Putting AI to Work

What the heck is an MCP server?

About Me

- Charlotte, NC
- Developer Relations
- Open Source Nerd
- Not an AI Expert

Our Agenda

- Introductions
- Timeline of AI
- What is MCP?
- Experimenting
- More Resources
- Q & A

Introductions

Tell us About You

- Your name
- What you do
- What was the first album you ever purchased?
  (or listened to)

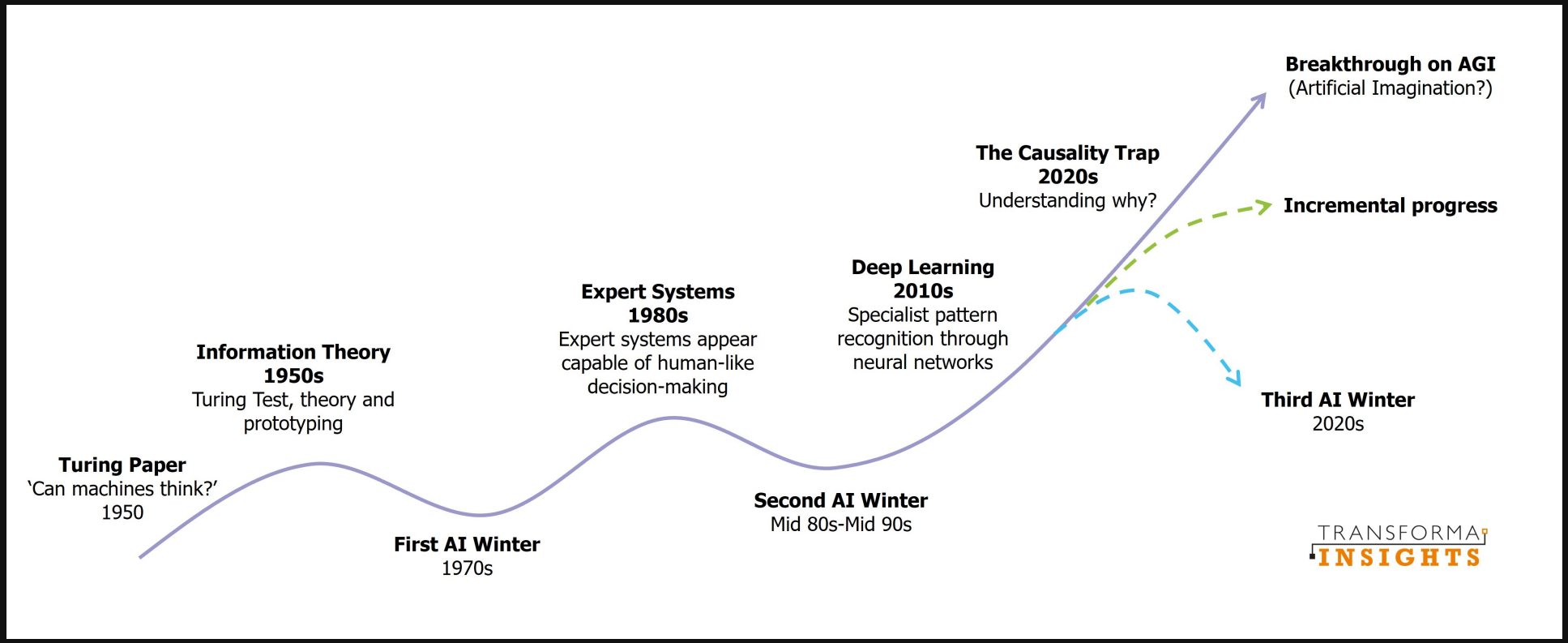
The Current AI Moment

Where did we come from?

1950s - 1970s (AI that Reasons)
the birth of AI, symbolic AI, reasoning

1980s - 1990s (AI Winter)
decision engines, expert systems, niche uses

2000s - 2010s (AI that Predicts)
ML/DL, faster hardware, speech and image recognition


Breakthrough on AGI
(Artificial Imagination?)
The Causality Trap
2020s
Understanding why?
Incremental progress
Deep Learning
2010s
Specialist pattern recognition through neural networks
Expert Systems
1980s
Expert systems appear capable of human-like decision-making
Information Theory
1950s
Turing Test, theory and prototyping
Turing Paper
'Can machines think?'
1950
First AI Winter
1970s
Second AI Winter
Mid 80s-Mid 90s
Third AI Winter
2020s
TRANSFORMA
INSIGHTS

Where are we right now?

