



SEPT 22 - 24, 2026 | NAIROBI, KENYA

Modernizing Enterprise Architectures to Unlock Agents



Vidura Gamini Abhaya, Ph.D.

VP & Head of Sales Engineering, WSO2



CUSTOMER SERVICE CENTER
OVER 11 MILLION CUSTOMERS STRONG.
40 YEARS OF SERVICE (1984-2024)

OVER 11
MILLION
CUSTOMERS
STRONG



**KULINDA
INSURANCE**





Kulinda Insurance is not a real company.

Kulinda is most of the companies I have walked into in the last twelve months, wearing one name.

If you recognise yourself in the next twenty minutes, that is not a coincidence. It is also not an accusation.



Kulinda Insurance at a glance

Insurance and lending. Not behind on anything.

40

years old

11m

customers

4

core platforms

~2,000

APIs

The digital transformation: *done*

The cloud migration: *done*

An API programme that *won* an award

Their architects are good

Hold on to two thousand. It comes back.



The demo that worked



The demo that worked

Four engineers on the innovation team. Nine days.

- A customer asks, in plain language, where their claim is
- The agent reads the claim and works out what state it is actually in
- It explains the state the way a person would, and offers the next step
- It answered questions the IVR had never handled in fifteen years

I sent the photos twice. Did you get them?

Both sets arrived on the 14th. The second set is what the assessor is working from. Nothing further is needed from you.

Claims status assistant

It was good. That is the part everybody skips.



How hard can the rest be?

The CIO, to the board. Nine days. Four engineers. Two thousand APIs sitting right there.

Budget approved in the room. Six business units told to find their agent use case by quarter end.

1

agent in production, eleven months later

It was the same one. It was the demo.

Wall one: nothing can be discovered

The lending team searches the catalogue for “customer”. It returns three services.

CustomerService v2

Authoritative.
Nobody can prove it.

CustomerService v3

A migration abandoned halfway.
Still responds.

CustomerProfileService

What the mobile app actually
calls, since 2018.

Specifications live in Confluence, last touched in 2019, drifted from the running code.

Six weeks of a twelve-week project went on archaeology, finding people who knew which API was real

Two thousand APIs, and no machine-readable answer to “what can this company actually do?”

Wall two: who is the agent?

Security review asked one question. There were exactly two answers available.

Option one

A service account holding the union of every permission it might conceivably need. More than any human employee is trusted with, on a credential that never expires.

Option two

Give it nothing.

What they actually wanted

This agent, acting on behalf of this customer, for this purpose, for the next ten minutes.

Security did the responsible thing and gave it read-only access

Eleven months in, every agent could answer a question and none could do anything.

(There's a word for it, especially from the CFO PoV)

Wall three: six percent

The claims adjudication assistant evaluated at ninety-four percent accuracy.

94%

accuracy in evaluation

6%

of adjudications wrong

~40,000

customers a year, at Kulinda's volume

But the accuracy is not what killed it.

- No downstream system was designed to be called by 'something that guesses'
- No idempotency keys — for thirty years the caller was a form, and a form does not hallucinate
- No compensating path to unwind a wrong adjustment
- No audit record that could reconstruct *why* it decided what it decided

The risk committee did not say no. They said: show me the trace.

Wall four: six teams, six of everything

Every business unit building at once, with nothing shared underneath.

6

parallel stacks

6

ways of doing memory

3

model contracts, two
unapproved

9

weeks to answer the
regulator

The regulator's question was one any regulator is entitled to ask.

What customer data has left your estate, and who received it?

The answer took nine weeks, and it began with the words “we believe...”

Four walls

Eleven months, one agent, four reasons.

Discovery

Two thousand APIs and no way to learn which one is true

Identity

All the permission, or none. Nothing in between

Blast radius

Six percent, and no way to unwind or explain it

Sprawl

Six of everything, and no way to govern things

Not one of them is a problem with the model

Not one of them gets better if you swap in a smarter model next quarter

Each one is a problem with the enterprise the model was pointed at

Kulinda's architecture was not broken
It was designed — for humans

Where the human was standing

Every gap was real. Every one was filled by a person, invisibly, for free, for decades.

Three services called “customer”

A developer asked a colleague. Tribal knowledge did the disambiguation. It worked well, nobody even had to write it down

A service account with everything

A person with judgment decided which permissions to actually use today.

