

Small Models, Big Toolboxes
Designing MCP for on-device and low-compute agents

Oluwaseun Nifemi, Technical Delivery Lead, Data Science Nigeria
Olawale Ahmed, AI Product Technical Lead, EqualyzAI



Agenda

01 Tool abundance is not free

02 From 100 tools to the right 5

03 Cheaper schemas

04 Local and remote models

05 Worked example and checklist

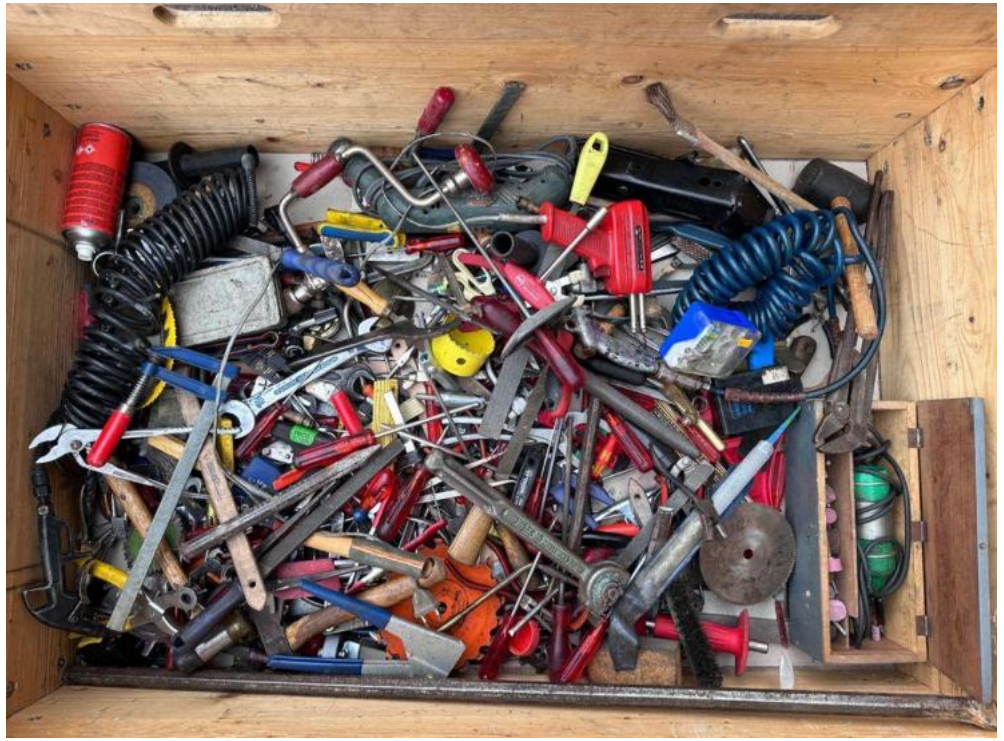
06 Questions

Tool abundance is not free

Adding tools to MCP is easy. Small models pay for everyone.

Important questions to ask:

1. **Does the model have enough room to process all the information?**
2. **Can it distinguish between similar tools?**
3. **Does the current task actually require all those tools?**



Four budgets every tool spends

1

Context

Tool descriptions fill the prompt.

Example: **The model receives 50+ lengthy tool descriptions before it even reads the parent's question.**

2

Memory

Longer prompts use more RAM.

Example: A low-memory phone struggles to run the model alongside the rest of the application



Accuracy

Similar tools cause wrong picks.

Example: It selects a tool for viewing attendance instead of a tool for retrieving examination results

4

Latency

More text, slower replies.

Example: The parent waits longer for a result because the assistant has to process unnecessary tool descriptions.

3

From 100 tools to the right 5

Minimum tool surface

Build tools for tasks.

Contextual loading

Show this step's tools only.

Domain routing

Pick the area, then its tools.



Hierarchical discovery

List first, details on request.

Pre-routing

Rules or classifier before the model.

Comparing the patterns

Pattern	When to use	Model sees	Cost/Trade off
Static	Few tools	All tools	Context, accuracy
Contextual loading	Clear steps	This step's tools	Client tracks state
Domain routing	Distinct areas	One area's tools	Router can miss
Hierarchical discovery	Big catalogues	Index, then one schema	More calls
Pre-routing	Predictable requests	One tool or none	Rules to maintain

Hierarchical discovery in four calls

Start small. Fetch details only when needed.

list_domains

Short list of areas.

01

list_tools

Tool names for one area.

