

# RPC > MCP

Turning a Decade of APIs Into Agentic Tools

# Ze'ev Klapow

Distinguished Engineer · HubSpot

Written with Claude.

# HubSpot.

CRM platform · ~2,300 engineers · thousands of microservices

The problem.

1,805 tools. 499 services.

# Agenda

1. How we got here
2. The architecture
3. The platform bet
4. From tools to agents



# How We Got Here

**We used to write REST.**

**We weren't really doing REST.**

# Typed RPCs.

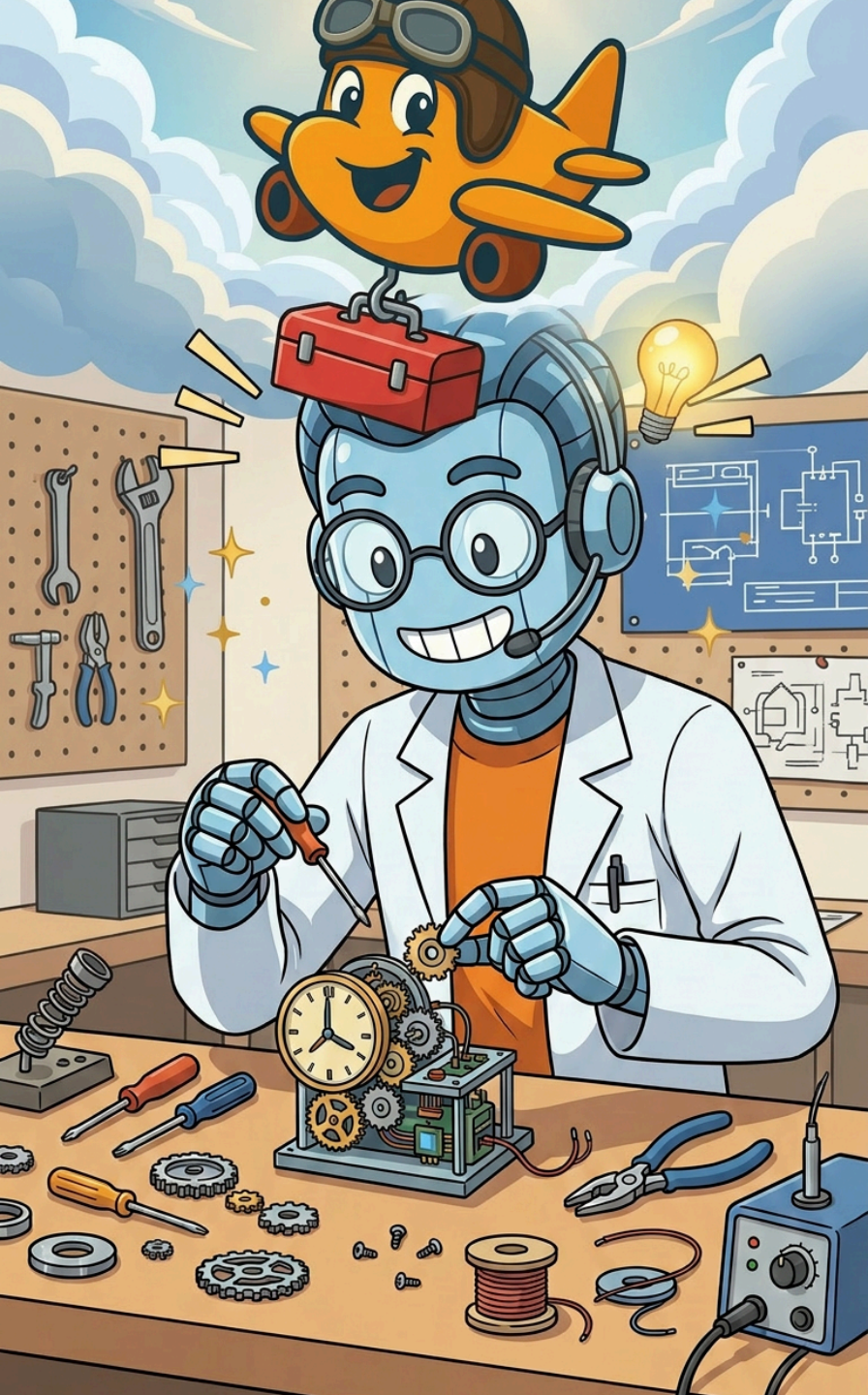
Contract-first. Generated clients. Schema-validated.

