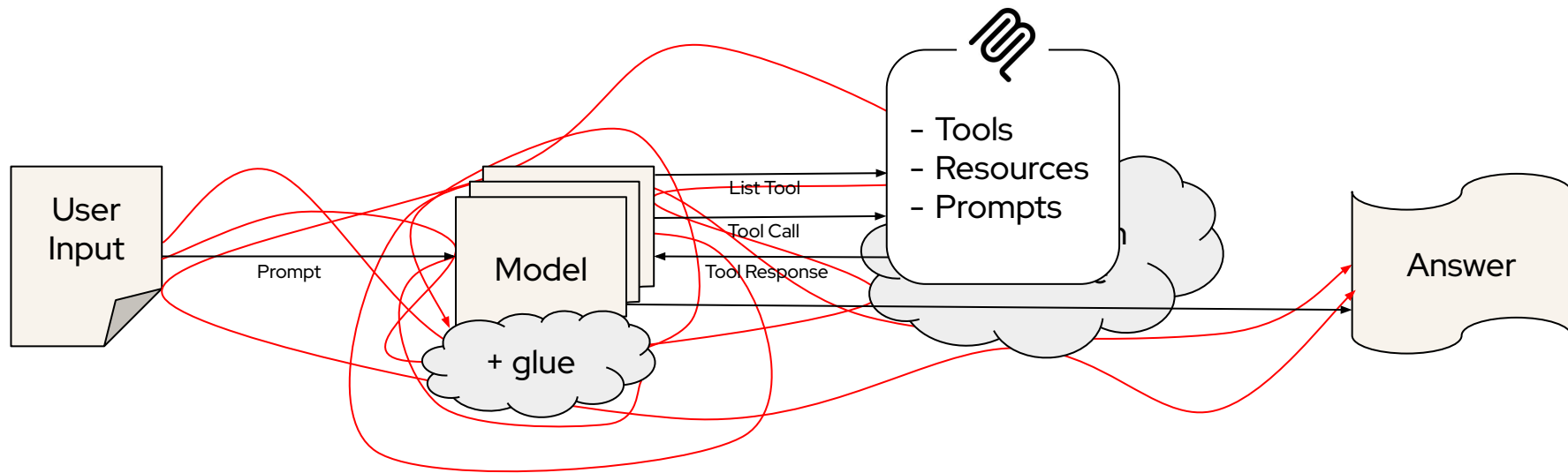


Self-Improving MCP Agents

Kemal Elmizan
GoTo, Indonesia



"Familiar Analogies" in Uncharted Territory



Blurred Boundaries

- Which API was called?
- Why did the model choose that action?
- Was that a prompt failure or integration failure?

The gap between intent and tool invocation

Fix the production
outage

Prepare me for
tomorrow's meeting

Onboard a new
Employee



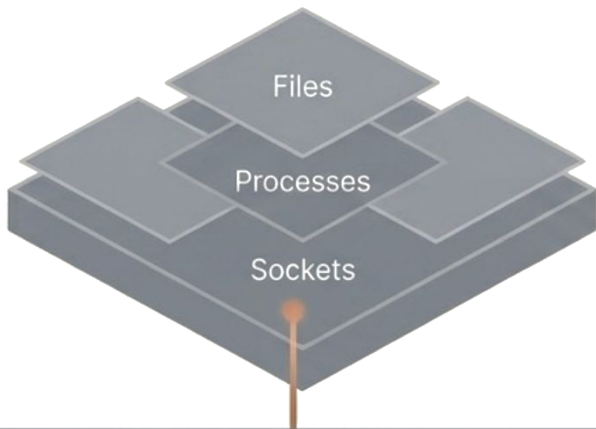
`read_logs()`

`search_documents()`

`create_ticket()`

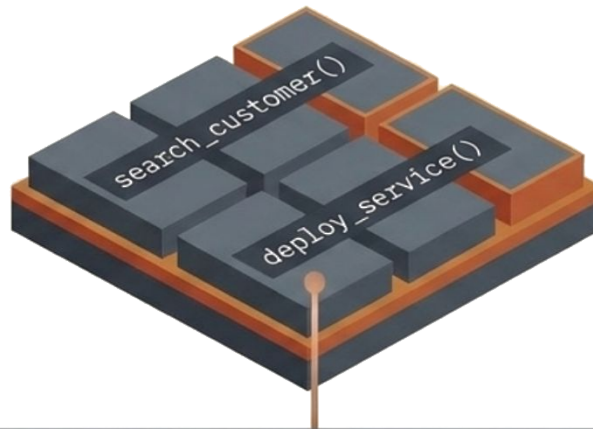
Operating-system primitives vs application primitives

Operating System Primitives



The OS knows a process wants to execute `kubectl apply`. It lacks semantics.

