WSO2CON AFRICA

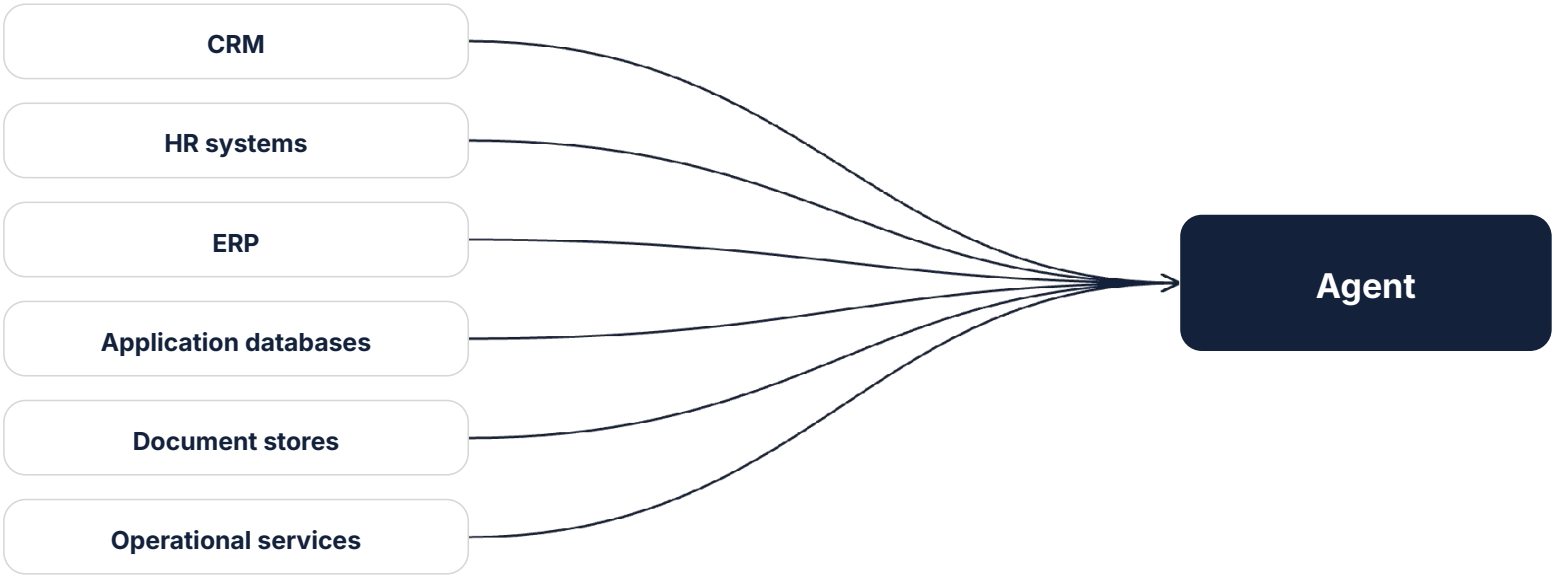
SEPT 22 – 24, 2026 | NAIROBI, KENYA

The Data Layer for Enterprise Agents

Maryam Ziyad

Senior Technical Lead

Your agents need access to your enterprise data



Connecting them is essential work – and if you're building enterprise agents, you're probably already doing it.

RAG: answering from your own content

AHEAD OF TIME

Documents → chunk → Chunks → embed → **Vector index**

AT QUERY TIME

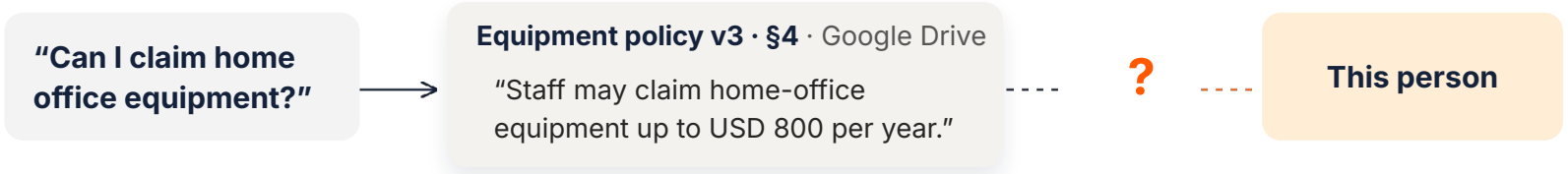
Question → embed, search the index → Relevant chunks → Question + chunks → Model → Answer

Question	Retrieved · Equipment policy v3 · §2	Answer
"What is the laptop allowance for a new joinee?"	"New employees may expense one laptop up to USD 1,500."	**"Up to USD 1,500 for one laptop."**

For a bounded question whose answer sits inside the retrieved text, this is enough.

