workato

HOW WE MCP | PART 3 OF 4

First *Principles*

The MCP Servers that work consistently follow four principles.
The ones that don't violate them, often in subtle ways.

01	02	03	04
CLARITY over completeness	**COHESION** over convenience	**EXPLICITNESS** over inference	**DETERMINISM** over cleverness

01 Clarity over completeness

More tools don't add capability — they add confusion. When multiple tools look valid, selection becomes probabilistic. Small phrasing changes produce different choices.

BAD	GOOD
search_records	search_records
find_records	(criteria)
query_records	get_record (by ID)

If multiple tools look valid, the model will guess.

02 Cohesion over convenience

Mixing unrelated workflows in one server means the same request can map to multiple interpretations. The LLM has no clear context — so it guesses.

BAD—ONE SERVER	GOOD—SEPARATE
search_opportunities	Sales server
create_support_ticket	Support server
manage_users	Billing server

If a request can mean multiple things, your scope is too broad.

03 Explicitness over inference

Hidden side effects, undocumented limits, missing prerequisites — if they're not stated, the model assumes incorrectly. What's obvious to you is invisible to the model.

BAD—HIDDEN EFFECTS	GOOD—EXPLICIT
update_order(id, status)	update_order_status
+ sends email	send_order_notification
+ updates inventory	process_payment

If it's not stated, the model has to guess.

04 Determinism over cleverness

If the same inputs produce different outputs, the model can't form stable expectations. Variation is fine — hidden variation is not. Make it vary on explicit inputs, not hidden context.

BAD—HIDDEN VARIANCE	GOOD—EXPLICIT
get_customer(id)	get_customer(id, level)
fields vary by role/state	summary | standard | full

Determinism isn't removing variation, it's making variation predictable.

*All four principles do the same thing: they **reduce ambiguity.***
When ambiguity goes down, reliability goes up.