Presented by:
Agentic AI Foundation

19-20 NOVEMBER | NAIROBI, KENYA

MCP Dev Summit Nairobi

WHERE THE AGENTIC STACK IS BEING BUILT.

Ronnie Atuhaire

Beyond the Demo:
What Breaks When One AI Agent Talks to Multiple MCPs



The protocol is not the architecture.

The thesis for the next 25 minutes.

THE THESIS

MCP standardizes the connection layer. → **It does not solve the coordination layer.**

One agent. Five MCP servers.

A deceptively simple happy path.

HAPPY PATH

Gmail MCP
Find organizer email
Read thread

Calendar MCP
Find event
Check time

Drive MCP
Find talk notes
Read files

Docs MCP
Update briefing
Write summary

Developer Knowledge MCP
Verify technical claims
Official Google docs

Gemini + Google ADK
orchestrator

A single prompt hides five systems.

This is the demo we will build.

LIVE DEMO



> “Prepare me for my conference next week. Find the latest organizer email, check my calendar, locate my latest talk notes, verify the technical claims in my slides against current Google documentation, and update my speaker briefing.”

EMAIL	TIME	FILES	WRITE	VERIFY
Gmail	Calendar	Drive	Docs	Developer Knowledge

WHERE THE AGENTIC STACK IS BEING BUILT.

Then we added Universal Search MCP.

1 server — a tool surface

2 servers — orchestration

5 servers — a distributed workflow

6th server: search_corpus overlaps Gmail, Drive + Calendar search

MCP 2026-07-28: stateless protocol core ≠ stateless application

Identity • routing • partial failure • retries • application state • observability

Break #1 — Which tool?

Standardized tools do not mean standardized intent.

ROUTING

- Gmail → search_threads
- Drive → search_files
- Workspace Search → search_corpus
- Developer Knowledge → search_documents
- Developer Knowledge → answer_query

“Search for the latest MCP authentication documentation.”

Which search?
Which corpus?
Which authority?

Mitigation: namespace tools • filter the tool surface • choose the right corpus • retrieve progressively

Break #2 — Identity & authorization

One user does not mean one authorization boundary.

IDENTITY

User signed in	✓
Agentic OAuth client	✓
Required OAuth scope	?
Resource permission	?

CURRENT GOOGLE EXAMPLE

Since 11 Nov 2026, OAuth 2.0 clients calling Workspace MCP must be marked “Use this client for an AI-powered agent.”

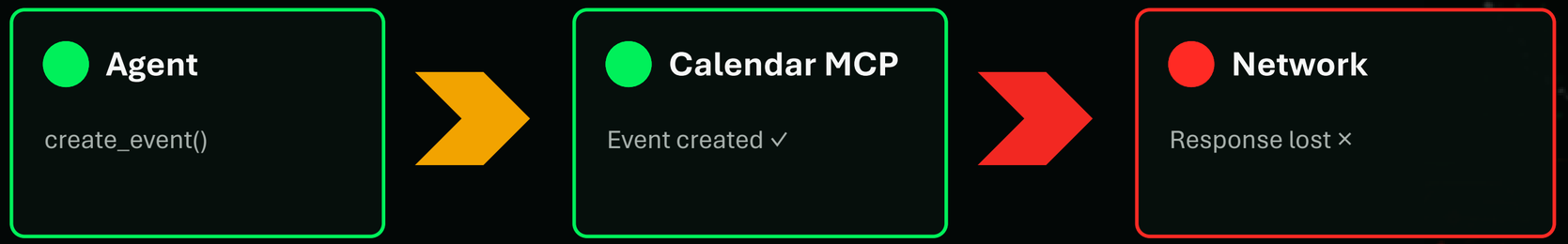
Google Workspace MCP • announced 12 Oct • enforced 11 Nov 2026

Auth failure policy matters: re-authenticate? continue partially? fail closed? mark the result as unverified?

Break #3 — “Did it actually happen?”

A timeout does not mean the action failed.

PARTIAL FAILURE



Agent retries

Risk: a retry can create a duplicate side effect. Mitigate with an execution ledger • dedup before retry • reconciliation • downstream idempotency keys where supported

create_event: idempotentHint = false

Break #4 — Context has a cost

More tools increase schema load, routing ambiguity and context pressure.

CONTEXT + LATENCY



Retrieve progressively.

Search → small relevant snippets → fetch full document only when required.

Google recommends search_documents first; full-document retrieval increases token use, latency and context-window pressure.

Break #5 — Untrusted content can steer powerful tools.

Reading can become control flow when the same agent can write.

