



MCP
Dev Summit
Seoul

Stop Wrapping APIs : Building Structured Execution Boundaries With MCP for Incident Triage

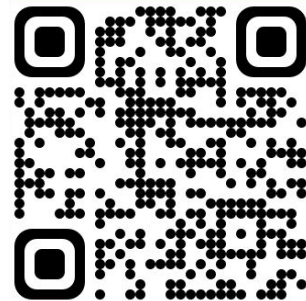
From fragmented infrastructure signals to structured operational verdicts

Sunyoung Park, KC-ML2



ML2

Hi, I'm Sunyoung Park 🖐️



Connect on LinkedIn
and our official Team

- working on internal LLM, RAG and Agentic systems as Research Engineer at **KC-ML2**
- **ML2** is a machine learning research team of **KC**, a semiconductor and display solutions company.

Starting point with our internal service : Mustard + SCv



Name	Mustard
Feature	Internal Slackbot
Goal	Agentic slack interface

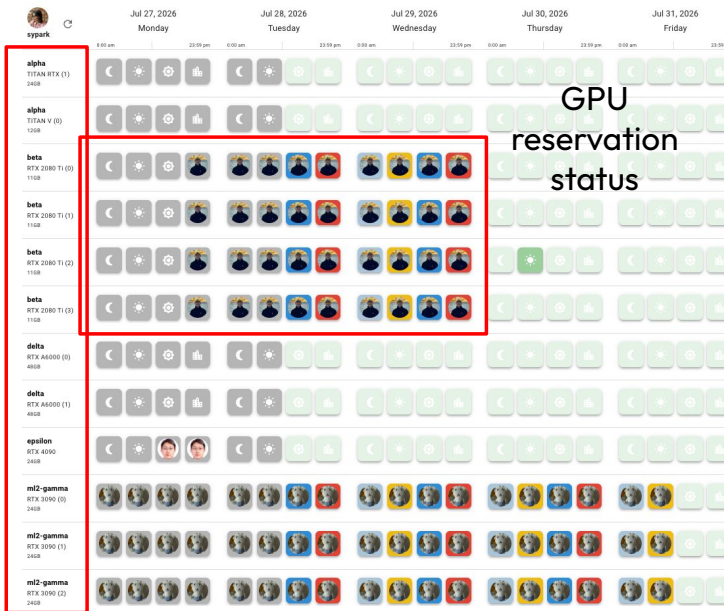


Name	SCv
Feature	Dashboards and signals for tracking server / GPU usage and health

By combining, move from dashboard-checking to natural language operational workflows

Fragmented signals

Our GPU Resources
(Servers,
Availabilities)



Violation alert via slack

7월 15일 수요일

A screenshot of a Slack conversation. The messages are from a bot named 'Mustard'. The first message is dated '오전 12:21' and says 'GPU Usage Violations - 1 found'. It lists a violation for 'sigma / NVIDIA GeForce RTX 4090' with the reason 'No reservation - @Sunyoung Park (syPark)'. The second message is dated '오전 12:31' and says 'GPU Usage Violations - 1 found'. It lists a violation for 'sigma / NVIDIA GeForce RTX 4090' with the reason 'No reservation - @Sunyoung Park (syPark)'. The third message is dated '오전 12:41' and says 'GPU Usage Violations - 1 found'. It lists a violation for 'sigma / NVIDIA GeForce RTX 4090' with the reason 'No reservation - @Sunyoung Park (syPark)'. The fourth message is dated '오전 12:51' and says 'GPU Usage Violations - 1 found'. It lists a violation for 'sigma / NVIDIA GeForce RTX 4090' with the reason 'No reservation - @Sunyoung Park (syPark)'. The messages are timestamped with dates like '2026-07-15 00:21:27 KST'.

+ historical reports

The information existed, but it was fragmented across interfaces and channels!