What a retrieved chunk cannot establish

A contractor asks: **“Can I claim home-office equipment?”**



Semantically relevant. Correctly retrieved.

Does it apply to <u>this</u> person?

engagement type · contract terms · location · effective dates – **none of it is in the chunk**

Retrieval works as expected.

The missing piece is often a relationship, not another paragraph.

Tools: how an Agent gets and does anything

Tools let an agent reach structured records, unstructured content, and live state alike – and act on them.

get_account(id)	→	record	search_documents(q)	→	passages – Retrieval, with a signature on it
get_approval(id)	→	record	submit_claim(claim)	→	performs an action

MCP standardises how agents discover and use tools.

Tools and their permissions have to be designed carefully – enough for the tasks the agent is for, and no more.

The agent chooses what to fetch, and when.

What individual tool calls cannot establish

A contractor asks: **“Can I claim home-office equipment?”**

get_person("PER-3391")
→ returns the person · Workday

?

search_documents("home-office equipment")
→ returns Equipment policy v3 · §4 · Google Drive

Both calls succeeded. Both returned exactly what was asked for.

The person's engagement record decides which policy applies.
Neither call surfaced it – or hinted that it exists.

Tools expose access points. They return what you asked for. **The missing piece is the same one – what connects them.**

Finding the relationships, resolving the conflicts, deciding what applies: that is **left to your agent**.

How much of what an agent tells you about your organisation would you act on – without checking?

relationships nobody surfaced · a different answer each run · nothing to point at when asked why

Access alone is not understanding.

WSO2

What connected facts establish

A contractor asks: **“Can I claim home-office equipment?”**

Person · Workday

PER-3391

→

Engagement · Workday

ENG-2208

→

Contract type · Workday

contractor, agency-supplied

→

Governing policy · Google Drive

Contractor engagement policy · §7

effective from 2026-03-01

Contractor engagement policy · §7

"Equipment entitlements do not extend to non-employees."

What retrieval returned

~~Equipment policy v3 · §4~~

Not the governing policy for this person.

No single source holds the answer.

Records, documents, and the relationships between them, assembled for one question.

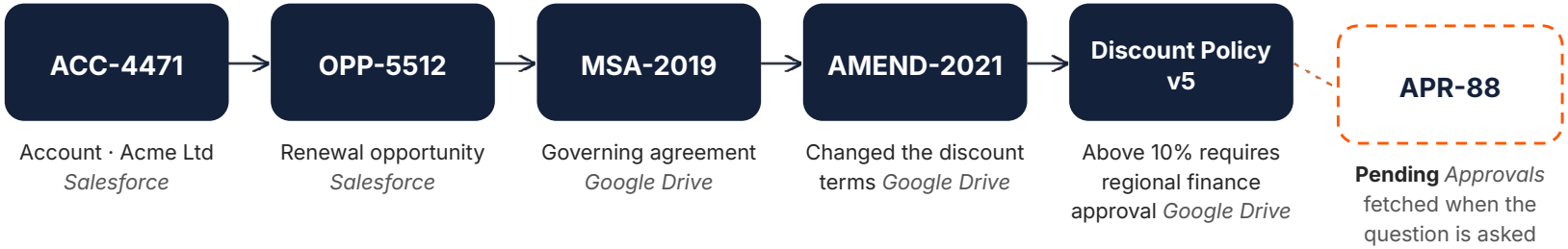
That's context.

One decision, six facts, three systems

“Can we offer Acme a 15% renewal discount?”

One decision, six facts, three systems

“Can we offer Acme a 15% renewal discount?”



The discount cannot yet be offered – approval is pending.

The question named one of these six. Nothing says when you've found the rest.

Runtime self-assembly has a reliability ceiling

Same question, two runs.

	Run A	Run B
Entity resolution	Resolves ACC-4471 → C-90021	Searches by account name
References	Finds the agreement, then the amendment	Finds the agreement, misses the amendment
Current state	Checks approval status	Assumes approval is not required
Outcome	**Do not offer – approval APR-88 is pending**	**Offer the discount**

Same question. Same tools. Same permissions. Different answer.