No idempotency on payments

People do not click submit twice. When they do, operations notices.

No record of why

You could walk across the floor and ask. That was the audit trail.

Agents are the first thing we have connected to these systems that does not fill those gaps

A genuinely new class of consumer

Three properties your architecture has never had to serve.

Non-deterministic

Same input, different path.
Every integration contract any of us has written assumed the opposite.

Autonomous

It acts in between your requests.
Your security model assumes a session, with somebody at the other end of it.

Scales like software, decides like a person

We know how to govern things that scale. We know how to govern people who decide. Not both, ten thousand times a minute.

It scales like software and decides like a person. That combination is the new thing.

**Your enterprise should be discoverable,
governable and safe to act on**

**Your Enterprise Architecture should
facilitate it**

This is not new work. It is the work most of you have been trying to get funded for a decade.

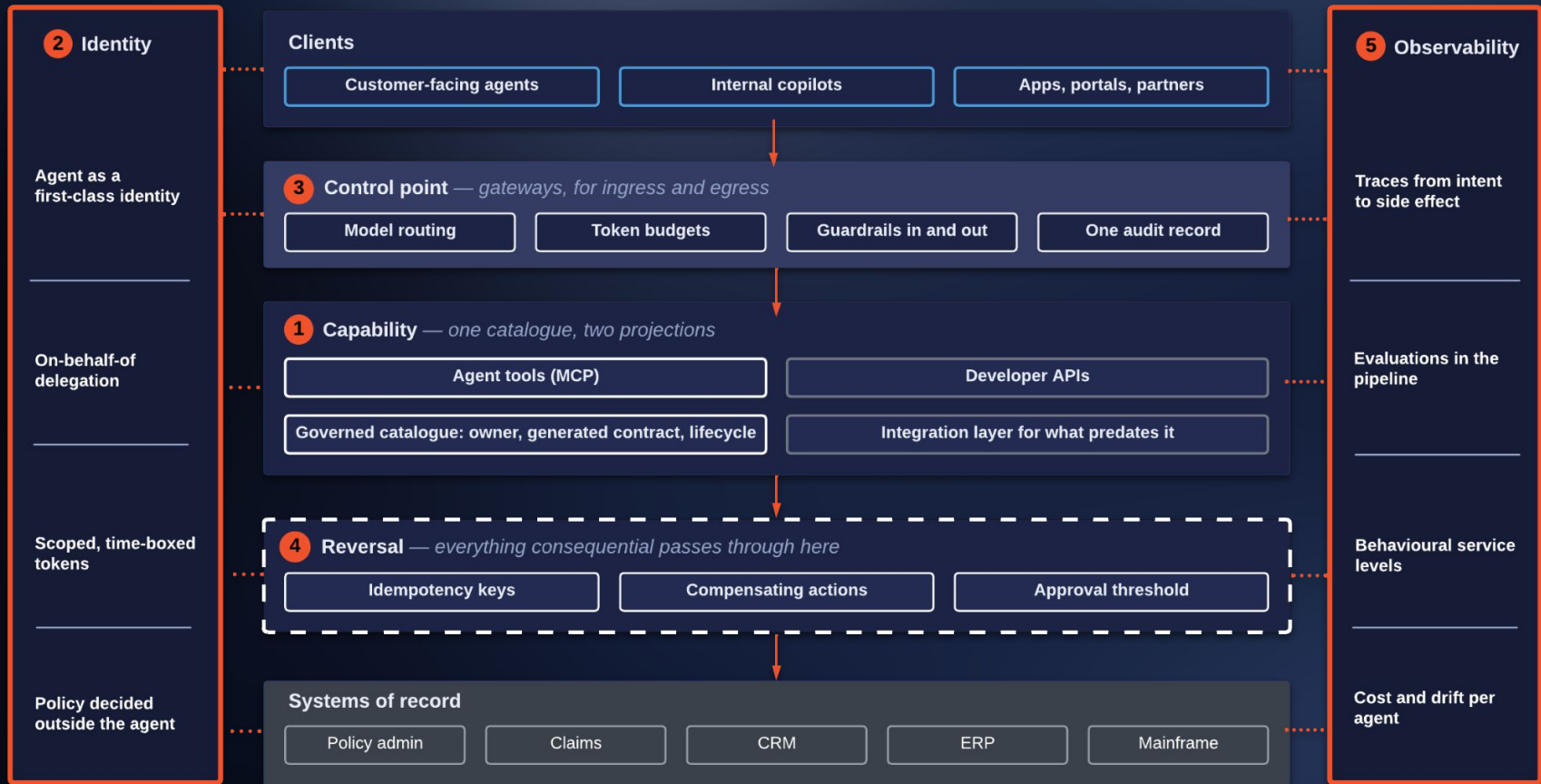
Agents are simply the first consumer that refuses to compensate for it not being done.