The workflow we wanted



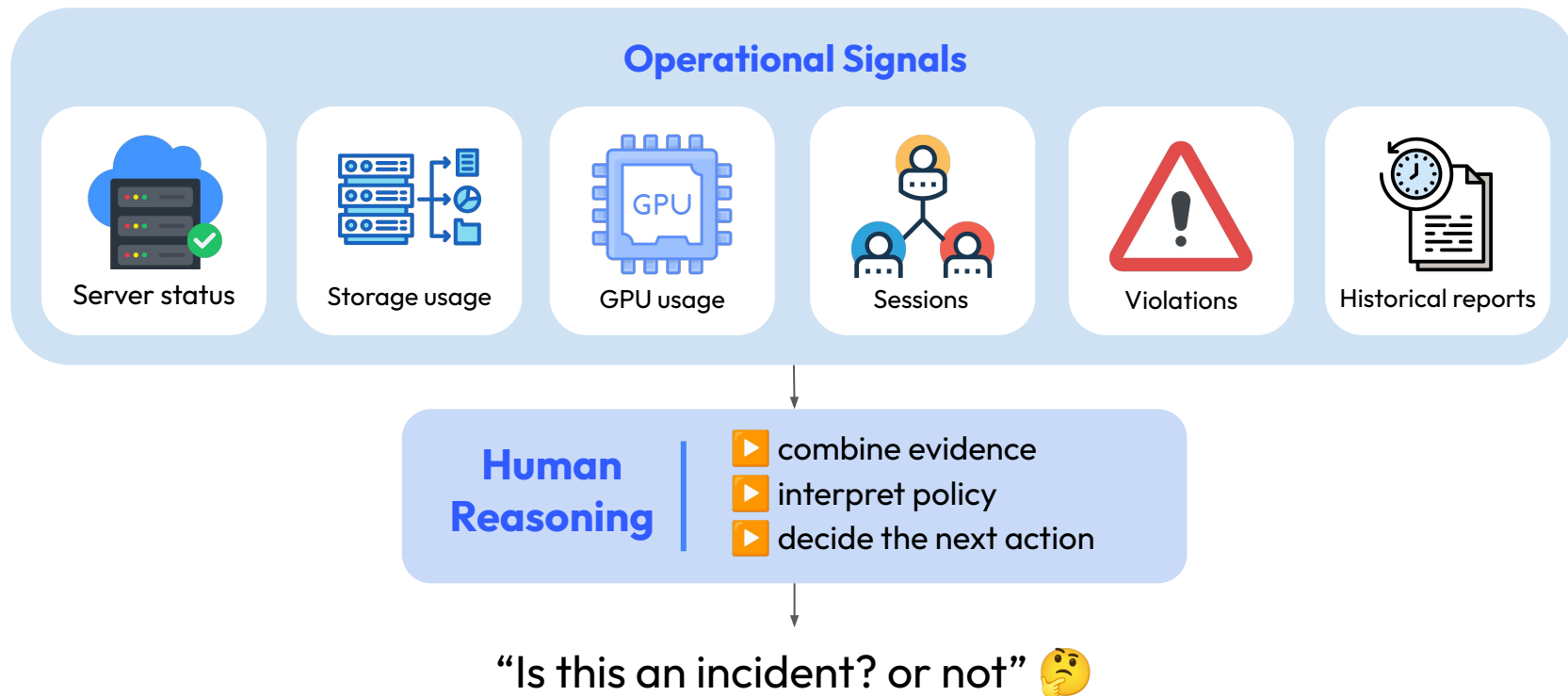
“Hey, Mustard. How is server-01 doing?”

“What happened?”

“What should I do next?”

- What we really need is
 - Is there a problem?
 - Why?
 - Then, what should we do next?
- SCV provides raw operational signals
- Design question : Who turns signals into a decision?

Human reasoning workflow



First Attempt : Thin wrappers

- A common first step : wrap existing APIs as MCP tools
- Our first MCP step was the same
- Easy to implement, test and explain

GET	/dashboard/api/servers	Api Servers
GET	/dashboard/api/reports	Api Reports
GET	/dashboard/api/sessions	Api Sessions
GET	/dashboard/api/usage	Api Usage

