

THE SERIES

Today and coming up

Mon 14 Sep · Manhari Room

TODAY

Equipping AI agents with tools

Daniel Russo-Batterham, MDAP

Mon 12 Oct · Room M13

Working with synthetic data

Irma Mooi-Reci, School of Social and Political Sciences and QuantLab

Mon 9 Nov · Room M01

Making AI more trustworthy with knowledge maps

Taylor Ey, Melbourne Institute

Mon 14 Dec · Manhari Room

Analysing historical data with AI

David Goodman, School of Historical and Philosophical Studies

All sessions 12–1pm at Melbourne Connect, with the optional drop-in until 1.30pm.

Equipping LLMs with tools

Dr Daniel Russo-Batterham
Melbourne Data Analytics Platform, The University of Melbourne

daniel.russo@unimelb.edu.au

Session structure

The Data Landscape and AI (5 min)

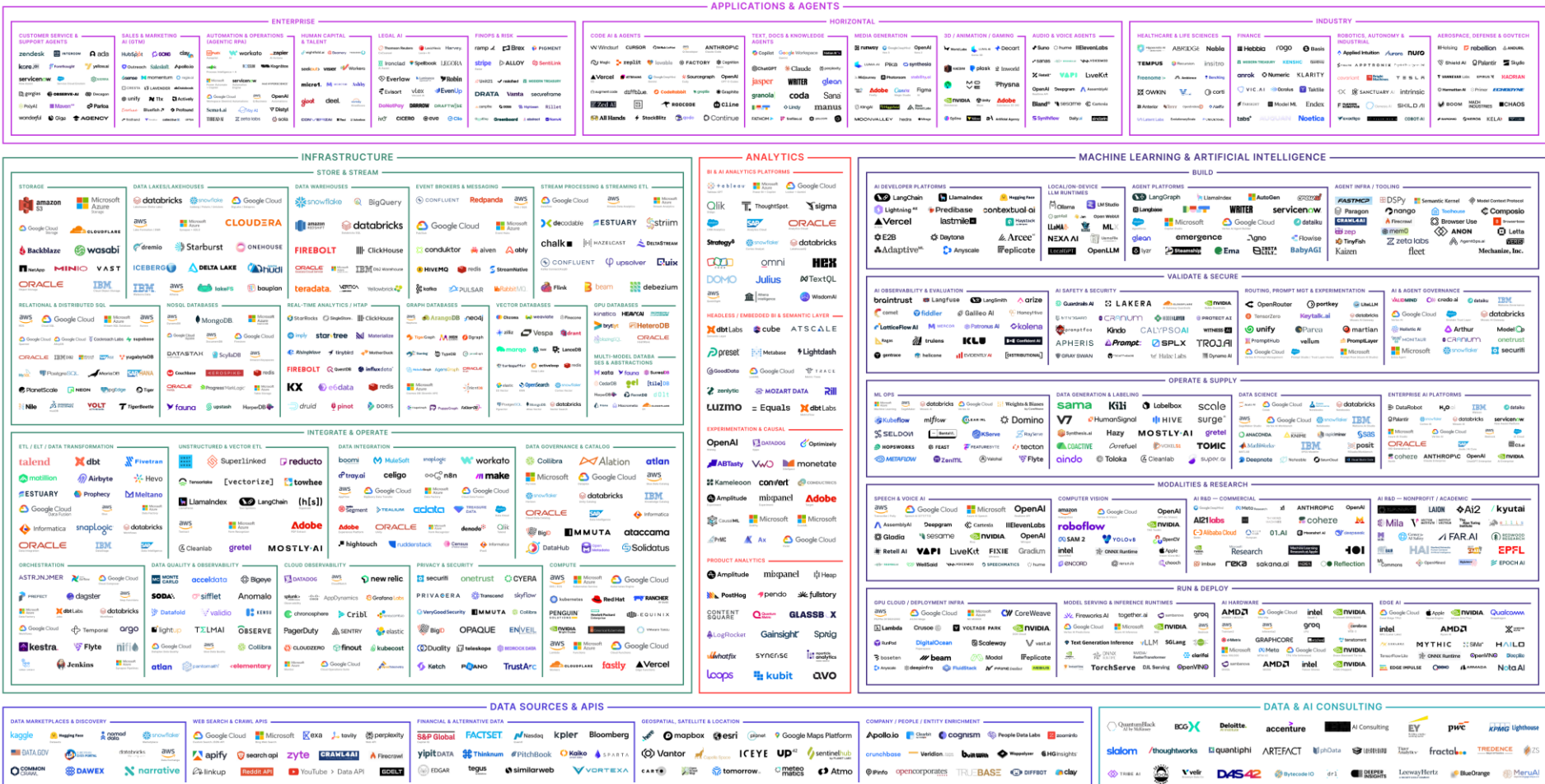
Why equip LLMs with tools and how? (10 min)