The rebuild



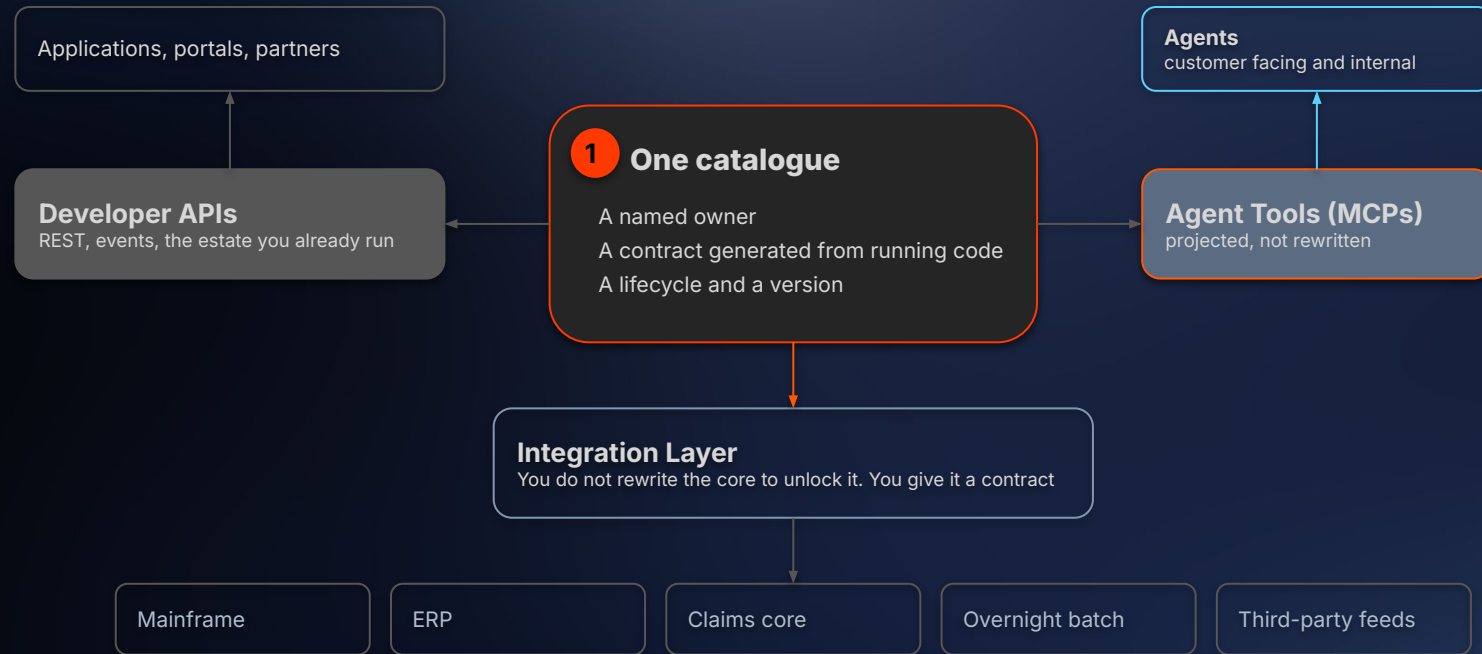
Five Layers

What Kulinda rebuilt. Each layer answers a question that the demo never had to



Capability that can be discovered

Answering the question that cost the lending team six weeks: What can this company do?