Application Primitives



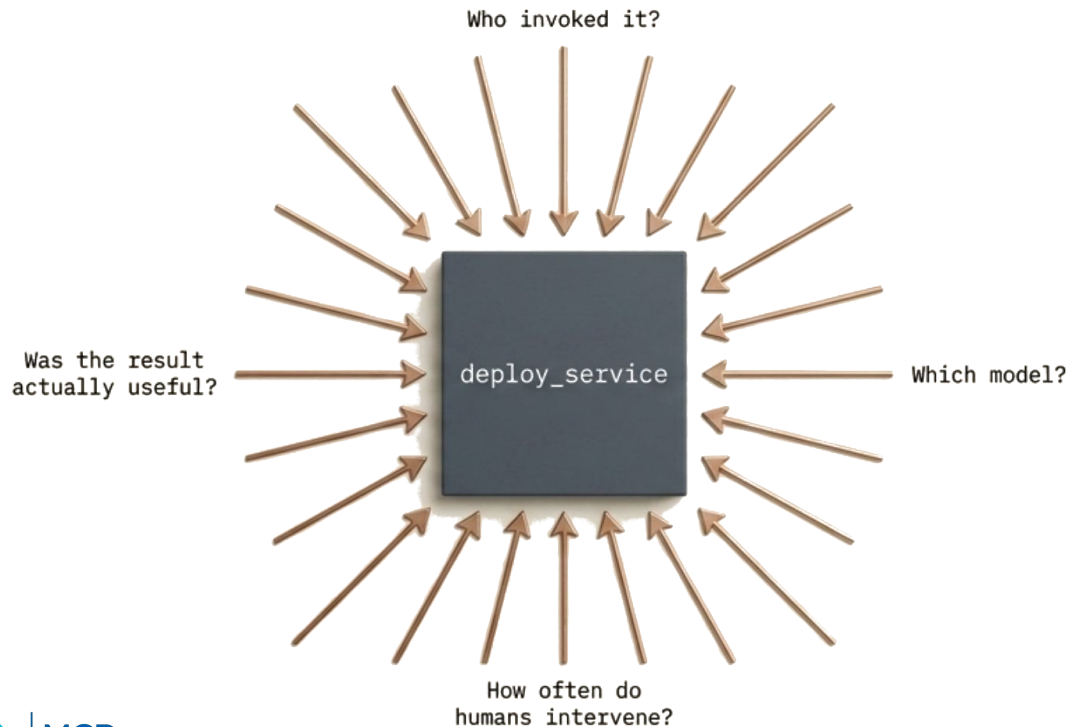
Provides the intentional semantics to tell us if the agent is refunding a customer or fixing production.



Abstractions too static it struggles to "self-improve"

Hierarchies	Workflows	Static Metadata
 <i>Intent -> Workflow -> Skill</i> -> create_user()	search_customer() -> create_ticket()	 Hardcoding risk: high in YAML.
 Intuitive mapping for human logic.	 Deterministic execution, reproducibility, and cost control.	 Attaches naturally to existing tool boundaries.
 Organizations evolve; hierarchies become rigid org charts struggling with overlapping skills.	 Captures how things worked in the past, failing entirely on novel, unseen problems.	 Risk and ownership are not fixed. Static metadata just becomes dead documentation.