Existing APIs

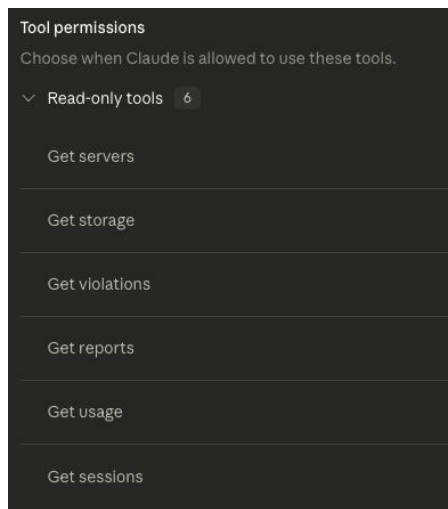


Thin wrappers

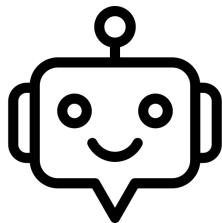
Tool permissions	
Choose when Claude is allowed to use these tools.	
Read-only tools	6
Get servers	
Get storage	
Get violations	
Get reports	
Get usage	
Get sessions	

MCP tools

The reasoning moved to the client LLM



Thin-wrapper MCP
(read-only tools)

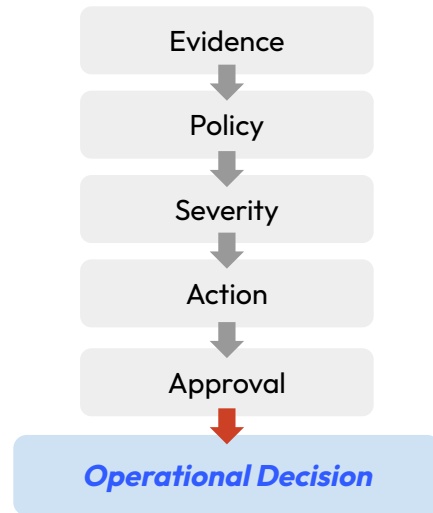


MCP client
(e.g., Claude, Cursor, ...)

*The client receives tools,
but not the operational logic*



Operational Reasoning



Thin wrappers did not remove **operational reasoning**.
They reassigned it to the **client model**

The first response looks reasonable

**Sunyoung Park** 2분 전

@Mustard Can you check the overall server status at 7/21 4:00PM UTC?
Based only on those measurements, which servers appear concerning, why, and what follow-up would you recommend?

✓ 1 🗨️

1개의 댓글

**Mustard** 에이전트 2분 전

@Sunyoung Park Here is the server status observed around 4:00 PM UTC on July 21, 2026:

- server-01 (Most Concerning)**
Why : It is critically stressed across all major resources.
CPU(92%): Indicates the machine is severely compute-bound, which will lead to significant latency or timeouts for any running jobs.
Storage (960 GB / 1 TB): At 96% capacity, this is an imminent risk of "disk full". If this hits 100%, it could corrupt training checkpoints or crash the OS.
Memory (88%): Approaching saturation; combined with high CPU, the system is operating at its braking points.

It offers ...

- CPU usage
- storage usage
- Memory usage

It looks reasonable... but 🤔

Missing decision



“Can you check the overall server status at 2026/07/21 4:00PM UTC?”

Which timestamp?
Which observation is valid?

Which evidence?
Which signals are required?

Which policy?
Which rule turns signals into a verdict?



Can we trust the verdict?



What action should we take?

The thin wrapper returned **observations**, not a **trusted operational decision**

What is an Execution Boundary?

- Execution Boundary
 - The responsibility line inside an operational workflow

Before

Raw signals ➡ Client LLM decides

After

Raw signals ➡ MCP server returns a checked operational result
➡ Client LLM explains

What does the boundary return?

Raw signals

- server storage
- gpu usage
- user reservation
- ... and so on



Public MCP tool surface

```
triage_server()
fleet_health()
investigate_timeline()
```



Checked operational result

```
{
  "outcome": "CRITICAL",
  "evidence": [...],
  "findings": [...],
  "actions": [
    {
      "approval_required": true,
      "executable": false
    }
  ]
}
```