Where this sits

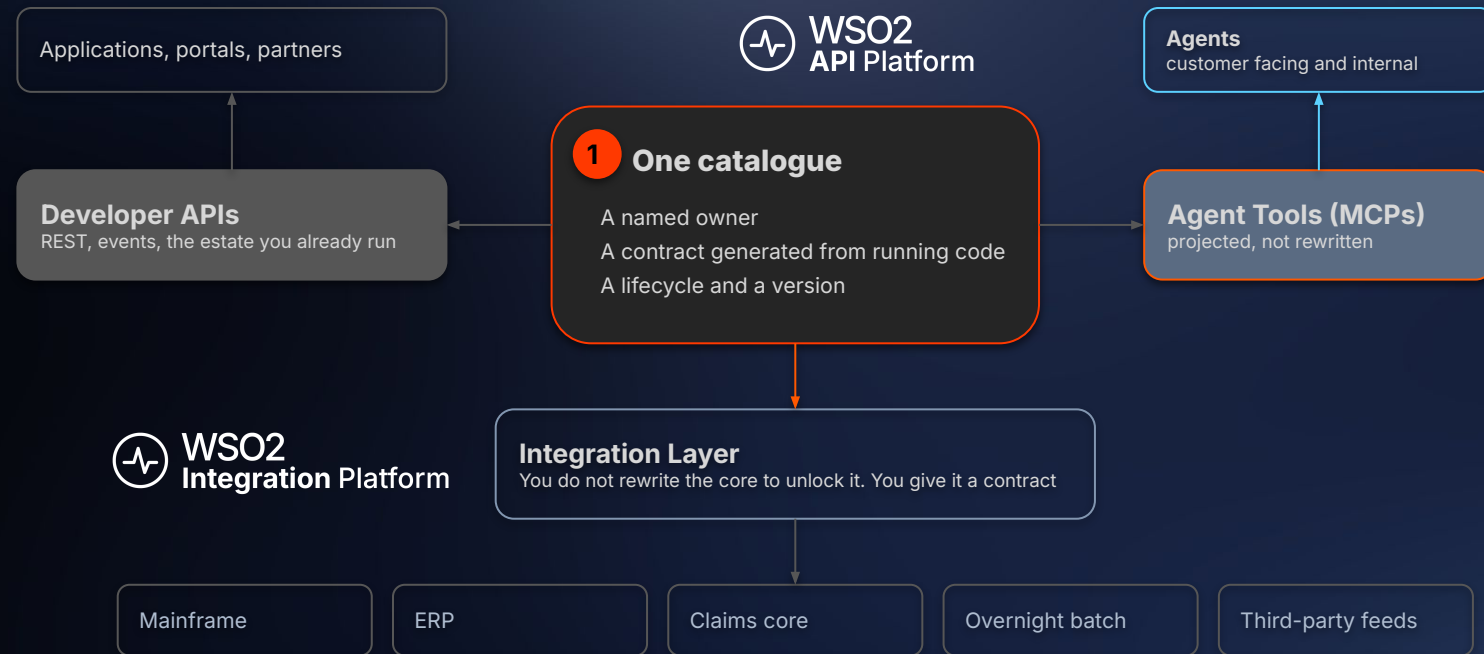
- 1 Capability
- 2 Identity
- 3 Control Point
- 4 Reversal
- 5 Observability

Same contract, same policy, same owner. Build separate agent APIs and within two quarters the copies have drifted



Capability that can be discovered

Answering the question that cost the lending team six weeks: What can this company do?



Where this sits

- 1 Capability
- 2 Identity
- 3 Control Point
- 4 Reversal
- 5 Observability

Same contract, same policy, same owner. Build separate agent APIs and within two quarters the copies have drifted

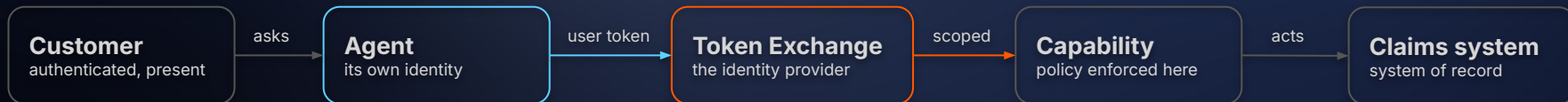


Identity for things that are not people

The delegation that moved Kulinda's agents from read-only to acting

Where this sits

- 1 Capability
- 2 Identity**
- 3 Control Point
- 4 Reversal
- 5 Observability



The decision lives outside the agent

Policy is evaluated at the capability, not in the agent

A prompt is not a policy enforcement point

Revoke or narrow without redeploying anything

2 What the scoped token actually says

Subject	agent / claims assistant
Acting for	customer 8842
Scope	Claim.read, claim.status.write
Expires	In ten minutes