**An unintended consequence.**

**The stdio server.**

# Just a proxy.

```
@mcp.tool()
def get_build_logs(module: str, build_id: int) -> str:
    """Get build logs for a module"""
    return BuildService.getBuildLogs(module, build_id)
```



# The Architecture

**Two annotations.**

# Two annotations.

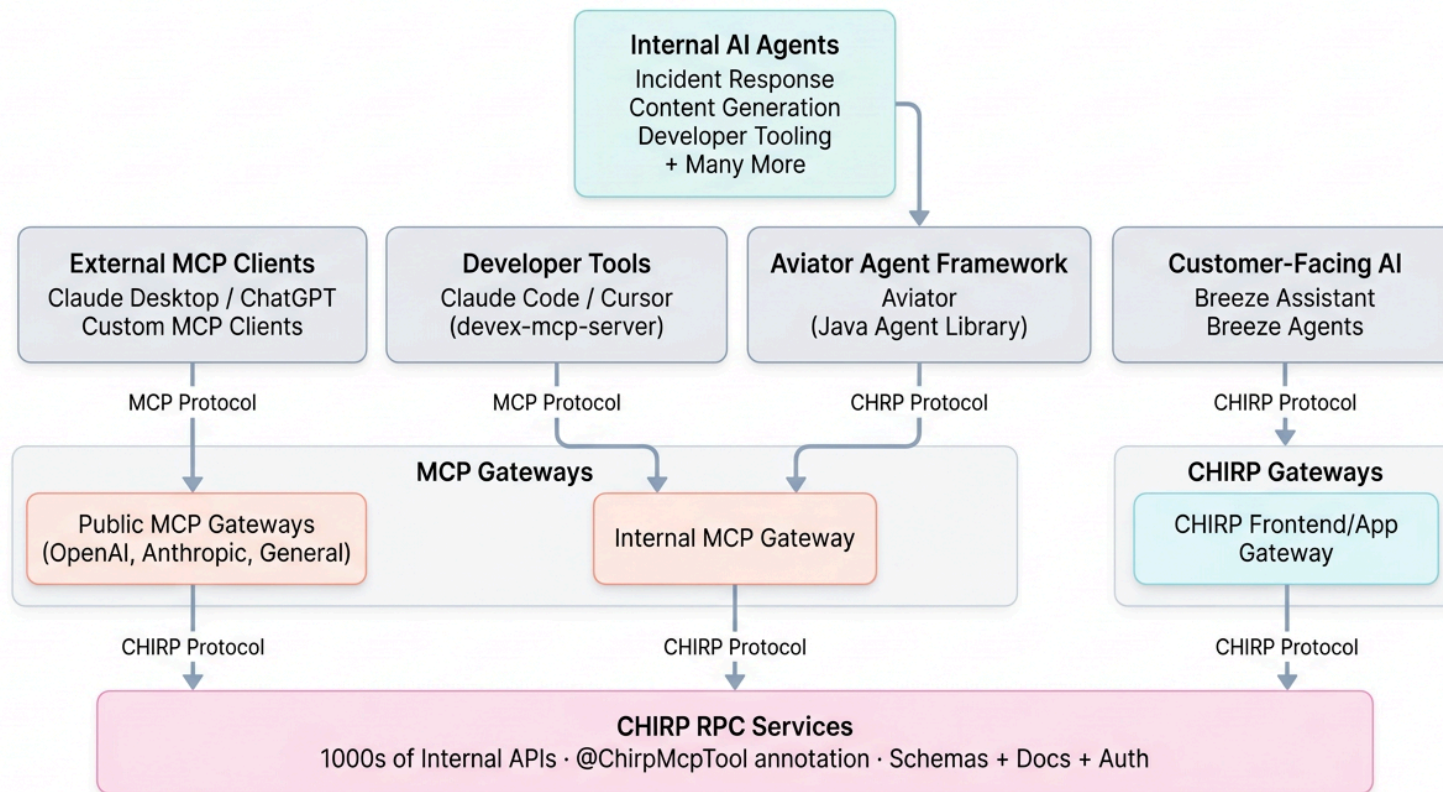
```
@ChirpToolsService(audience = INTERNAL_EMPLOYEE)
public interface DeploymentTools {

    @ChirpTool(description = "Get deploy history for a service")
    DeployHistoryResponse getDeployHistory(DeployHistoryRequest req);
}
```

**What already existed.**

**The gateway.**

# The gateway.



# Audiences.

internal employees · specific model providers · public developers · ...

One annotation. Every surface.