"Adaptive", focuses on runtime



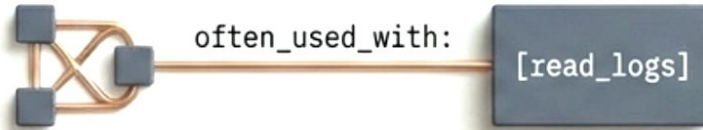
```
tool:  
  name: deploy_service  
  
insights:  
  preferred_model: gpt-5-mini  
  
  often_used_with:  
    - read_logs  
  
  human_review_frequency: 82%  
  
  observed_failure_rate: 12%
```

TOOL: `deploy_service`

Tags

`preferred_model: gpt-5-mini`

Connections



Metrics



human_review_frequency: 82%



observed_failure_rate: 12%

"Collecting Evidence"

Log inspector

```
EVENT: tool_invocation  
tool: deploy_service  
model: gpt-5  
success: false  
duration_ms: 1320  
human_override: true
```

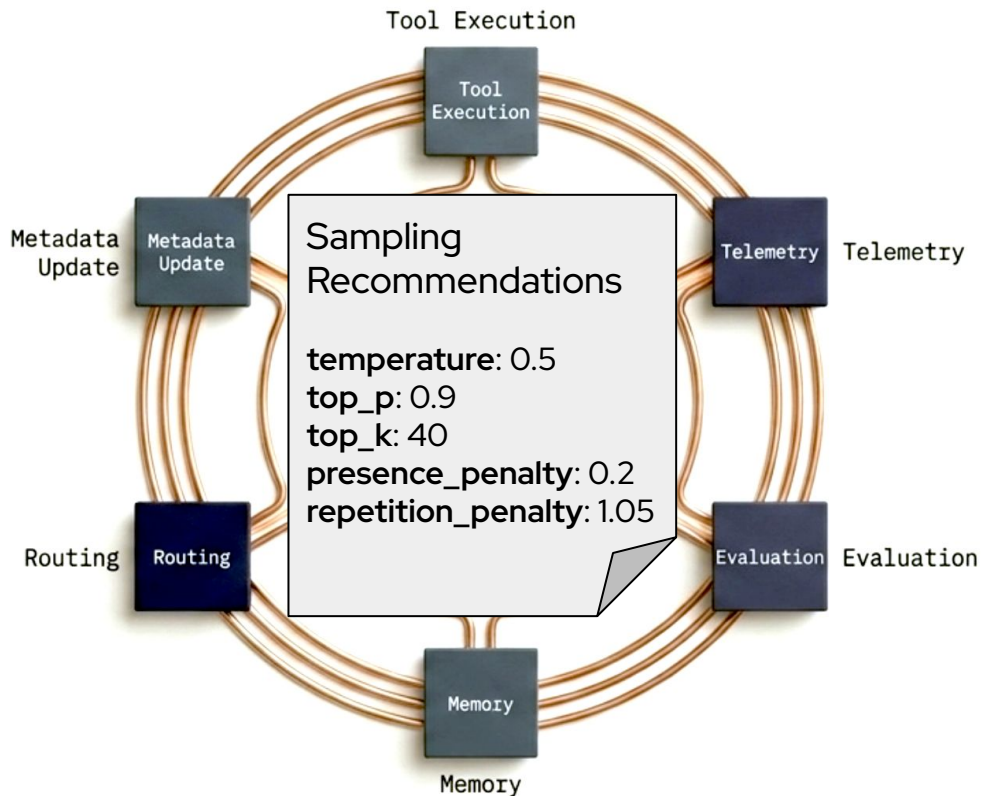
Which tools fail
most often?

Which are
expensive?

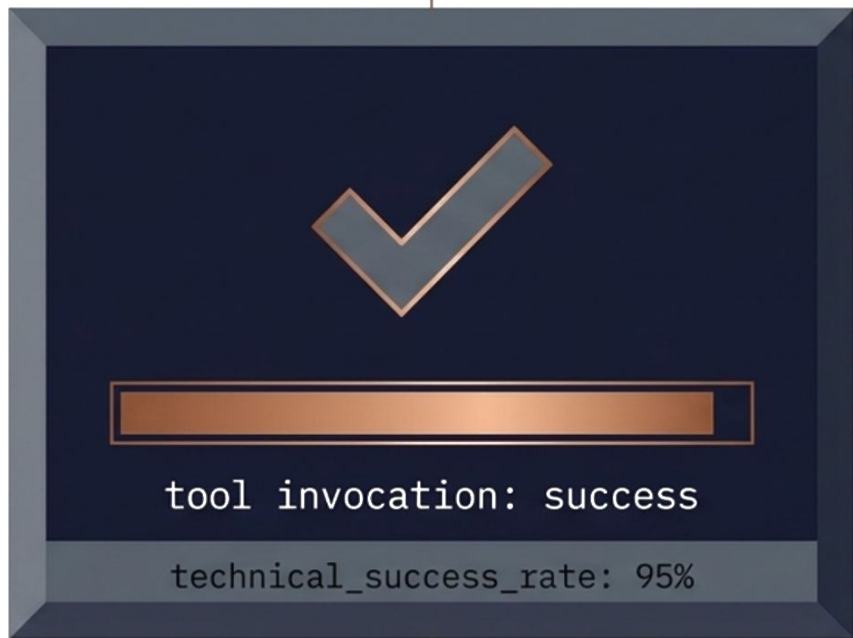
Which require
human intervention?