The agent can never do more than the person who asked it. Usually less. Not for long

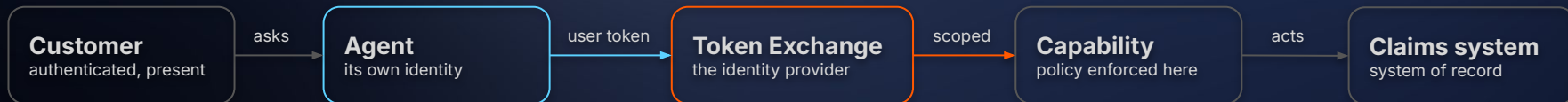


Identity for things that are not people

The delegation that moved Kulinda's agents from read-only to acting

Where this sits

- 1 Capability
- 2 Identity**
- 3 Control Point
- 4 Reversal
- 5 Observability



The decision lives outside the agent

Policy is evaluated at the capability, not in the agent

A prompt is not a policy enforcement point

Revoke or narrow without redeploying anything

2 What the scoped token actually says

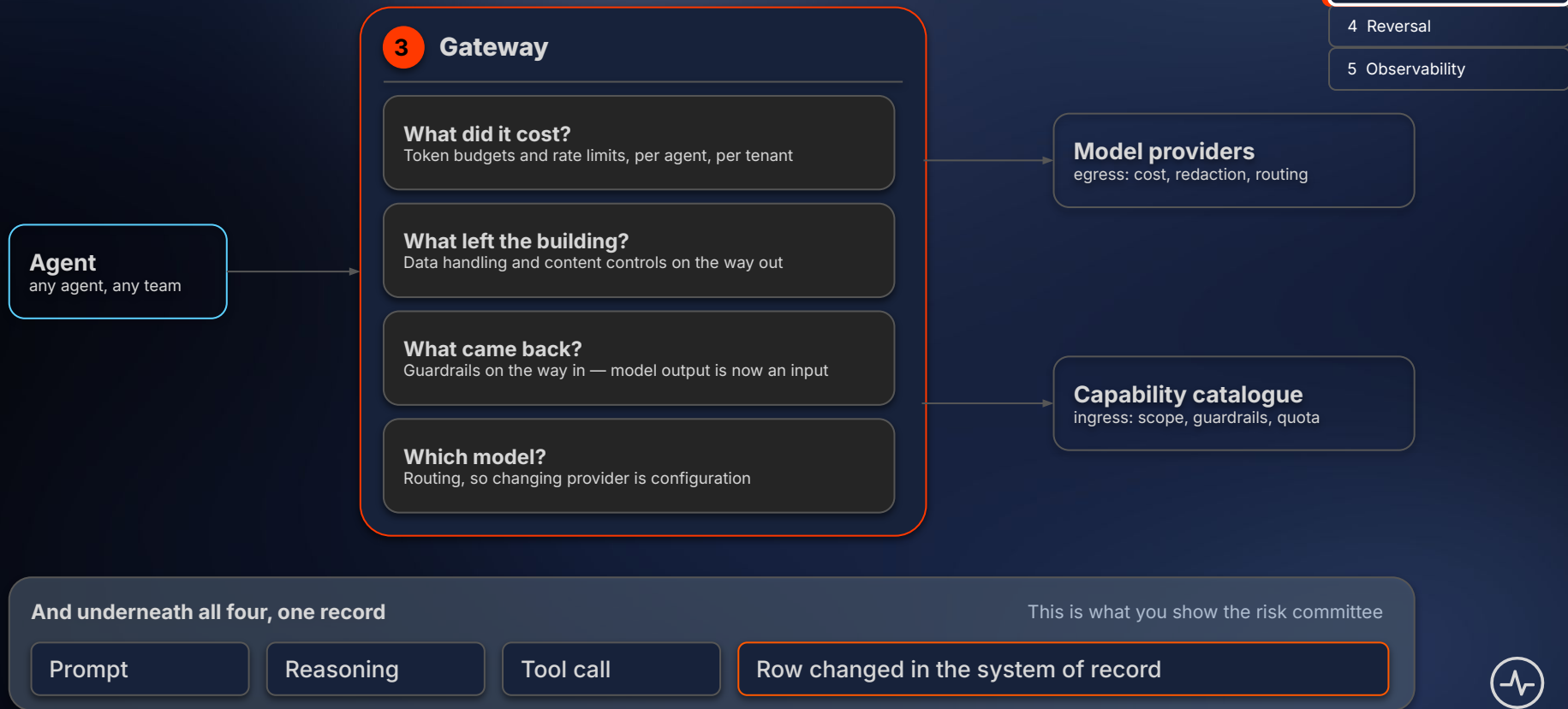
Subject	agent / claims assistant
Acting for	customer 8842
Scope	Claim.read, claim.status.write
Expires	In ten minutes