MCP examples and case studies (10 min)

Activity (15 min)

Discussion and questions (10 min)

The Data Landscape and AI



Prompt engineering
Fine-tuning
Embeddings
Hallucination
Context engineering
Chain-of-thought
Agentic AI
Function calling
Loop engineering
A2A
Observability
Sandboxing
Permissions
Vector database
Ollama
vLLM
Orchestration
Human-in-the-loop
Spec-driven development
Quantization
Routing
Tracing
Structured outputs
Claude
Local models
Inference
CrewAI
PydanticAI
Graph engineering
Context window
Guardrails
Tokens
LangChain
CLIs
Multimodal
Hooks
Evals
LlamaIndex
Sub-agents
Coding agents
Tool use
Plugins
Agents
Skills
MCPs
RAG
Pinecone
ChromaDB
LangGraph
Hugging Face
OpenCode
Codex
Herdr

Skills, Connectors (MCPs) and Plugins



Customize

Skills

Connectors

Plugins

Yours

Discover



Add

From Anthropic

Productivity

Tasks, planning and follow-ups — the everyday toolkit.

Add



New

See all >



Maven Bio

Evidence-backed BioPharma research workflows over Maven Bio's drug, trial,...
from Maven Bio · added 23 hours ago



NetSuite AI Companion

Guides AI assistants that use the NetSuite AI Connector: tool-selection order, output...
from Oracle NetSuite · added yesterday



NetSuite Finance Analyst

Director-level finance analysis on live NetSuite data: financial reporting, period...
from Oracle NetSuite · added yesterday



Unstructured Foundation

Connect sources and search processed documents with Unstructured Foundatio...
added yesterday



learn

Use this skill when the user wants intellectual understanding — learning ho...
from Anthropic · added May 29



doc-coauthoring

Guide users through a structured workflow for co-authoring documentation. Use wh...
from Anthropic · added Dec 5, 2025



Why equip LLMs with tools?

- Runs battle-tested code so tends to be better at specialised tasks than ad hoc LLM generations
- More reliable, verifiable, and reproducible
- Democratises access to non-technical audiences through Natural Language Interfaces

AI Skills

1. **brainstorming** - Activates before writing code. Refines rough ideas through questions, explores alternatives, presents design in sections for validation. Saves design document.
2. **using-git-worktrees** - Activates after design approval. Creates isolated workspace on new branch, runs project setup, verifies clean test baseline.
3. **writing-plans** - Activates with approved design. Breaks work into bite-sized tasks (2-5 minutes each). Every task has exact file paths, complete code, verification steps.
4. **subagent-driven-development** or **executing-plans** - Activates with plan. Dispatches fresh subagent per task with two-stage review (spec compliance, then code quality), or executes in batches with human checkpoints.
5. **test-driven-development** - Activates during implementation. Enforces RED-GREEN-REFACTOR: write failing test, watch it fail, write minimal code, watch it pass, commit. Deletes code written before tests.
6. **requesting-code-review** - Activates between tasks. Reviews against plan, reports issues by severity. Critical issues block progress.
7. **finishing-a-development-branch** - Activates when tasks complete. Verifies tests, presents options (merge/PR/keep/discard), cleans up worktree.