Which tools tend to
be used together?

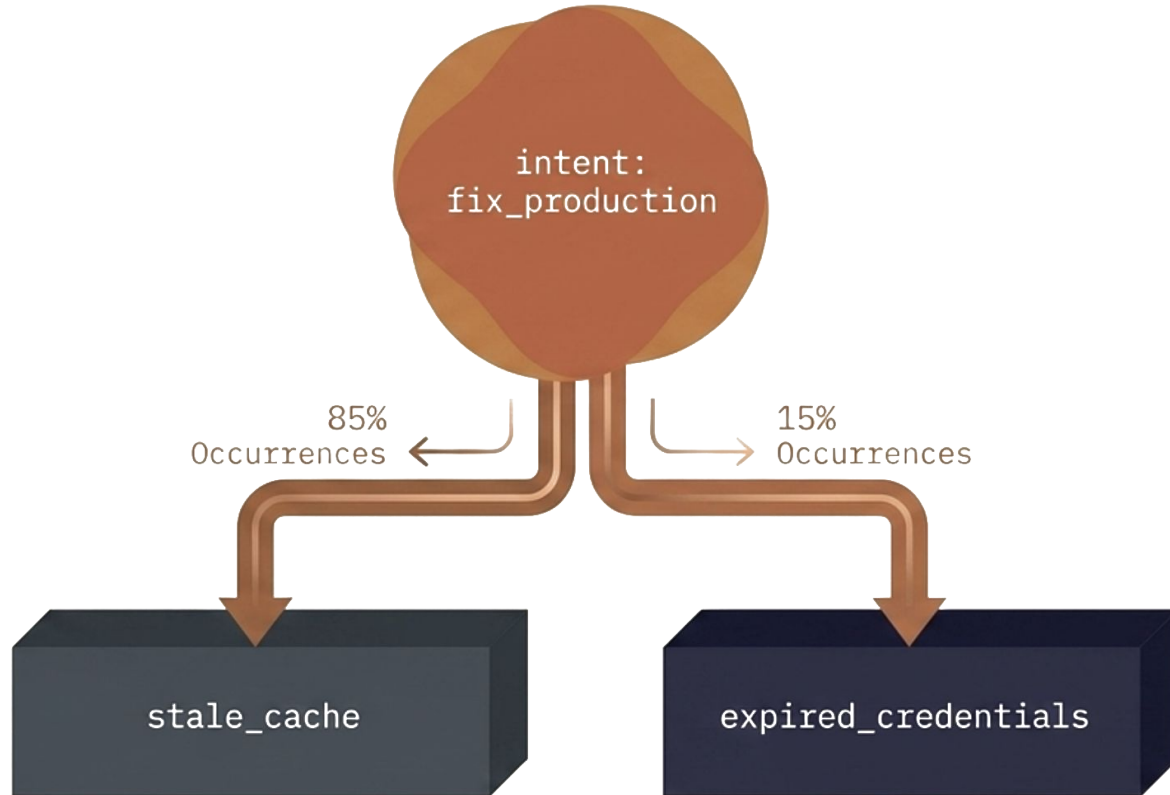
Continuous Adaptation in Middleware