Run A did the work correctly, then discarded it. And no model, however good, can traverse a relationship nobody recorded or call an interface nobody built.

A familiar data problem, under a new kind of demand

WHAT YOU MAY ALREADY HAVE

warehouses · lakes · lakehouses · MDM · semantic layers

Bringing enterprise data together, and putting a consistent shape over it – for reporting, operations, and analysis someone specified in advance.

WHAT CARRIES OVER

entity resolution · shared definitions · ownership · systems of record · data contracts

Where this work is done, it carries over directly. Where it isn't, it surfaces here instead.

Those systems were designed around anticipated reports, models, and flows.

Agents introduce questions we did not know to specify at integration time.

A category is forming around this.

Context layers, by various names.

WSO2

How do you represent enterprise data so an agent can use it – and you can trust what comes back?

How do you represent enterprise data so an agent can use it – and you can trust what comes back?

Maintained relationships – drawn from records and documents across your systems, held outside any single run, and kept current as they change.

A graph is a good fit.
Knowledge Graphs have been doing this for decades, now put to new use.

How do you represent enterprise data so an agent can use it – and you can trust what comes back?

Maintained relationships – drawn from records and documents across your systems, held outside any single run, and kept current as they change.

CARRYING

Provenance

Which record or document it came from, and when it was observed

Supporting data

The passage or row that backs it

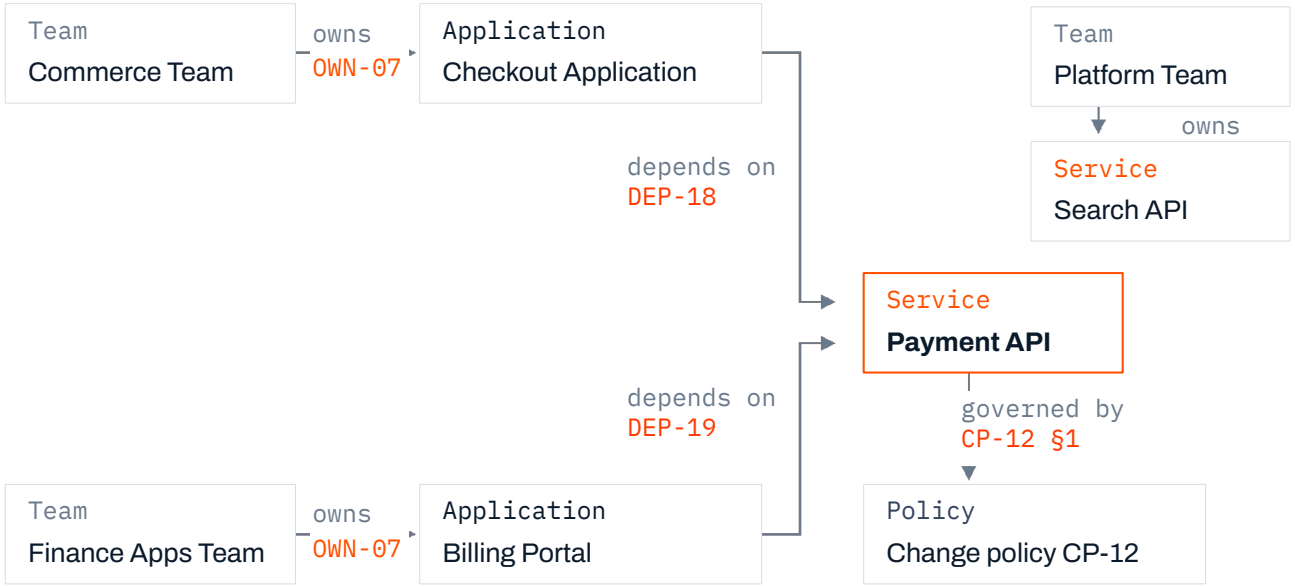
Decision traces

Which relationships and sources were used to assemble the context for a particular query

This is the basis for a **Context Graph**.

Enough to explain an answer, not just produce one.