<https://github.com/obra/superpowers>



12 people

Release v6.3.0: Devin CLI and Hermes Agent suppo...



b36e082 · last month



History

250 lines (198 loc) · 15.1 KB

Preview

Code

Blame



Raw



name	brainstorming
description	You MUST use this before any creative work - creating features, building components, adding functionality, or modifying behavior. Explores user intent, requirements and design before implementation.

Brainstorming Ideas Into Designs

Help turn ideas into fully formed designs and specs through natural collaborative dialogue.

Start by classifying how much process the request needs, then work through your path: understand the context, refine the idea, present a design, and get your human partner's approval.

Do NOT invoke any implementation skill, write any code, scaffold any project, or take any implementation action until you have told your human partner what you intend and they have approved it. This applies to EVERY task on EVERY path below — the ceremony scales with the task; the approval gate never does.

Three Paths

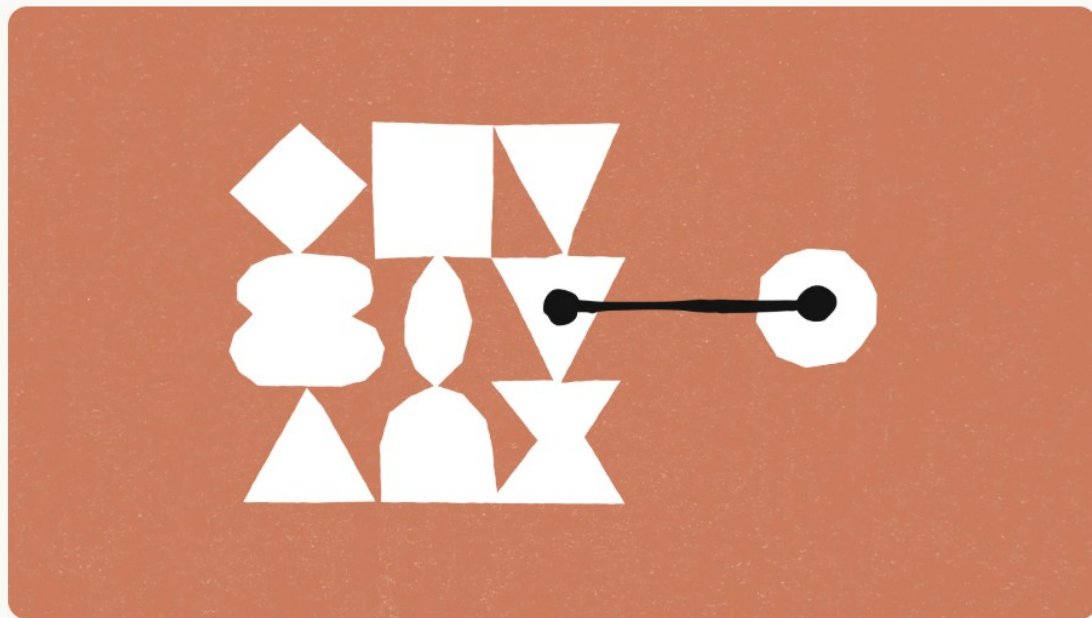
Before your first question, classify the request and say the classification out loud — "this looks bounded, so I'll present a short design here rather than write a spec" — so your human partner can override it:

- **Spike** — a feasibility question ("can we...", "is it possible...", "quick and dirty is fine") whose output is an answer, not code you keep. Present the question and what you'll try in 2-3 sentences, get a nod, then find out as cheaply as correctness allows. No design doc, no spec file. Report findings as a recommendation; anything you built stays labeled throwaway.
- **Bounded** — a well-scoped change to code that already exists in this repo: a new flag, a small endpoint, a one-file fix. Understanding the kind of app is not enough — bounded