TRUST + SIDE EFFECTS



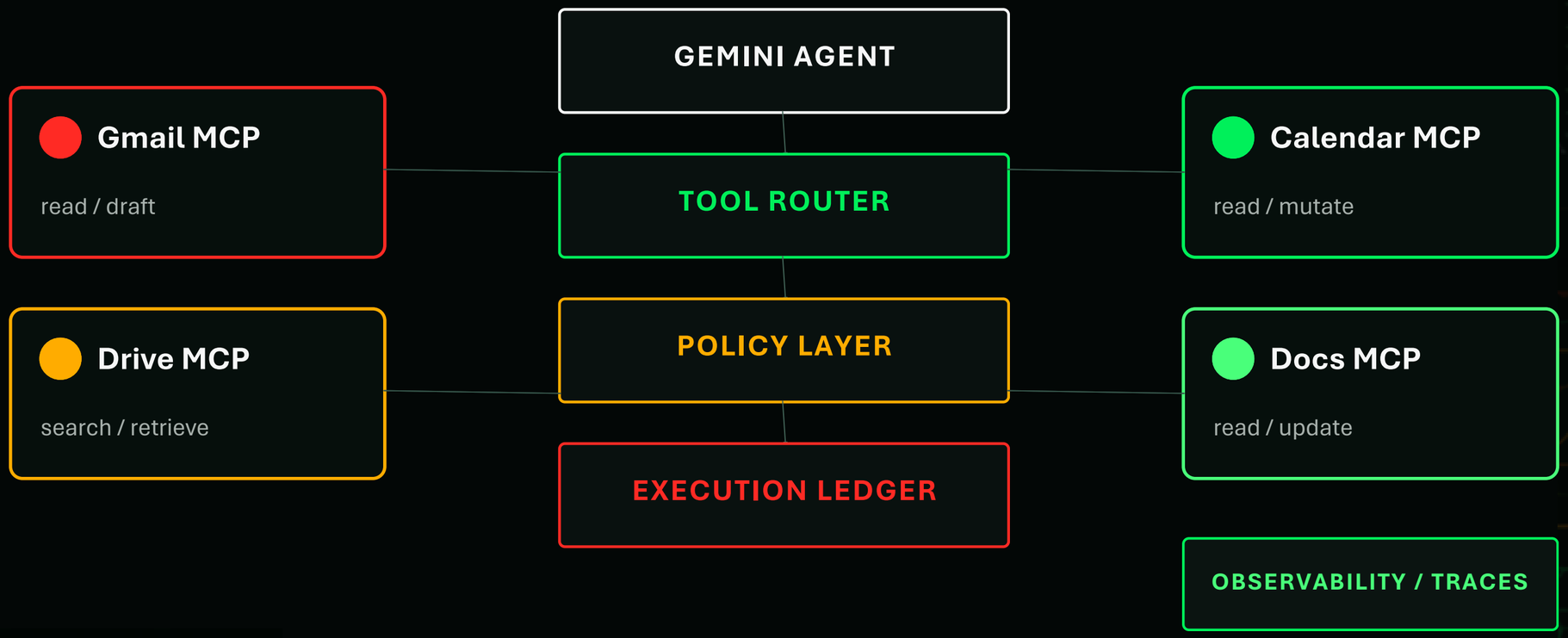
The model may infer intent. Deterministic policy decides permission.

Google requires Workspace MCP traffic to be screened for malicious content / prompt injection. Tool annotations help describe risk, but they remain untrusted hints.

A production pattern for multi-MCP agents

Put deterministic controls around probabilistic routing.

REFERENCE ARCHITECTURE



Six rules I would ship with

The minimum guardrails for a multi-MCP agent.

01 Discover less

Expose only the tools relevant to the current task.

02 Namespace everything

Use explicit prefixes: gmail_search_threads, drive_search_files.

03 Authorize per boundary

Identity, OAuth scope and resource permission are separate checks.

04 Separate reads from writes

Consequential mutations cross a policy or approval boundary.

05 Retry by effect, not by error

On timeout, reconcile unknown outcomes before repeating side effects.

06 Trace + reconcile

Propagate trace context and keep an execution ledger across servers.

MCP
Dev Summit
Nairobi

19-20 NOVEMBER | NAIROBI, KENYA

MCP standardizes interfaces. Engineering makes multi-server agents reliable

The interesting question is no longer:

“Can my agent call an MCP server?”

It is:

“Can I still understand, secure and trust it when it is calling ten of them?”



Demo stack & primary sources

Keep this as an appendix / backup slide.

- Google ADK
- Gemini
- Workspace MCP
- Developer Knowledge MCP
- MCP 2026-07-28

PRIMARY SOURCES

developers.google.com/workspace/guides/configure-mcp-servers

developers.google.com/workspace/guides/configure-mcp-security

developers.google.com/knowledge/mcp

adk.dev/tools-custom/mcp-tools/advanced/

blog.modelcontextprotocol.io/posts/2026-07-28/

developers.google.com/workspace/calendar/api/v3/reference/mcp/tools_list/create event

As of 17 Nov 2026: Workspace MCP = Developer Preview; Developer Knowledge MCP = GA; MCP 2026-07-28 remains current (next release target: 30 Mar 2027).