2020s (AI that Creates)

LLMs/Generative AI - text, images, code, etc.

2023 - Present (AI that Acts)

Agentic AI - operationalizing LLMs

What is MCP?

How did it come to be?


Gemini

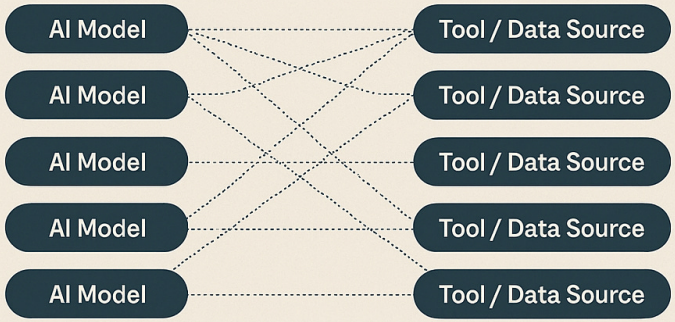
10^8–10^9

APIs and Programmatic Interfaces

The Fundamental Problem MCP Solves

N×M Problem

For every AI model that needed to connect with every external tool or data source, a unique integration had to be built and maintained.



1 Development inefficiency

Teams spent countless hours building and maintaining separate integrations

2 Innonsistent implementations

Different developers implemented similar functions in wildly different ways

3 Innovation barriers

Organizations struggled to scale their AI applications

4 Siloed capabilities

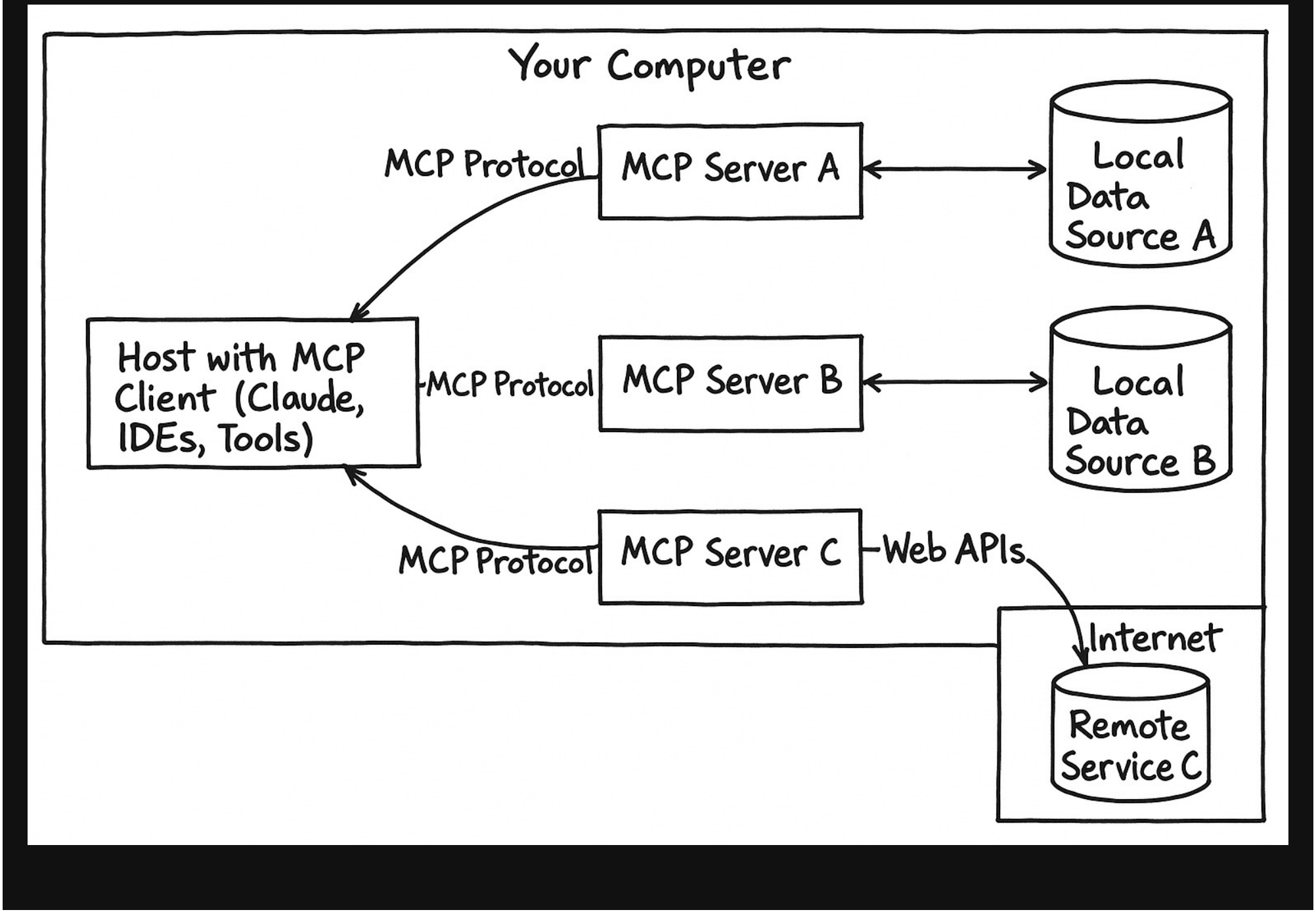
Advanced integration work remained trapped within specific systems