The context graph



WHERE EACH RELATIONSHIP CAME FROM

DEP-18 DEP-19

Dependencies · service catalogue · observed 12 March 2026

OWN-07

Ownership · service catalogue · observed 12 March 2026

CP-12 §1

Policy · Google Drive · In force since 1 February 2026

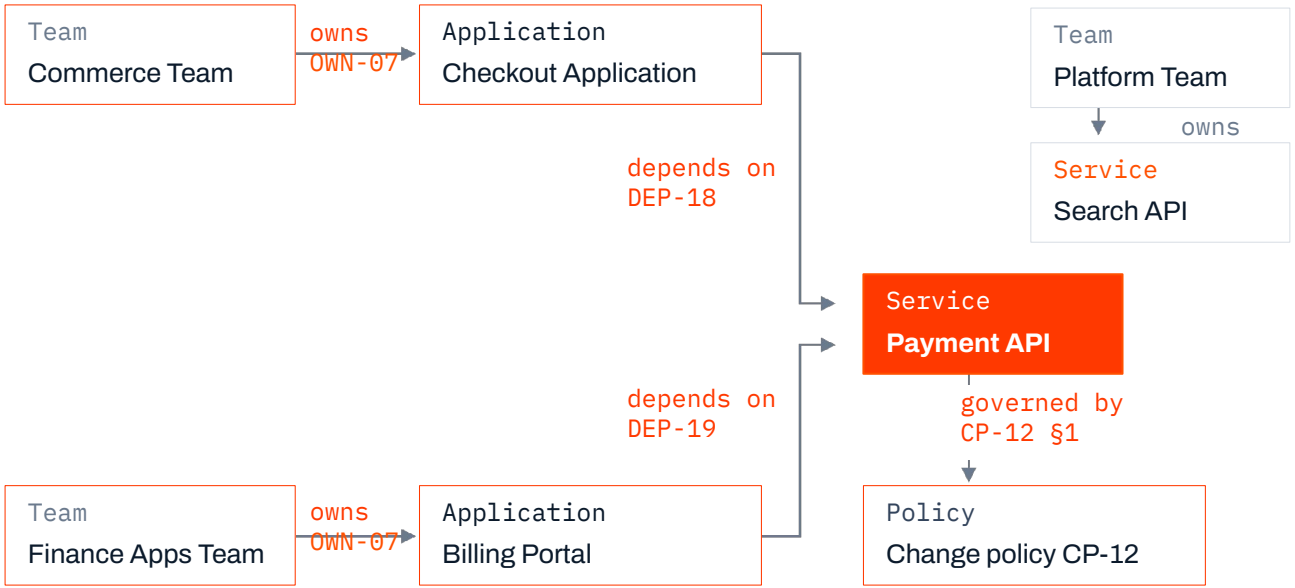
"This policy governs all production services and their dependent applications."

Maintained enterprise relationships. Nothing here was assembled for a question.

Relationships and references – not a copy of your records, and not your live operational state.

GraphRAG: retrieval guided by relationships

Who needs to sign off on a change to how payments are made?



1 **Ground** `vector index`
"how payments are made" → Payment API

2 **Traverse** `graph`
dependencies, ownership, governing policy

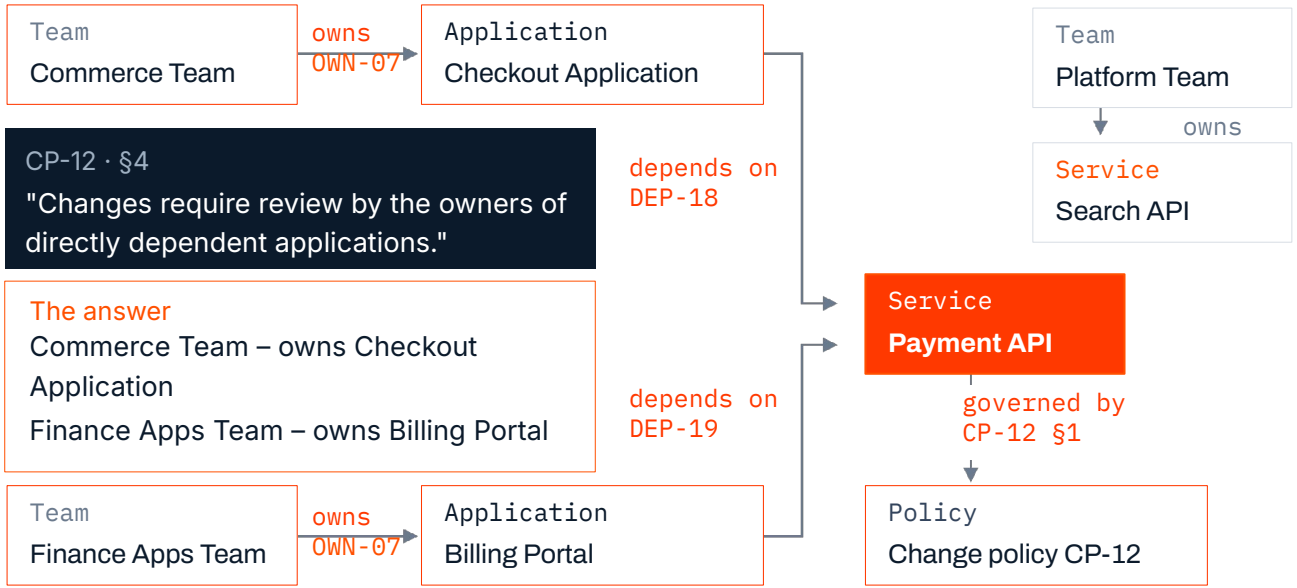
3 **Retrieve** `vector index`
the graph gave a reference, not the text. §4, out of fourteen sections.