The agent can never do more than the person who asked it. Usually less. Not for long



A control point in the path

Four questions you will be asked. All four are answered in the same place



A control point in the path

Four questions you will be asked. All four are answered in the same place



Agent
any agent, any team

3 Gateway

What did it cost?

Token budgets and rate limits, per agent, per tenant

What left the building?

Data handling and content controls on the way out

What came back?

Guardrails on the way in — model output is now an input

Which model?

Routing, so changing provider is configuration

Model providers

egress: cost, redaction, routing

Capability catalogue

ingress: scope, guardrails, quota

Where this sits

1 Capability

2 Identity

3 Control Point

4 Reversal

5 Observability

And underneath all four, one record

This is what you show the risk committee

Prompt

Reasoning

Tool call

Row changed in the system of record

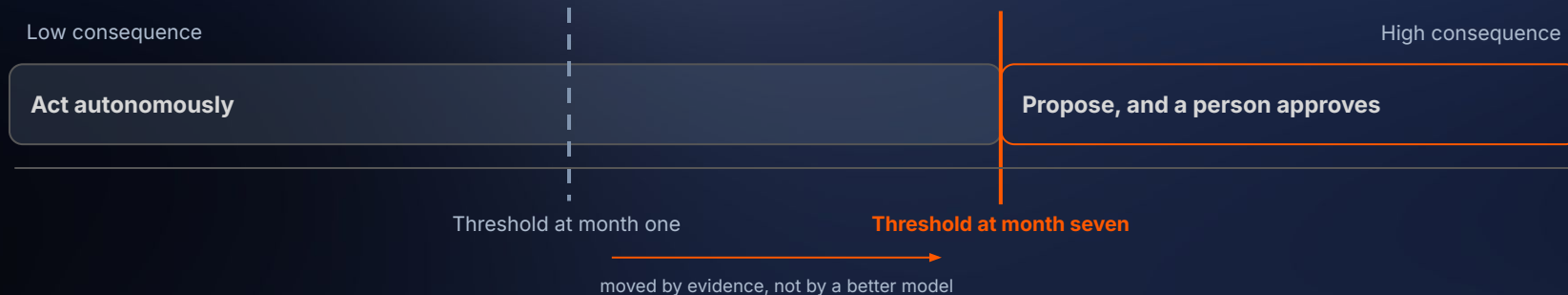


Design for being wrong

The approval step is not friction to remove later. It is what raises the ceiling

Where this sits

- 1 Capability
- 2 Identity
- 3 Control Point
- 4 Reversal**
- 5 Observability



Idempotency keys

Your agent will retry. Thirty years of callers were forms, and a form does not retry with a slightly different plan

Compensating actions

Every consequential action has a defined way back. If it cannot be unwound, it does not get done autonomously

Approval as evidence

Each human decision above the threshold is a labelled example. That is what lets you move the line next quarter

Reversibility is what buys permission to go faster



Observe behaviour, not infrastructure

An agent can return a 200 on every call and be doing precisely the wrong thing

Where this sits

1 Capability

2 Identity

3 Control Point

4 Reversal

5 Observability

One trace, from intent to side effect

A customer asks where their claim is

Agent reasoning...

verify_customer

get_claim

write claim.status_note

Row changed in the claims core

Latency and error rate are still true. They are now insufficient

Evaluations in the pipeline

A prompt change is a deployment. It can fail a test like any other deployment, before it reaches a customer

Behavioural service levels

Not whether it responded. Whether it did the right thing, measured on the cases you actually care about