Returns a **checked result**
with **suggested actions**

Checked result contains

Outcome examples

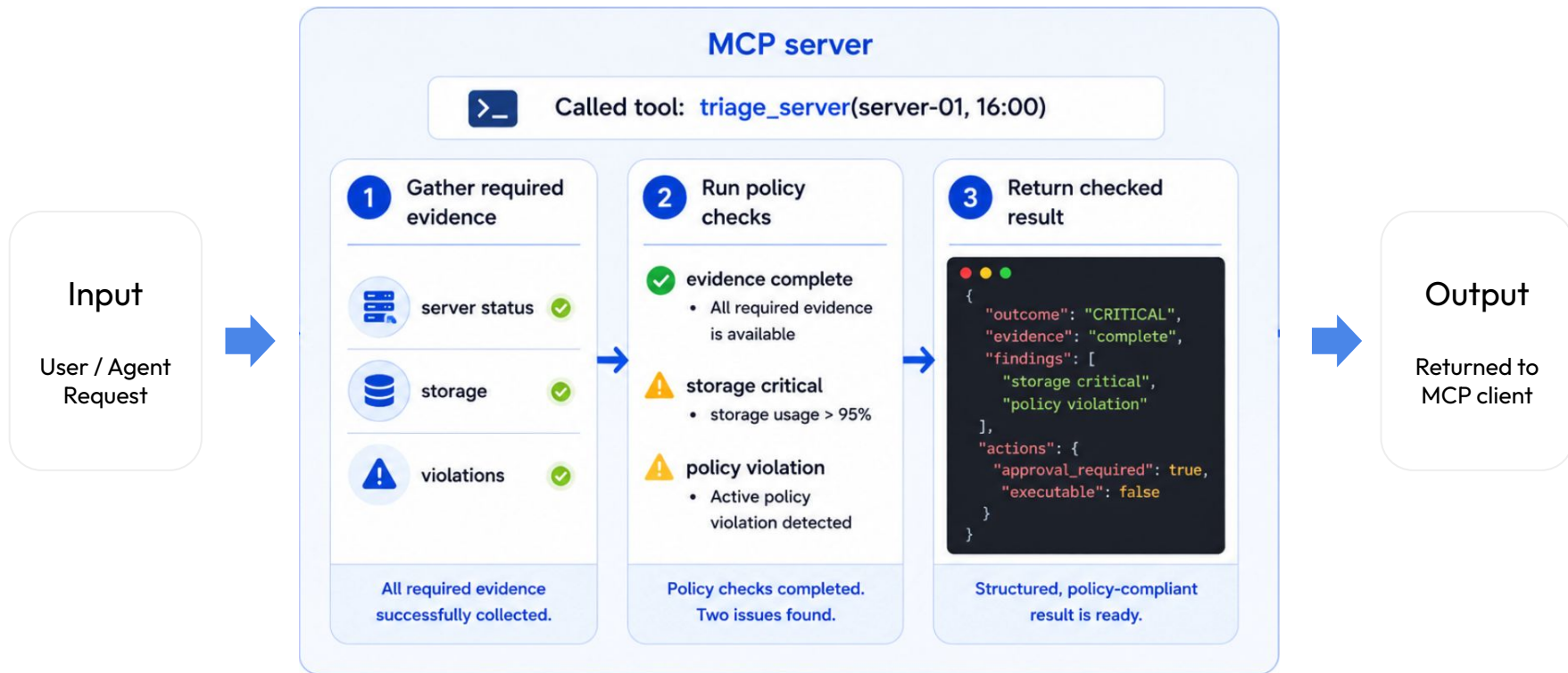
- **HEALTHY** : no issue found from the required evidence
- **CRITICAL** : action-worthy issue found
- **INDETERMINATE** : cannot safely conclude

Reasons

Suggested Actions

The execution boundary **defines** what the MCP server decides
before control returns to the client LLM

Inside the boundary



Comparison

Thin wrapping APIs



Sunyoung Park 2분 전

@Mustard Can you check the overall server status at 7/21 4:00PM UTC?

Based only on those measurements, which servers appear concerning, why, and what follow-up would you recommend?



1개의 댓글



Mustard 에이전트 2분 전

@Sunyoung Park Here is the server status observed around 4:00 PM UTC on July 21, 2026:

1. server-01 (Most Concerning)

Why : It is critically stressed across all major resources.

CPU(92%): Indicates the machine is severely compute-bound, which will lead to significant latency or timeouts for any running jobs.