4 **Assemble**
the subgraph(s) + §4 → the model

`vector index` `graph`

GraphRAG: retrieval guided by relationships

Who needs to sign off on a change to how payments are made?



1 **Ground** `vector index`
"how payments are made" → Payment API

2 **Traverse** `graph`
dependencies, ownership, governing policy

3 **Retrieve** `vector index`
the graph gave a reference, not the text. §4, out of fourteen sections.

4 **Assemble**
the subgraph(s) + §4 → the model

§4 is the same sentence for every service. The graph is what makes it an answer about this one.

`vector index` `graph`

Decision traces

Every relationship now says where it came from. **So does the context assembled for one question**.

decision trace · asked 18 September 2026

question	Who needs to sign off on a change to how payments are made?
assembled	Payment API · Checkout Application · Billing Portal · Commerce Team · Finance Apps Team · CP-12 §4
left out	Search API · Platform Team – not visited
as of	DEP-18 · DEP-19 · OWN-07 observed 12 Mar 2026 CP-12 §1 in force since 1 Feb 2026

That's a **decision trace**, kept with the graph.

When the response changes, the old one is still explainable.

Observability records what the system did. A decision trace records what the answer rested on.

How the graph is built

Relationships can be curated by hand, wired from systems that already hold them, or extracted from documents and records.

AI-assisted extraction is what made the third practical at enterprise scale.

One maintained representation – even an AI-generated one – beats each agent working it out alone.

Two runs see the same relationships – so an error shows up every time, instead of hiding in the variation.

Make it better

This is where the data work goes now – and how good it is sets the ceiling on what your agents can do.

Agree on the types – an ontology: which entities exist, what each relationship means, and where a term stops. Agreed once, shapes both what extraction recognises and what retrieval will traverse.

Is a contractor an Employee, or a separate type?

Start from what you have – MDM, shared definitions, catalogues, where available. They seed the graph, and they're what extraction gets checked against.

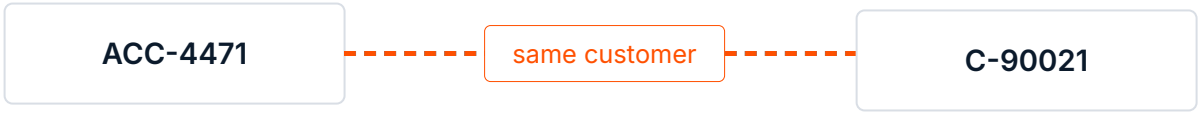
Is ACC-4471 the same customer as C-90021? Your MDM already decided.

Done once, it holds for every agent – and when something's wrong, one correction reaches all of them.

Validate what you can. What you can't, you can still trace.

What gets derived doesn't have to be discarded

Run A worked out that *ACC-4471* and *C-90021* were the same customer. Nothing in either system said so.



Then the run ended, and it was gone.

What one agent establishes, the next one starts from – with the question that produced it and the trace behind it.

Durable data and live evidence

Durable – maintained ahead of a request

agreements · organisational structure · customer and product data · policies

Live – fetched because a request needs it

approvals · balances · operational state · logs, metrics, traces

The boundary is not rigid – one source may provide both.

Some of what an answer needs can be maintained. Some is only true at the moment you ask. Assembling the proper context often needs both.

MSA-2019 and AMEND-2021 were durable. APR-88 was live.

How the layer assembles

Assembly spans a spectrum, from fixed-path to model-driven.