MCP at the edge, not everywhere.

# Schema remapping.

Anthropic and OpenAI want different things.

# Same tool. Different schemas.

```
// Anthropic
{
  "objectType": "string",
  "query": {
    "anyOf": [
      {"type": "null"},
      {"type": "string"}
    ]
  },
  "sorts": [{
    "propertyName": "string",
    "direction": "ASCENDING"
  }]
}
```

```
// OpenAI
{
  "object_type": "string",
  "query": "string",

  "sorts": [{
    "property_name": "string",
    "direction": "ASCENDING"
  }],
  "required": ["object_type",
    "query", "sorts"]
}
```

**Dynamic tool filtering.**



# The Platform Bet

# Inversion of control.

97% of tools built by product teams, not the platform team.

One pattern.

Convergence.

# Security.

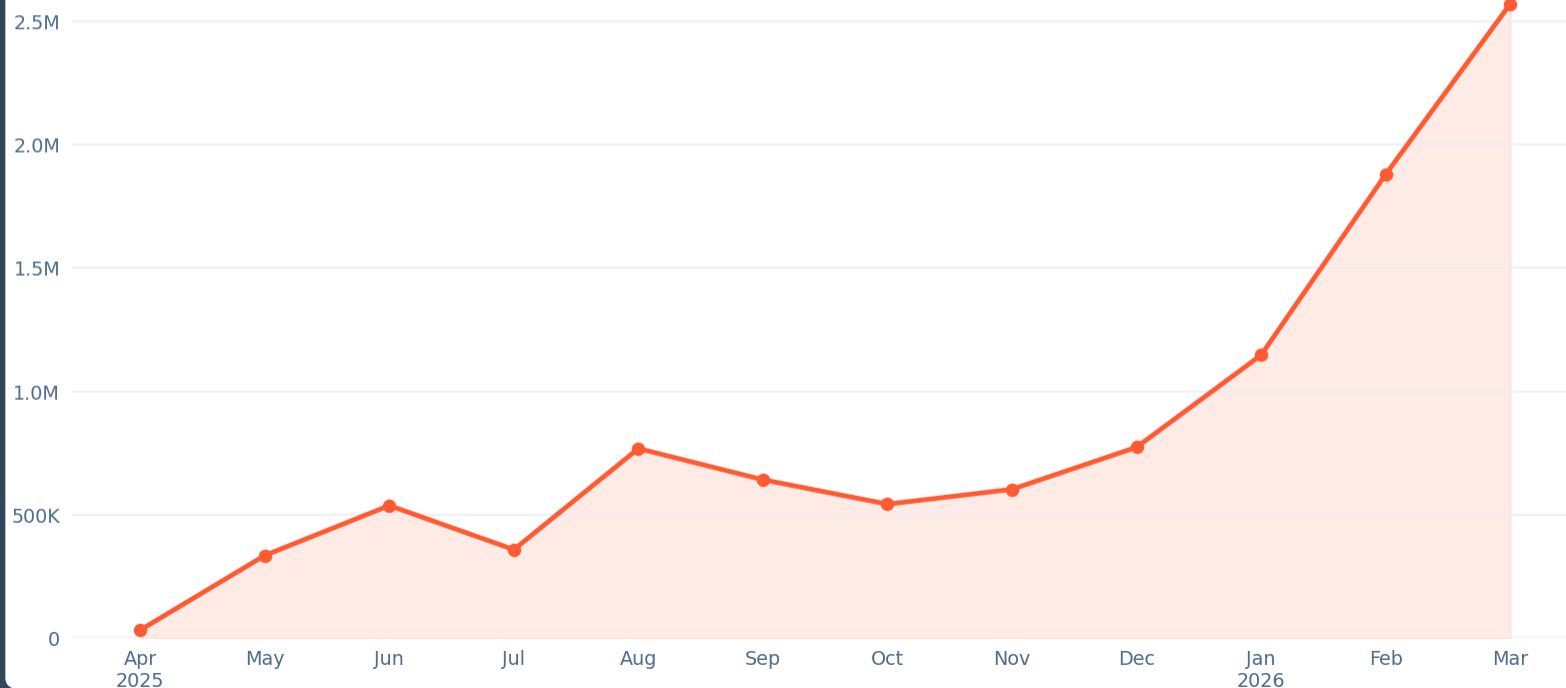
One gateway. Not a thousand servers.