Storage (960 GB /1 TB): At 96% capacity, this is an imminent risk of "disk full". If this hits 100%, it could corrupt training checkpoints or crash the OS.

Memory (88%): Approaching saturation; combined with high CPU, the system is operating at its braking points.

Execution boundary



Sunyoung Park Just now

@Mustard Can you check the overall server status at 2026/7/21 4:00PM UTC? and based on that measurements, which servers appear concerning, why, and what follow-up would you recommend?



3 replies



Mustard AGENT Just now

@Sunyoung Park At 2026/7/21 4:00PM UTC, the fleet health status is CRITICAL.

Concerning servers and reasons:

- 1 server is offline (out of 3 total).
- There are critical storage entries.
- There are high-severity violations.

Recommended follow-up:

- Escalate storage issues (approval required).
- Notify stakeholders (approval required). (edited)



Comparison - Thin wrapping APIs



Mustard 에이전트 2분 전

@Sunyoung Park ### Concerning Servers

server-01 (Most Concerning)

- **Why:** It is critically stressed across all major resources.
- **CPU (92%):** Indicates the machine is severely compute-bound, which will lead to significant latency or timeouts for any running jobs.
- **Storage (960 GB / 1 TB):** At 96% capacity, this is an imminent risk of "disk full" If this hits 100%, it could corrupt training checkpoints or crash the OS.
- **Memory (88%):** Approaching saturation: combined with high CPU, the system is operating at its breaking

Recommended Follow-up

1. Immediate Action (server-01):

- **Storage:** Immediately check for large, non-essential files or old experiment logs. **Move old data to cold storage (S3/GCS) to clear space below 80%.**
- **Compute:** "Identify which processes are driving the CPU usage. **If this is a production model inference endpoint, consider scaling horizontally or moving traffic to 'server-02.**

2. Schedule Maintenance (server-01):

- Because the CPU is so high, you may not be able to log in easily to perform cleanup. **If possible, schedule a brief downtime window to reboot or clean up the drive while load is lower (e.g., early morning).**

3. Monitoring (server-03):

- Set up a low-priority alert for when storage hits 70% **so you can prune data before it becomes urgent like server-01.**

The LLM evaluates each server independently from raw signals, without fleet-level context



Comparison - Thin wrapping APIs



Mustard 에이전트 2분 전

@Sunyoung Park ### Concerning Servers

server-01 (Most Concerning)

- **Why:** It is critically stressed across all major resources.
- **CPU (92%):** Indicates the machine is severely compute-bound, which will lead to significant latency or timeouts for any running jobs.
- **Storage (960 GB / 1 TB):** At 96% capacity, this is an imminent risk of "disk full" If this hits 100%, it could corrupt training checkpoints or crash the OS.
- **Memory (88%):** Approaching saturation: combined with high CPU, the system is operating at its breaking

Recommended Follow-up

1. Immediate Action (server-01):

- **Storage:** Immediately check for large, non-essential files or old experiment logs. Move old data to cold storage (S3/GCS) to clear space below 80%.
- **Compute:** "Identify which processes are driving the CPU usage. If this is a production model inference endpoint, consider scaling horizontally or moving traffic to 'server-02'.

2. Schedule Maintenance (server-01):

- Because the CPU is so high, you may not be able to log in easily to perform cleanup. If possible, schedule a brief downtime window to reboot or clean up the drive while load is lower (e.g., early morning).

3. Monitoring (server-03):

- Set up a low-priority alert for when storage hits 70% so you can prune data before it becomes urgent like server-01.

LLM's suggestion) If this is a production model inference endpoint, consider scaling horizontally or moving traffic to **server-02**.

Without fleet-level context, the LLM recommends moving traffic to an offline server.



This could confuse the operator when taking action



Comparison - Execution boundary

The execution boundary provides the context and policy needed for the decision

It gathers and validates the required evidence before producing a checked result



Fleet-level context, Checked result, Bounded actions 🕶️

Execution boundary



Sunyoung Park Just now

@Mustard Can you check the overall server status at 2026/07/21 4:00PM UTC?



1



1 reply



Mustard AGENT Just now

@Sunyoung Park The server fleet status as of 2026/07/21 4:00 PM UTC is CRITICAL.