BACKUP / CODE

Remote MCP wiring — the repeatable pattern

Real current ADK APIs: Streamable HTTP, per-user headers, filtering and namespacing.

```
from google.adk.agents.readonly_context import ReadonlyContext
from google.adk.tools.mcp_tool import McpToolset, StreamableHTTPConnectionParams

async def google_headers(ctx: ReadonlyContext) -> dict[str, str]:
    # Token comes from your OAuth flow / token store.
    return {"Authorization": f"Bearer {ctx.state['google_access_token']}"}

def workspace(prefix, url, tools):
    return McpToolset(
        connection_params=StreamableHTTPConnectionParams(url=url),
        header_provider=google_headers,
        tool_name_prefix=f"{prefix}_",
        tool_filter=tools,
    )

gmail = workspace("gmail", "https://gmailmcp.googleapis.com/mcp/v1",
                  ["search_threads", "get_thread", "create_draft"])
drive = workspace("drive", "https://drivemcp.googleapis.com/mcp/v1",
                  ["search_files", "read_file_content"])
```

ADK uses tool_name_prefix to prevent collisions and tool_filter to reduce the tool surface.

Sources: adk.dev/tools-custom/mcp-tools/advanced • Google Workspace MCP configuration • verified 17 Nov 2026

BACKUP / CODE

Add the sixth server: Universal Search

The sixth server is Universal Search: one search tool can overlap Gmail, Drive and Calendar.

```
calendar = workspace("cal",
"https://calendarmcp.googleapis.com/mcp/v1",
["list_events", "get_event", "create_event"])

docs = workspace("docs",
"https://docsmcp.googleapis.com/mcp/v1",
["read_doc", "update_doc"])

workspace_search = workspace("ws",
"https://workspacemcp.googleapis.com/mcp/v1",
["search_corpus"])
```

```
knowledge = McpToolset(
connection_params=StreamableHTTPConnectionParams(
url="https://developerknowledge.googleapis.com/mcp",
headers={"X-Goog-Api-Key": os.environ["GOOGLE_API_KEY"]},
),
tool_name_prefix="gdk_",
tool_filter=["search_documents", "get_documents", "answer_query"],
)

agent = LlmAgent(
name="speaker_ops", model="gemini-3.8-flash",
tools=[gmail, drive, calendar, docs, knowledge],
)
# Add workspace_search after the happy path demo.
```

Sources: Google Workspace MCP endpoints • Universal Search search_corpus • Developer Knowledge MCP • Gemini API • verified 17 Nov 2026

BACKUP / CODE

Separate read surfaces from write surfaces

Use tool_filter for scope, tool_name_prefix for collisions, and confirmation for consequential tools.

```
calendar_read = McpToolset(
connection_params=calendar_conn,
header_provider=google_headers,
tool_name_prefix="calr_",
tool_filter=[
"list_events",
"get_event",
"suggest_time",
],
)
```

```
calendar_write = McpToolset(
connection_params=calendar_conn,
header_provider=google_headers,
tool_name_prefix="calw_",
tool_filter=[
"create_event",
"update_event",
"delete_event",
],
require_confirmation=True,
)
```

Do not make security decisions from MCP annotations alone: the spec defines them as untrusted hints unless the server is trusted.

Sources: ADK advanced MCP configuration • MCP Tool Annotations guidance • verified 17 Nov 2026

BACKUP / CODE

Timeout means UNKNOWN, not FAILED

Google Calendar create_event is currently non-idempotent; reconcile before retrying a side effect.

create_event annotations
Google Calendar MCP

readOnlyHint: false
destructiveHint: false
idempotentHint: false
openWorldHint: false

```
# Application pattern — not an MCP primitive
op_id = stable_operation_id(run_id, step_id)

if ledger.status(op_id) == "SUCCEEDED":
    return ledger.result(op_id)

try:
    result = await calw_create_event(**args)
    ledger.succeed(op_id, result)
except TimeoutError:
    ledger.mark(op_id, "UNKNOWN")
    # Query/reconcile before any retry
    raise NeedsReconciliation(op_id)
```

MCP has an idempotentHint risk vocabulary; it does not provide a universal cross-tool idempotency key.

Source: Google Calendar MCP create_event reference (updated 16 Nov 2026) • MCP tool annotations guidance

MCP
Dev Summit
Nairobi

19-20 NOVEMBER | NAIROBI, KENYA

Let’s Connect...

atuhaire.com/connect