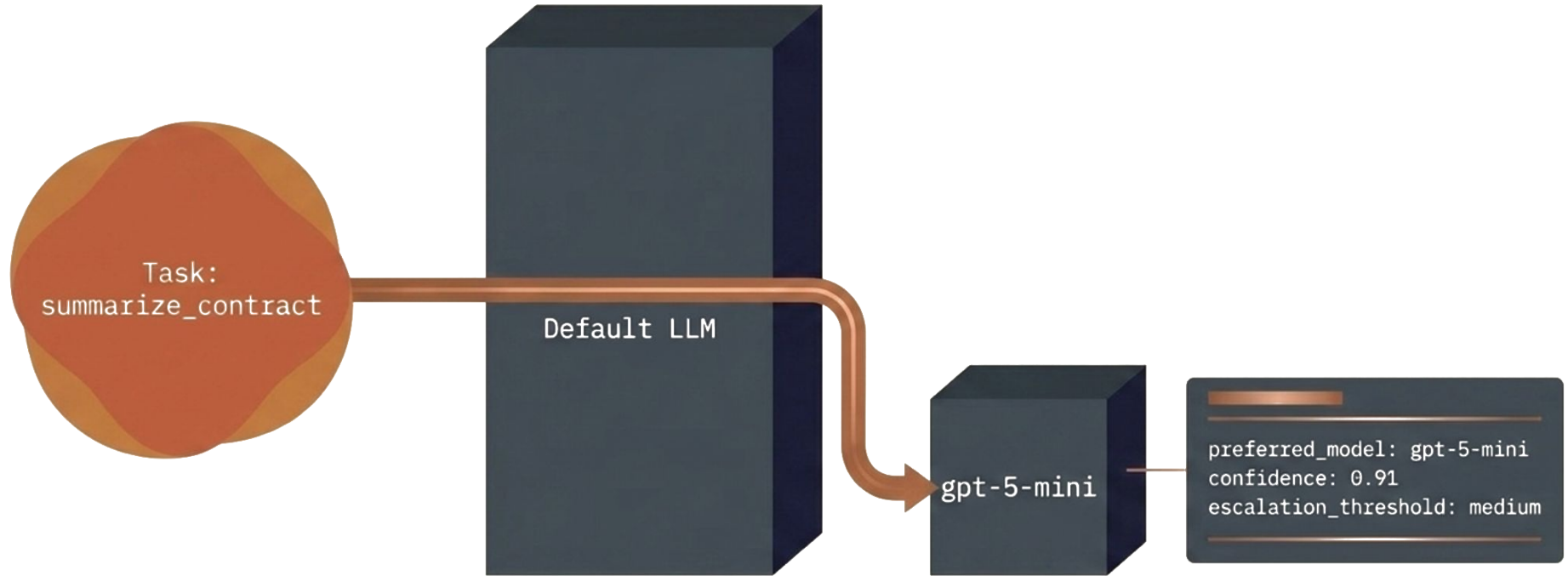
Scoring Technical Execution + Human Satisfaction