02

describe_tool

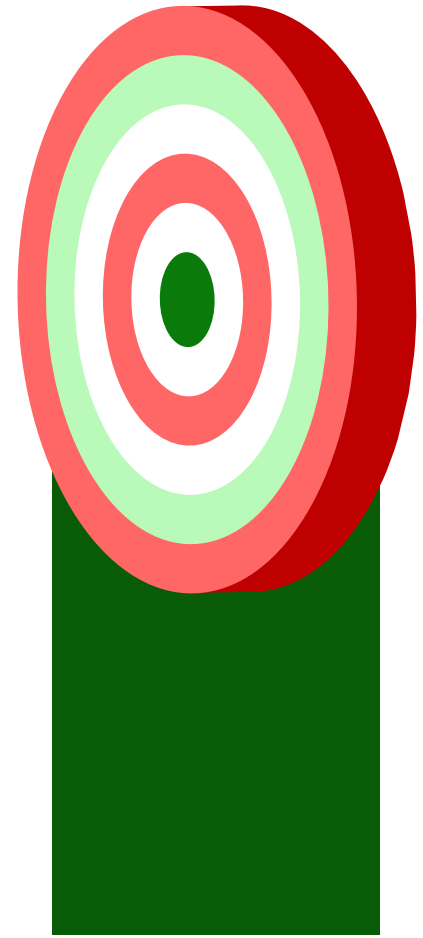
Full details for one tool.

03

call_tool

Server checks the inputs.

04



Cheaper schemas

Before

```
name: getStudentTermResultInfo
description: "This tool can be used
  to retrieve comprehensive
  information about a student's
  academic performance..."
params: studentId?, admissionNo?,
  termName (free text), year?,
  includeDetails?, format?
```

After

```
name: term_report
description: "One student,
  one term."

params:
  student_id    (required)
  term: First | Second | Third
  year          (optional)
```