The NxM Problem

- Many competing AI tools and many, many, many external tools
- Developed by Anthropic, announced Nov 2024
- Improved upon ChatGPT's plug-in framework / function-calling API (2023)
- Aimed at addressing the NxM data integration problem

Model-Context-Protocol

- A standard way for LLM tools (Claude, Copilot, ChatGPT) to talk to the outside world.
- Minimizes the need for custom integrations.
- Think of it kind of like ODBC or a USB-C port for AI applications.


Your Computer
MCP Protocol
MCP Server A
Local Data Source A
Host with MCP Client (Claude, IDEs, Tools)
MCP Protocol
MCP Server B
Local Data Source B
MCP Protocol
MCP Server C
Web APIs
Internet
Remote Service C

Key Pieces of MCP

Hosts + Clients

The AI-powered applications that communicate with MCP Servers using dedicated MCP Clients.

Servers

Expose data and actions to AI Host's clients (often via plugins or adapters)

Protocol

It's JSON-RPC under the hood.

Layers

Transport Layer

Outer layer; manages communication and authentication

Data Layer

Inner layer; defines primitives (tools, resources, prompts, etc.)

Server Primitives

Tools ("What I can do")

Executable functions (API Calls, file operations, DB procedures)

Resources ("What I know")

Data sources (file contents, DB records, API responses)

Prompts (structured interactions)

Reusable templates for structured interactions (plan a vacation, draft an email, etc.)

Client Primitives

Sampling

Lets servers request LLM completions from AI application (ex: Pick the best flight of these)

Elicitation

Lets servers request additional info from users (ex: Ask for the user's seat preference)

Logging

Lets servers send log messages to clients for debugging/monitoring.

Notifications

- Real-time updates
- Tool changes or new functionality
- Clients don't need to poll for changes
- Lets AI apps adapt to changing contexts

Why is MCP a big deal?

- Devs: Reduced development time
- Users: "Smarter" AI for end users
- Apps: Plug-and-play functionality
- Everyone: Cross-platform AI synthesis

Let's Tinker a Bit

More Resources

- https://modelcontextprotocol.io
- https://mcp.so
- https://github.com/umbraco/Umbraco-CMS-MCP-Dev
- https://www.youtube.com/@IBMTechnology
- Model Context Protocol with Tim Berglund

Final Thoughts

Final Thoughts

Reason ➡️Predict ➡️Create ➡️Act ➡️???

Final Thoughts

Reason ➡️Predict ➡️Create ➡️Act ➡️???

- 🤝Collaborate
- 🌱Adapt
- 💡Invent

Q & A