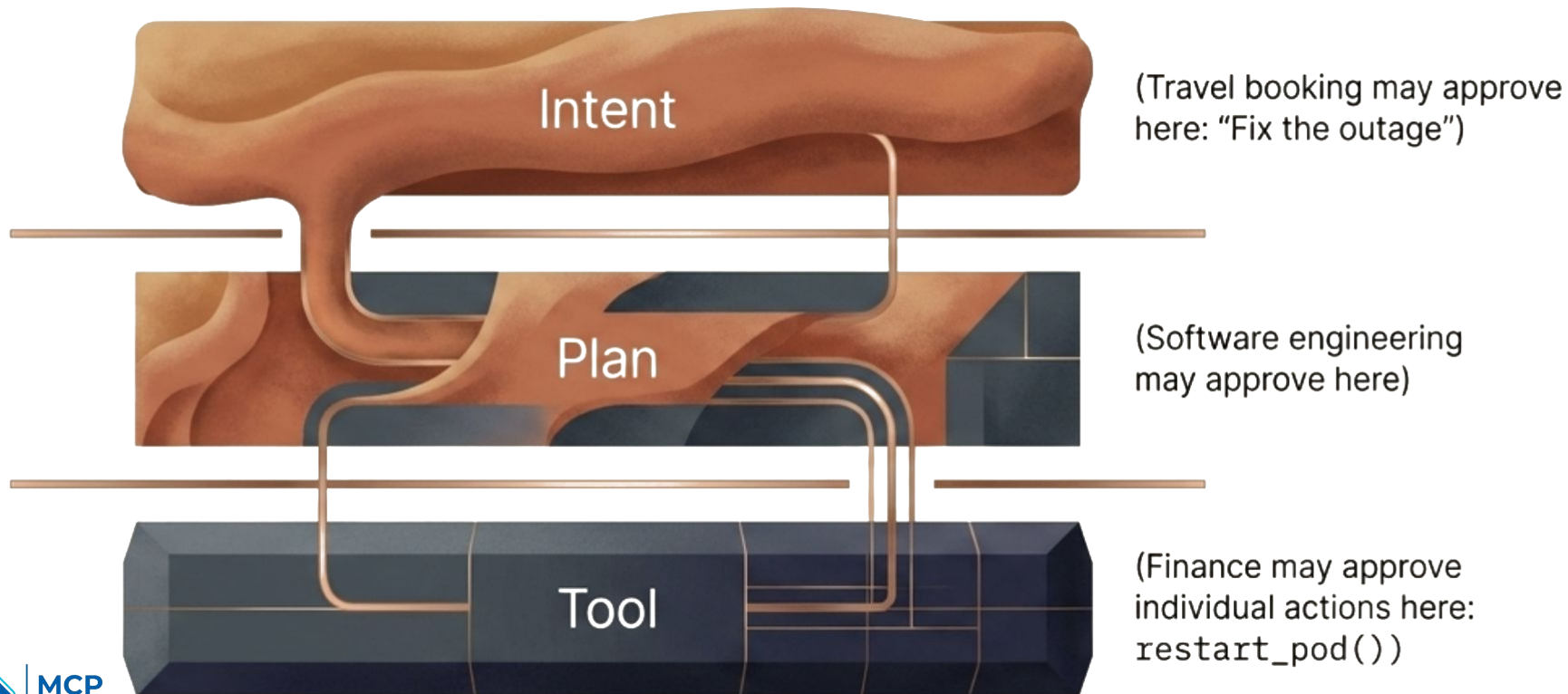
Organizational Knowledge



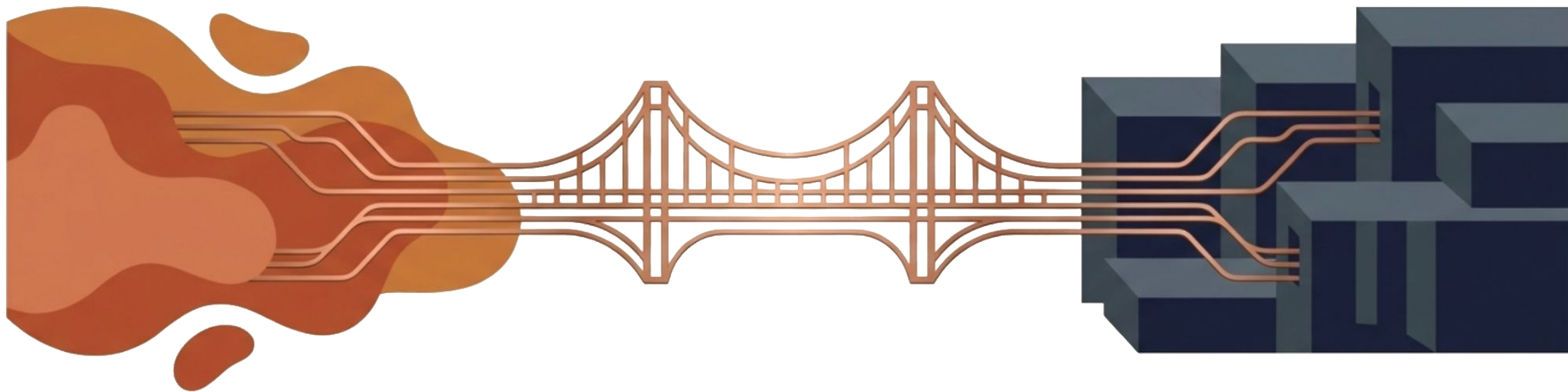
Routing and orchestration



Intent Approval