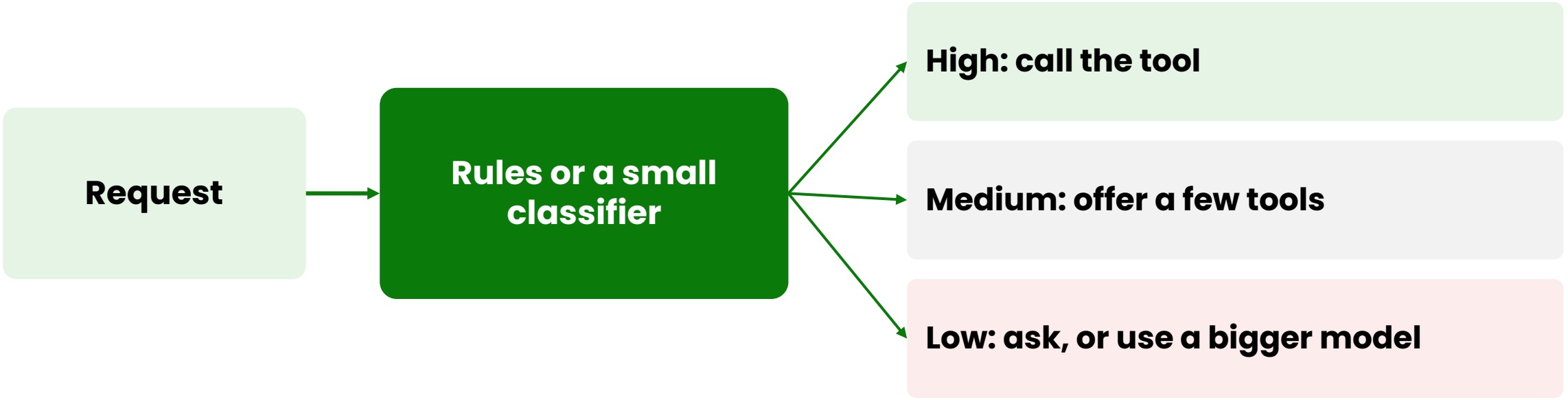
One job per tool

Enums, not free text

Few optional fields

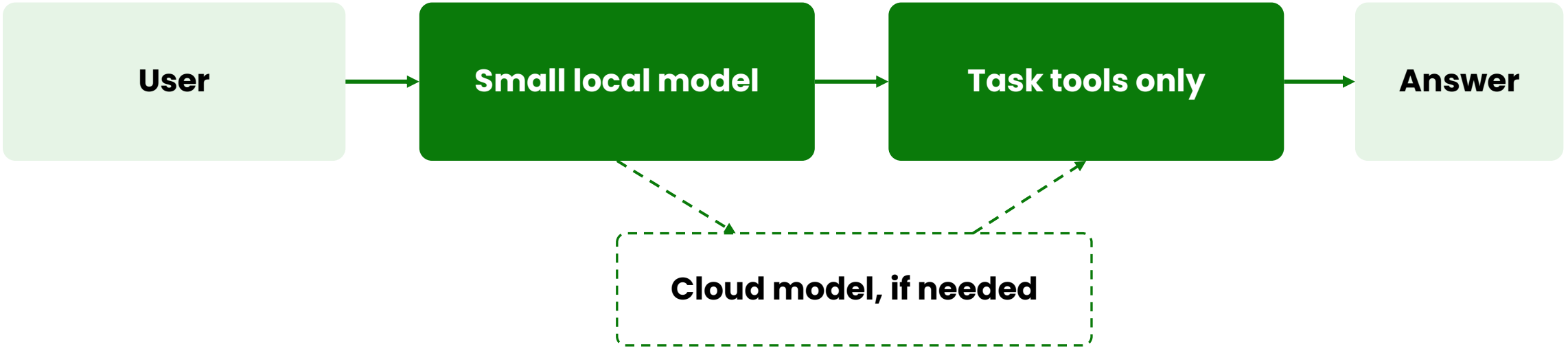
Say when to use it

Pre-routing and confidence



Set thresholds from real data. Log every decision.

Hybrid Architecture for Local and remote models



Escalate when unsure

Send only what is needed

Same tools for both

A system designed for low-connectivity environments

Cloud model
Full service

Mode 1

Small model
Fewer tools

Mode 2

Rules only
Menus and forms

Mode 3

Queue
Confirm later

Mode 4

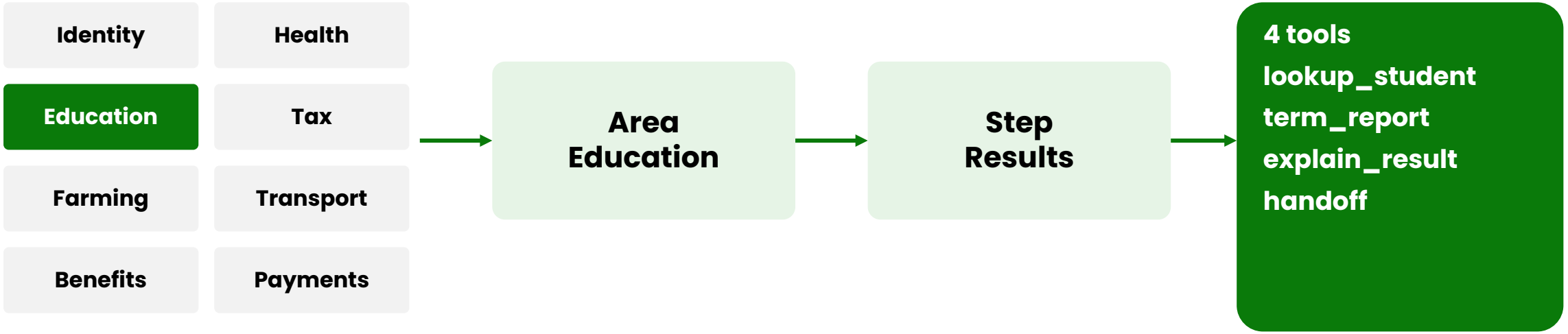
Test every level

Tell users the mode

Hold actions until confirmed

Worked example: citizen services

Task: a parent asks for a term result



Then test again with a smaller model and a weaker network.

What to measure

Same tasks. Change the tools, the model and the network.

Right tool?	**Right inputs?**	**Prompt size**
Speed on device	**Fallback rate**	**Cost of a wrong call**

Checklist

1. **Count the cost**
   Measure tool tokens.
2. **Scope to the task**
   This step's tools only.
3. **Route first**
   Rules for clear requests.
4. **Cheap schemas**
   Short names, enums.
5. **Escalate on purpose**
   Use a threshold.
6. **Test every mode**
   Including offline.

Thank You

- Data Science Nigeria
- DSNai - Data Science Nigeria
- Dsn_ai_network
- DSNai_Data Science Nigeria/ Data Scientists Network
- www.datasciencenigeria.org
- www.dsn.foundation

