



MCP
Dev Summit
North America

Progressive Tool Discovery

Using MCP Notifications to Manage Context at Scale

Lilia Abaibourova, Billy Hickman | Prime Video

Chilling suspense TV [See more >](#)



Covert ops and investigations TV



Historical TV [See more >](#)

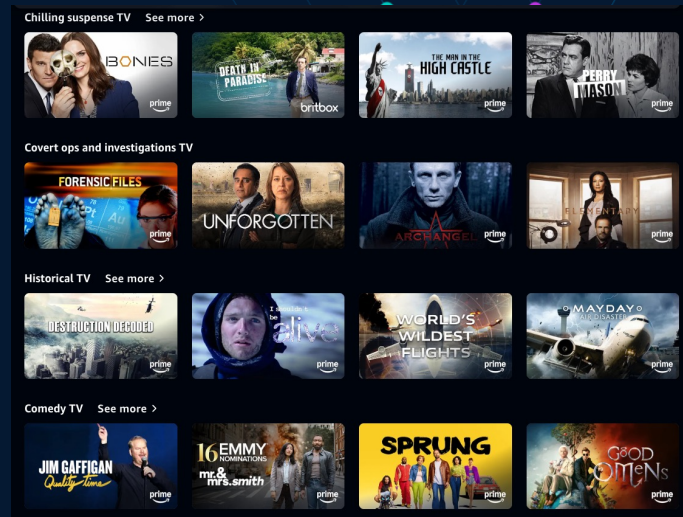


Comedy TV [See more >](#)



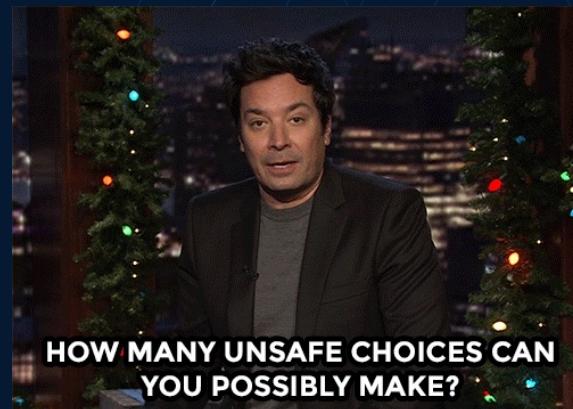
You only need one show ...

But you have 200 options!



You connect to an MCP

And get 200 tools!



Progressive Tool Discovery through MCP Notifications Feature



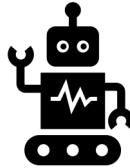
Billy Hickman, Senior Software Engineer
He | Him



Lilia Abaibourova, Principal Product Manager
She | Her

Agents: from Copilots to Collaborators

- Taking on operational and repetitive tasks
- Interacting with production systems and data
- Operating across Software Development Lifecycle

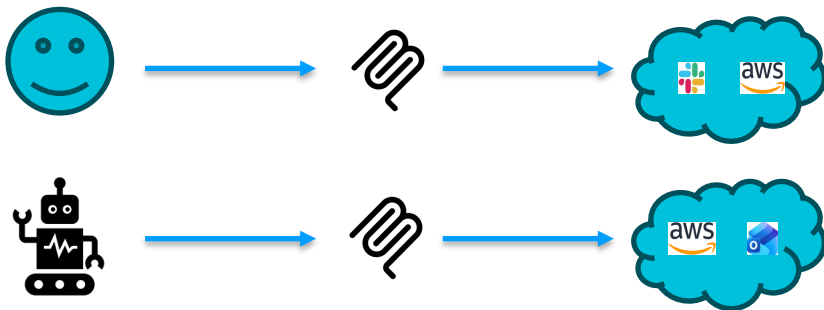


- But they still depend on the same context and tools engineers use today!



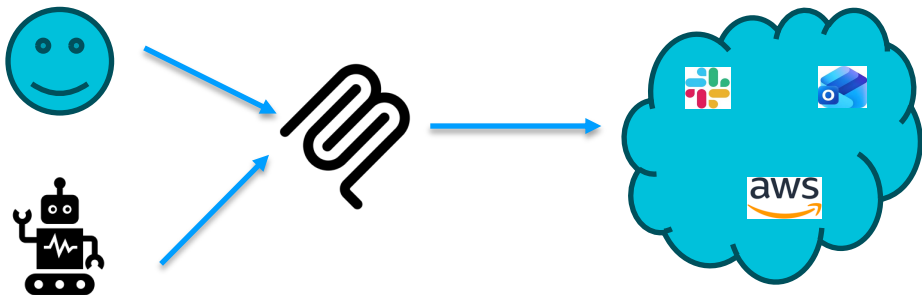
How agents are enabling Prime Video teams to deliver more