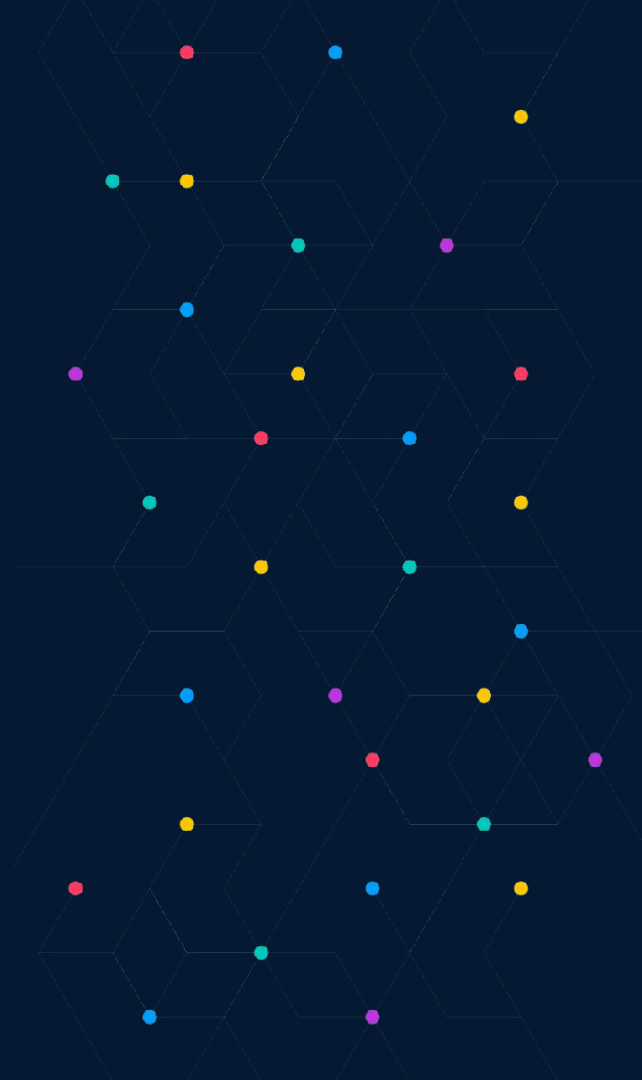


Closing



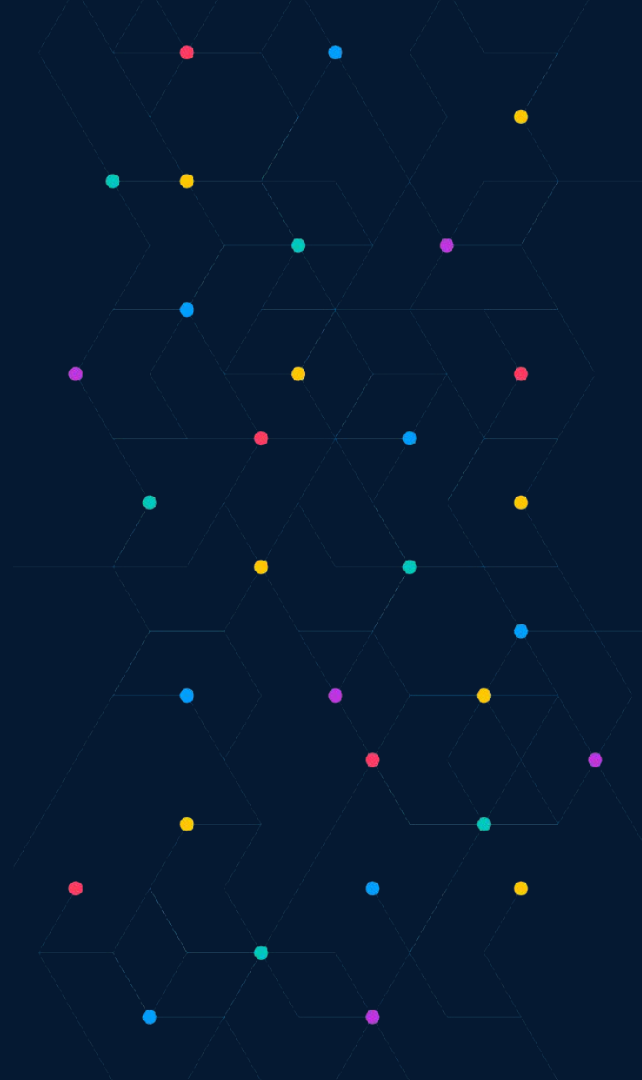


MCP
Dev Summit
Seoul





MCP
Dev Summit
Seoul





MCP
Dev Summit
Seoul



MCP
Dev Summit
Seoul