Latency tells you the system is healthy. Only behaviour tells you it is right



Observe behaviour, not infrastructure

An agent can return a 200 on every call and be doing precisely the wrong thing



WSO2
Agent Platform

Where this sits

1 Capability

2 Identity

3 Control Point

4 Reversal

5 Observability

One trace, from intent to side effect

A customer asks where their claim is

Agent reasoning...

verify_customer

get_claim

write claim.status_note

Row changed in the claims core

Latency and error rate are still true. They are now insufficient

Evaluations in the pipeline

A prompt change is a deployment. It can fail a test like any other deployment, before it reaches a customer

Behavioural service levels

Not whether it responded. Whether it did the right thing, measured on the cases you actually care about

Latency tells you the system is healthy. Only behaviour tells you it is right



The paved road

The last piece, which is not technology at all.

One platform carrying all five layers

Capability · Identity · Control point · Reversal · Observability

Lending

Claims

Servicing

Fraud

Underwriting

Partner

The alternative is six teams each building all five, badly, six times over.

How long does a team that has never built an agent take to get one into production, safely, without talking to you?

At Kulinda that went from eleven months to about nine days. Which is where we came in.

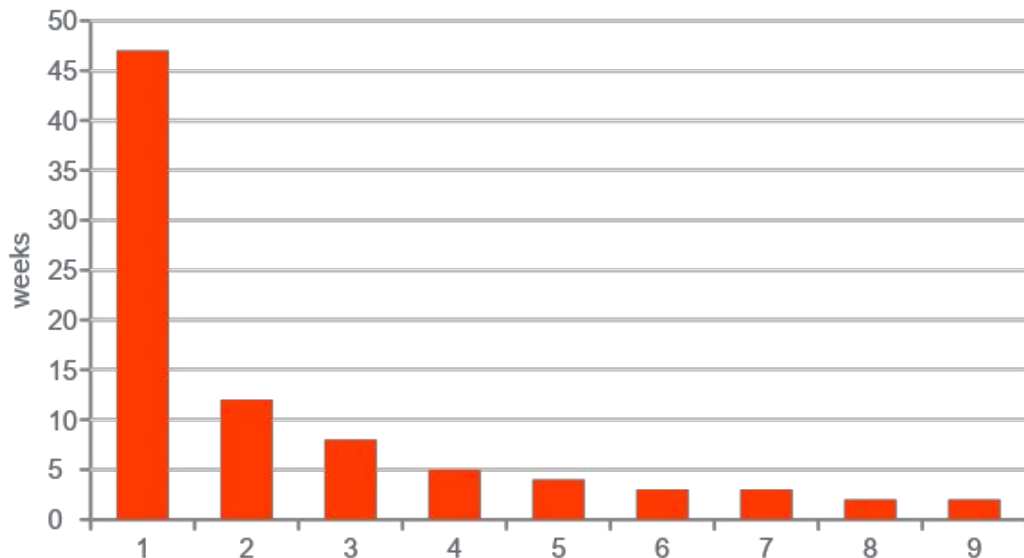
9

agents shipped last quarter

Not because the ninth was easy to build. Because by the ninth there was almost nothing left to build.

The return is in the second agent

What the first agent actually buys is the cost of every agent after it.



The first agent cost eleven months.

Not because it was hard. The demo took nine days. It cost eleven months because the first agent pays for the architecture.

The second cost weeks. The ninth cost days.

You are funding the cost of every agent after this one.

Ask for the business case on the first agent and you will get a correct answer to the wrong question.

Monday morning

One ask for each group in this room.

If you are an architect

Pick your single highest-value capability. One, not forty. Take it through all five layers: catalogued, contracted, projected as a tool, behind the gateway, delegated identity, traced end to end.

A quarter of work, not a year.

If you run a platform

Your deliverable is not an agent. It is the paved road. Measure yourself on time to first safe agent, for a team you have never met.

Publish that number.

If you are an executive

Stop asking which use case. Start asking which capability. Use cases are infinite and they change every quarter.

Capabilities compound.

Every one of those four walls was there long before anybody said the word “agent”.



SEPT 22 - 24, 2026 | NAIROBI, KENYA

You are *not* late to agents.
You are early to the architecture that makes
agents worth having



Vidura Gamini Abhaya, Ph.D.

VP & Head of Sales Engineering, WSO2