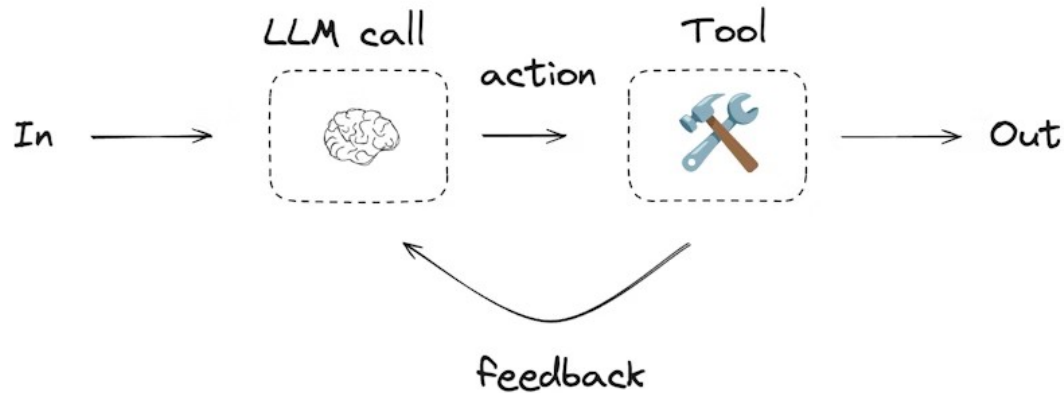
Announcements

Introducing the Model Context Protocol

26 Nov 2024 • 3 min read



<https://www.anthropic.com/news/model-context-protocol>



<https://docs.langchain.com/oss/python/langgraph/workflows-agents>

```
# Define tools
@tool
def multiply(a: int, b: int) -> int:
    """Multiply 'a' and 'b'.
```

```
Args:
    a: First int
    b: Second int
    """
    return a * b
```

```
@tool
def add(a: int, b: int) -> int:
    """Adds 'a' and 'b'.
```

```
Args:
    a: First int
    b: Second int
    """
    return a + b
```

```
@tool
def divide(a: int, b: int) -> float:
    """Divide 'a' and 'b'.
```

```
Args:
    a: First int
    b: Second int
    """
    return a / b
```

- Tools
- Resources
- Prompts

Servers provide the fundamental building blocks for adding context to language models via MCP. These primitives enable rich interactions between clients, servers, and language models:

- **Prompts:** Pre-defined templates or instructions that guide language model interactions
- **Resources:** Structured data or content that provides additional context to the model
- **Tools:** Executable functions that allow models to perform actions or retrieve information

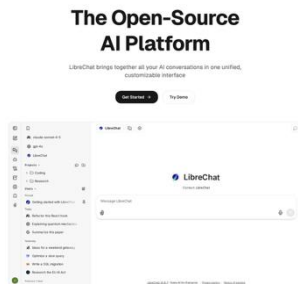
Each primitive can be summarized in the following control hierarchy:

Primitive	Control	Description	Example
Prompts	User-controlled	Interactive templates invoked by user choice	Slash commands, menu options
Resources	Application-controlled	Contextual data attached and managed by the client	File contents, git history
Tools	Model-controlled	Functions exposed to the LLM to take actions	API POST requests, file writing

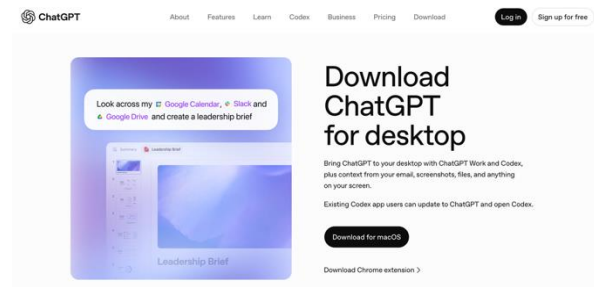
<https://modelcontextprotocol.io/specification/2025-06-18/server>