# The flywheel.

## Monthly MCP Tool Calls

32K → 2.6M · 80x growth in 11 months

Monthly MCP Tool Calls





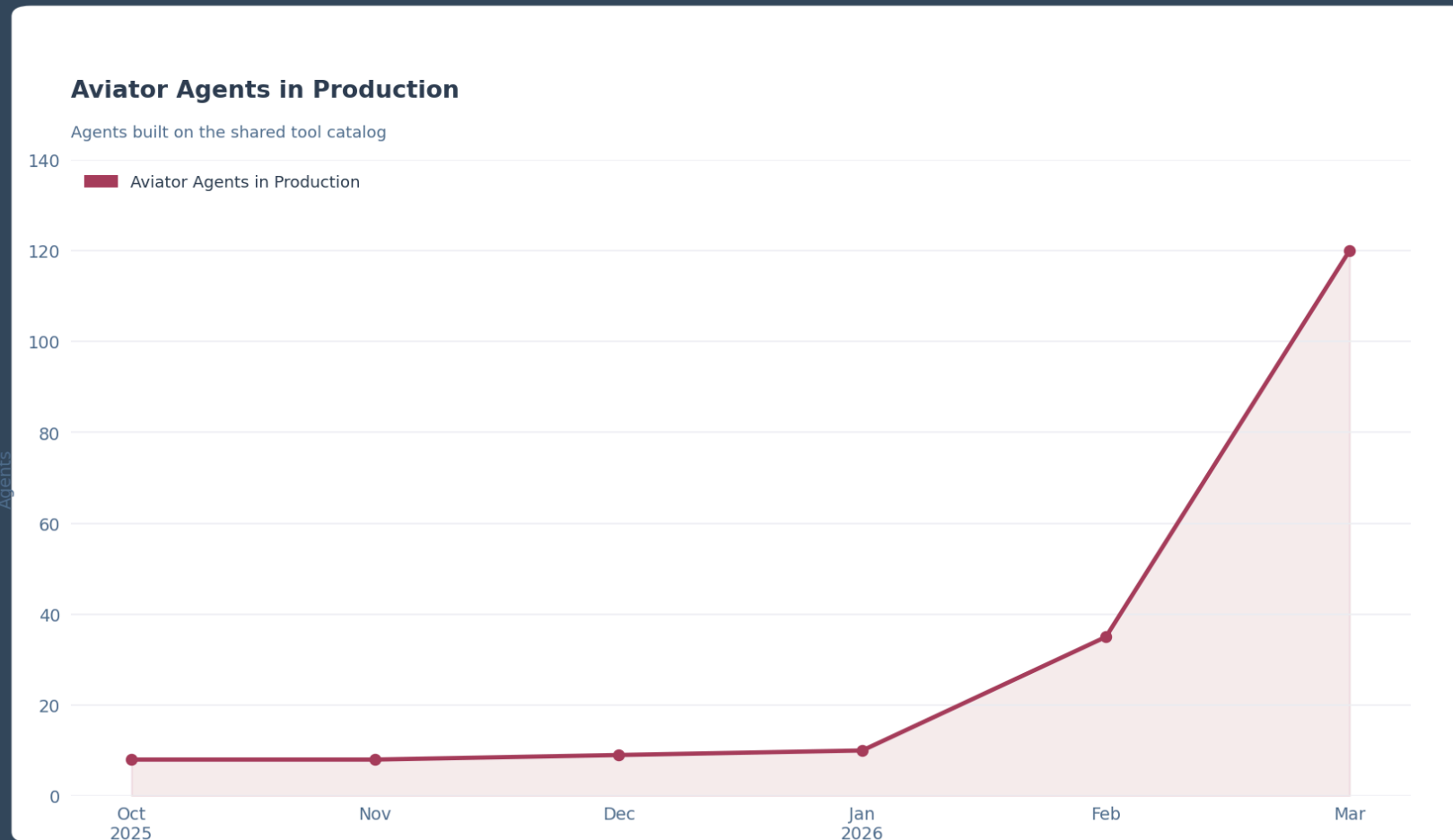
# From Tools to Agents

**The same bet for agents.**

# Aviator.

```
AviatorAgent agent = agentFactory.newBuilder()  
    .withSystemPrompt("You are a deployment assistant...")  
    .withModelClientProvider(() -> claudeSonnet)  
    .addChirpTool("DeploymentTools#getDeployHistory")  
    .addChirpTool("DeploymentTools#rollbackDeploy")  
    .build();
```

# The downstream effect.



**Use what you already have.**

**Distributed ownership. No  
fragmentation.**

# Questions?

Ze'ev Klapow · HubSpot