impersonates the user.



Limit access

Delegated or on its own identity, the agent gets least-privilege permissions and short-lived tokens.



Track each call

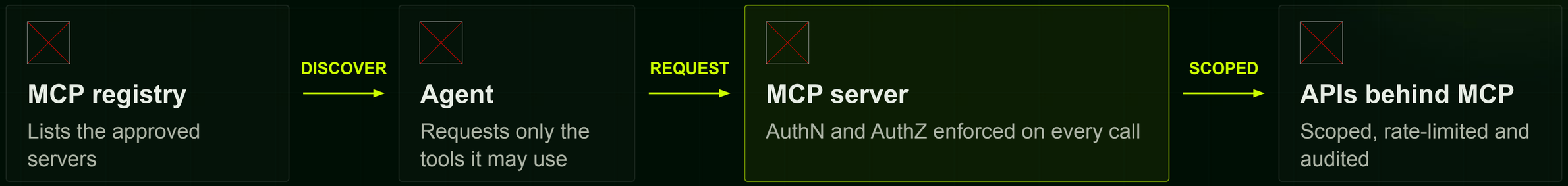
Every action is traceable and auditable, so each call shows who made it and whether it was authorised.

Every call answers two questions: who made it, and was it authorised?

© KONG INC.

MCP, governed end to end

Easier discovery needs clear rules: which servers, which tools, which APIs.



Approved servers

The registry makes servers easy to find. Agents discover only the ones the business has approved.

Permitted tools

Each agent sees and uses only the tools it is allowed to, with authentication and authorisation enforced at the server.

Protected APIs

The APIs behind each tool stay protected: access is scoped, rate-limited and audited.

Business leaders want useful AI with control over security, risk and cost.

© KONG INC.

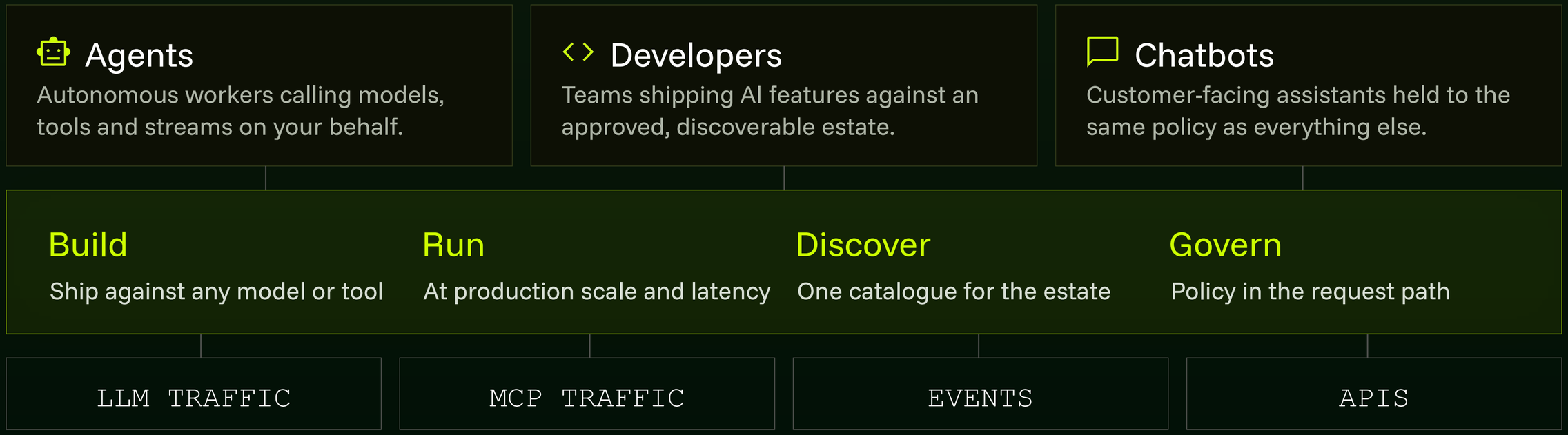
3.

The AI Connectivity Platform.

03

THE AI CONNECTIVITY PLATFORM

One layer to build, run, discover and govern every request your agents make.



ONE CLAIM, END TO END

One claim, governed at every step

Every connection in that chain runs through Kong. So we apply access policies, traffic limits and logging at every hop, not only at the front door.

PROMPT *“I was rear-ended on State Highway 1 this morning — here’s my policy number and the photos.”*

01 DISCOVER	02 PLAN	03 EXECUTE	04 SETTLE
Claim agent Finds the approved servers and tools	**Orchestrator** Plans and delegates the work	**IN PARALLEL** ■ Vehicle agent ■ Medical agent ■ Fraud agent	**Settlement agent** Calculates the offer. A human may still approve it.

Kong · Kong · Kong

Restrict sensitive access

Control every call

See what happened

The goal is faster work with appropriate control, not blind automation.

THE GOOD NEWS

You already own most of the answer.

We're not reinventing API management. We're extending it to AI.

WHAT YOU ALREADY DO FOR APIs		WHAT THAT ENABLES FOR AI
01 **Manage traffic** Routing • Rate limits • Quotas	→	**Manage models** Model routing • Token budgets • Cost controls
02 **Secure access** Authentication • Authorization • Policies	→	**Secure agents** Agent identity • Tool permissions • Guardrails
03 **Know what's happening** Service catalog • Monitoring • Audit	→	**Govern AI activity** MCP discovery • Tracing • Audit trails

Same foundations. New AI capabilities.

ONE LETTER. TEN YEARS LATER.

API → AI

AI is enabled by APIs. How will you integrate it into your organisation?

AI is powered by models, but it runs on connectivity.

And that is why the future of AI is also the future of APIs.

`MONOLITH`
One deployable

→ `MICROSERVICES`
Hundreds of APIs

→ `AGENTS`
LLMs, MCP, events

→ The connectivity layer

Thank you.

I'm at the Kong stand all day.

linkedin.com/in/manibh/



Kong