Fixed-path	The route is the same every time. What varies is how you reach the entity: a keyword match, or semantic grounding. GraphRAG we saw is the second.
Model-driven	A model decides what to ground to, what to traverse, and what to fetch – and whether to go again

All work from relationships already established, under one set of rules, and leave a trace.

Same question, different permissions

“Can PER-3391 claim home-office equipment?”

as hr-ops

Person	Engagement	Type	Policy
PER-3391 · Workday	ENG-2208 · Workday	contractor, agency-supplied	Contractor engagement policy · §7 Google Drive

→ **No – equipment entitlements don't extend to non-employees.**

as it-helpdesk

Person			Policy
PER-3391 · Workday			Equipment policy v3 · §4 Google Drive

→ **Cannot establish which policy applies.**

Assembled under the requester's authority, not the layer's. Less context – and no confident answer built on it.

Governance: what trustworthy context requires

Evidence

What each fact rests on

Provenance
the record, the system, when it was observed

Authoritative source
which system decides, when several hold the same field

Conflicts & gaps
where sources disagree, and what couldn't be established

Time

What was true, and when

Freshness
kept in step with the source, stamped when read

Temporal validity
when in force. Today's policy may not be the one that governed a decision in March.

History
superseded values kept, so a past answer still makes sense

Access

Who can see what

Requester's authority
not the layer's

Scope
Only what this request needs

Accountability

What the answer rested on, kept

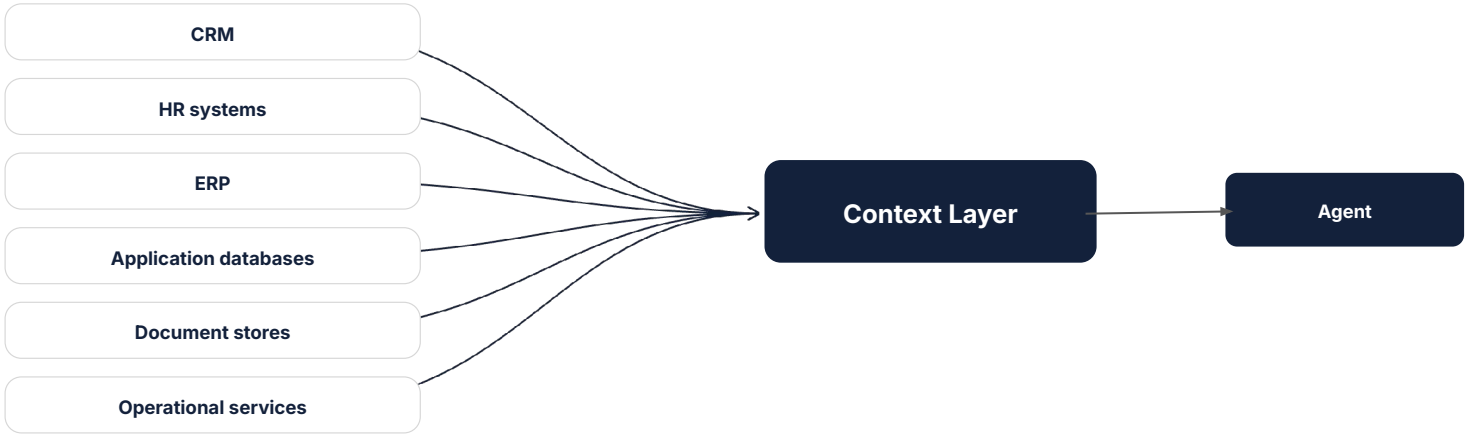
Decision trace
what was assembled for a particular query, as of when

These lines have to be drawn once. Drawn per agent, agreement is coincidence.

Focused agents on a maintained data layer

The data work moves out of the agent

No connections to each of your systems, no mappings between them, no knowing which policy governs what. It asks a question and gets governed context, with the evidence behind it.



The agent gets more focused on its core responsibilities. What it can reach becomes comprehensive – and accountable.

A data and integration problem, for open-ended work

It arrives as an AI problem. Most of it is a data and integration problem.

Connecting systems. Resolving entities. Enforcing policy. Governing data.

Similar work – for questions nobody specified.

Connected context lets an agent know more. Trust comes from governing how it's assembled and accessed, and seeing what its answer rested on.

How much of what an agent tells you about your organisation would you act on – without checking?

WSO2CON AFRICA

SEPT 22 - 24, 2026 | NAIROBI, KENYA

Thank You!

Maryam Ziyad

Senior Technical Lead