- Total servers: 3

- Online: 2

- Offline: 1

Key findings include:

- STORAGE_CRITICAL_ENTRY (Critical severity)

- VIOLATION_PERIOD (High severity)

- SERVER_OFFLINE (High severity)

Recommended actions (require approval):

- ESCALATE_STORAGE

- NOTIFY_STAKEHOLDER (edited)

Consistent Result Across MCP clients

Slack MCP client + Qwen3.6-32B (quantized)

 **Sunyoung Park** Just now
 **@Mustard** how was the server fleet doing at 2026/07/21 4:00PM UTC??
 1 

1 reply

 **Mustard** AGENT Just now
 **@Sunyoung Park** At 2026/07/21 4:00 PM UTC, the server fleet was in a CRITICAL state.

- Total servers: 3
- Online: 2
- Offline: 1

Key findings included:

- Storage critical entry
- Violation period
- Server offline

Actions recommended were to escalate storage issues and notify stakeholders, both requiring approval. (edited)

Claude Desktop + Sonnet5

Can you check the overall server status at 2026/07/21 4:00PM UTC?

Loaded tools, used scv-triage integration

At 2026-07-21 16:00 UTC, fleet health came back **CRITICAL**. Here's the breakdown:

Servers: 2 online, 1 offline (out of 3 total)

Findings:

- **STORAGE_CRITICAL_ENTRY** — severity CRITICAL, affecting the fleet
- **VIOLATION_PERIOD** — severity HIGH, affecting the fleet
- **SERVER_OFFLINE** — severity HIGH, affecting the fleet

Suggested actions (require approval, not auto-executed):

- **ESCALATE_STORAGE**
- **NOTIFY_STAKEHOLDER**

Note: this is a categorical severity snapshot, not a confirmed root cause — it flags a critical storage entry and a server offline during this window, but doesn't pinpoint the exact cause. Want me to check a specific server's timeline around this window for more detail?

Cursor + Composer 2.5 Fast

Thought briefly

At 2026-07-21 16:00 UTC (4:00 PM UTC), the server fleet was in **CRITICAL** condition.

Summary

Metric	Value
Total servers	3
Online	2
Offline	1
Overall outcome	CRITICAL

Findings

1. **STORAGE_CRITICAL_ENTRY (CRITICAL)** — Storage had entered a critical state.
2. **VIOLATION_PERIOD (HIGH)** — A policy violation period was detected.
3. **SERVER_OFFLINE (HIGH)** — One server was offline.

Evidence status

All required evidence sources were fully collected:

- Servers — COMPLETE
- Storage — COMPLETE
- Violations — COMPLETE

Recommended actions (approval required, not auto-executable)

 Send follow-up

Composer 2.5 Fast 

Different clients, Same boundary. Consistent result



Let's run the execution boundary



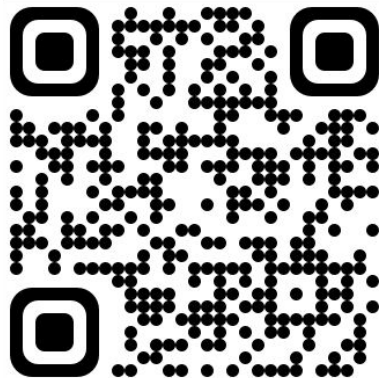
Takeaways

- Exposing tools is not the same as defining a workflow
- Operational judgment should live in a reusable execution boundary
- Return bounded verdicts and constrained actions. not just raw evidence

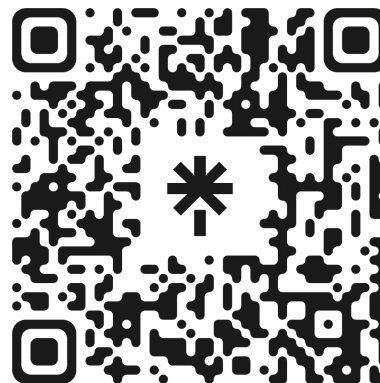
Stop asking “What tools should I expose?”

Start asking “What decision should this boundary make?”

Contact



Full DEMO code for this session



Additional Info of ML2

Feel free to reach out... and give any comments for DEMO!