- Real-time playback quality analysis across every live event
- Triaging, root causing and resolving operational issues
- Analyzing experiment results that previously weren't funded



How agents are enabling Prime Video teams to deliver more

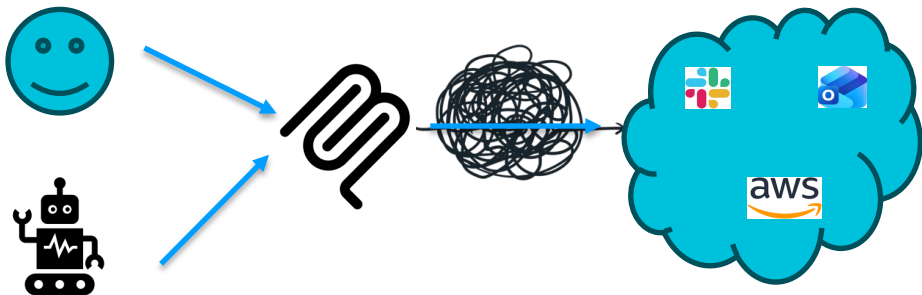
- Real-time playback quality analysis across every live event
- Triaging, root causing and resolving operational issues
- Analyzing experiment results that previously weren't funded



- As more teams relied on MCP tools, we invested in better infrastructure
- Shared MCP server for common tools to reduce duplication and maintenance

More Tools = Less Signal

- Context hits the limit due to tool bloat
- Tool selection becomes unreliable
- Performance degrades



- More tools \neq better agents

What did we need from our MCP use cases?

**Dynamic Tool Discovery
and Re-Discovery**

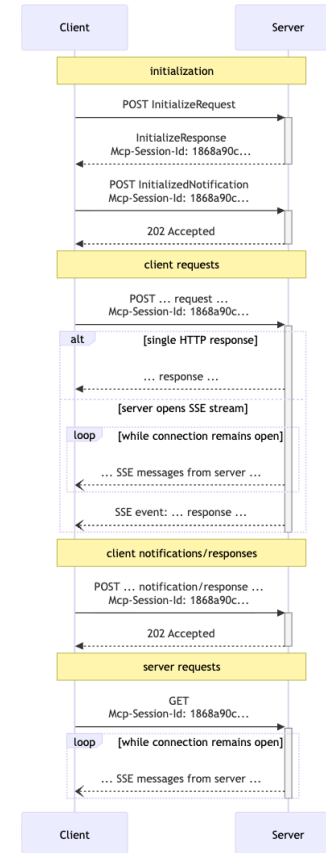
Security and Permissions

MCP Protocol Features



Thought process: How did we discover our approach

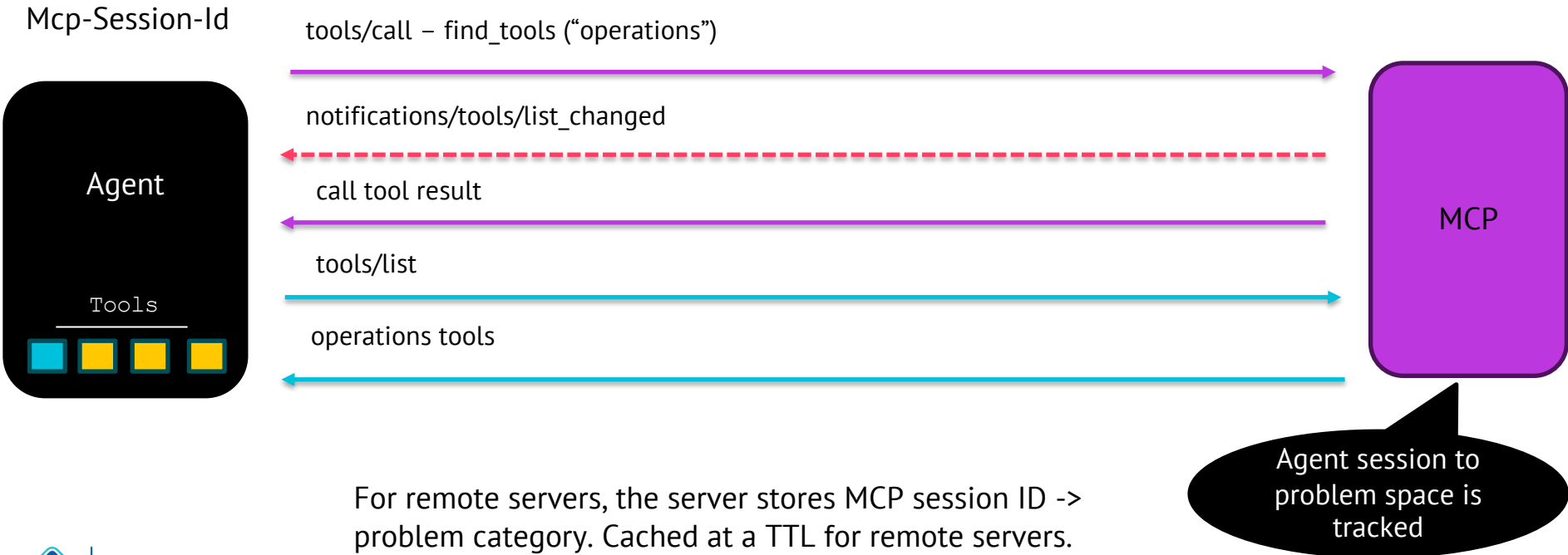
- **MCP notifications**, specifically the ability to notify an agent that their tool list needs updated.
 - “When the list of available tools changes, servers that declared the listChanged capability **SHOULD** send a notification”.
 - `{ "jsonrpc": "2.0", "method": "notifications/tools/list_changed" }`
- **MCP session tracking** (via `Mcp-Session-Id`) for remote MCP or a stateful stdio local MCP.
 - “If an MCP-Session-Id is returned by the server during initialization, clients using the Streamable HTTP transport **MUST** include it in the MCP-Session-Id header on all of their subsequent HTTP requests”



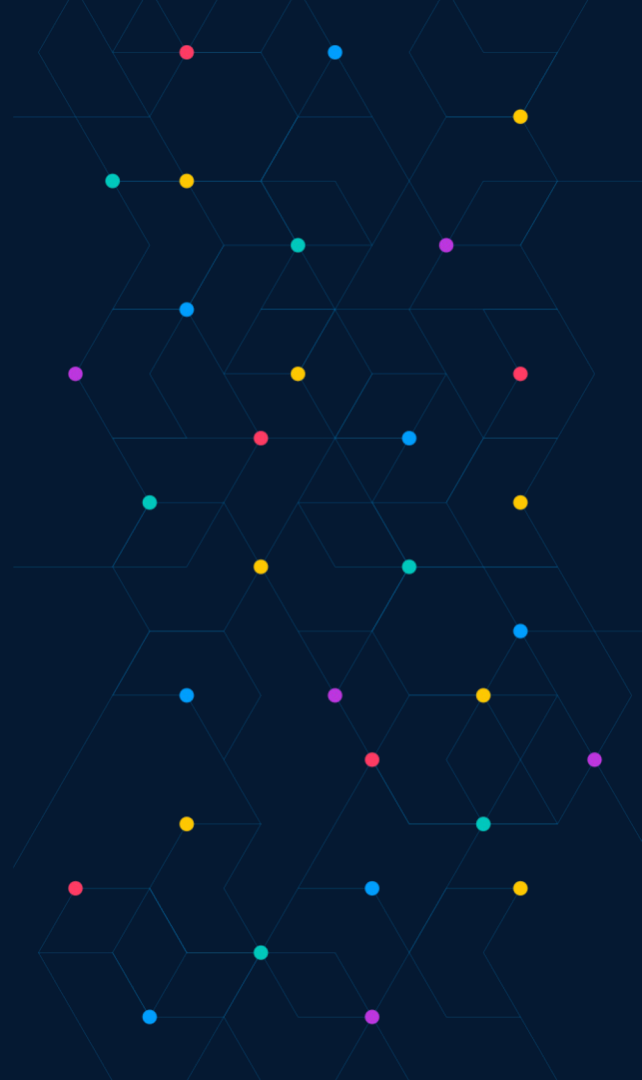
Mechanism: how did we use notifications for our flow

- Our idea was that after **initialization** we expose a single tool - `find_tools`.
- Deterministically map tools to problem category e.g. “operations”. A tool can be part of many problem spaces.
- Tool description for `find_tools` includes the description of each problem category and encourages invocation to discover additional tools.

Mechanism: how does this work in practice



Mechanism: Demo (🏃)



Mechanism: benefits of this approach

- We achieve **dynamic tool discovery**. Agent can refresh and re-discover tools multiple times.
- Tools within **the context window are dynamically added/removed**. Keeping context windows lean.
- We unlock the ability to **share cross cutting capabilities** across many agents without bloating context windows.
- Looking to use "natural language" discovery as a next step.
- Approach works for both remote MCP servers with HTTP streaming + SSE and local stdio MCP.

Thank You!

Any Questions?





MCP
Dev Summit
North America