**Where can you use
these tools?**

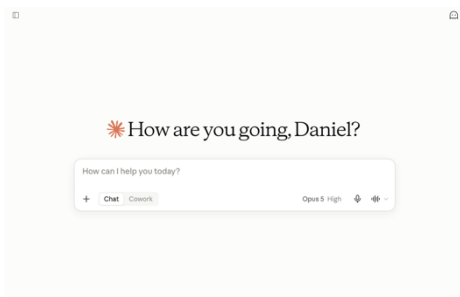
Example client applications



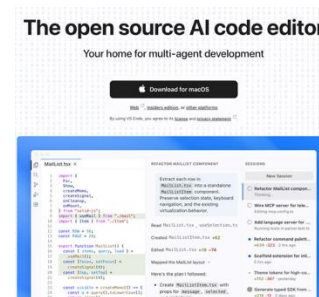
LibreChat · librechat.ai



ChatGPT · chatgpt.com



Claude · claude.ai, Claude Code, Desktop



VS Code · code.visualstudio.com

...and many other chat applications

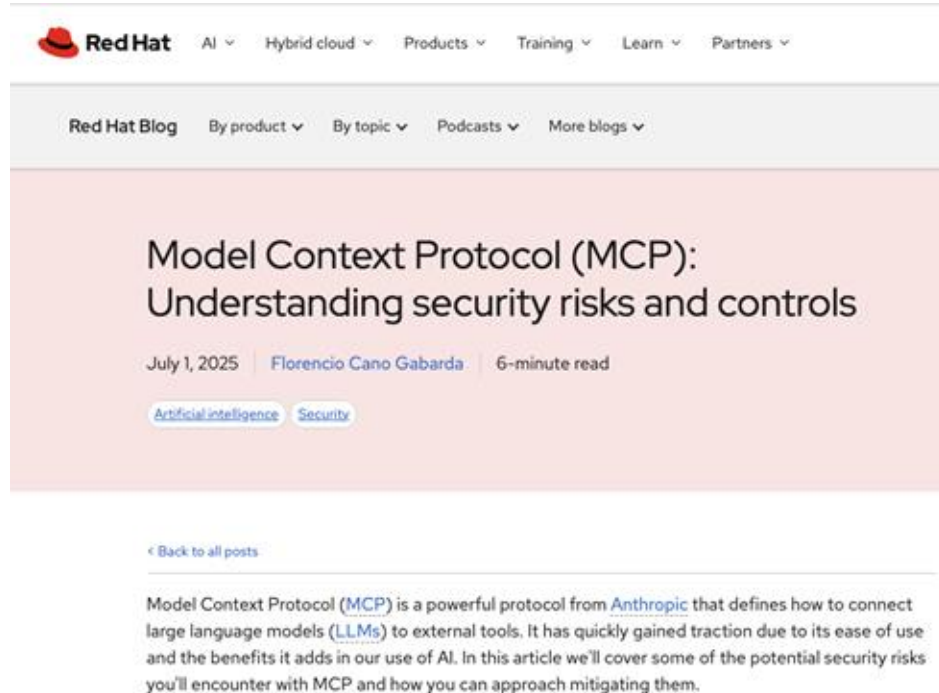
STDIO (local)

- Client application runs the server
- Typically uses U VX, NPM or Docker
- Often allows direct operation on local files (!!)

```
{
  "mcpServers": {
    "airbnb": {
      "command": "npx",
      "args": [
        "-y",
        "@openbnb/mcp-server-airbnb"
      ]
    }
  }
}
```

Streamable HTTP (remote)

Security



The image is a screenshot of a Red Hat blog post. At the top, the Red Hat logo is on the left, and navigation links for AI, Hybrid cloud, Products, Training, Learn, and Partners are on the right. Below this is a secondary navigation bar with 'Red Hat Blog' and filters for 'By product', 'By topic', 'Podcasts', and 'More blogs'. The main content area has a light pink background. The article title 'Model Context Protocol (MCP): Understanding security risks and controls' is centered. Below the title, the date 'July 1, 2025', the author 'Florencio Cano Gabarda', and the duration '6-minute read' are listed. Two tags, 'Artificial intelligence' and 'Security', are shown in rounded boxes. A link '< Back to all posts' is at the bottom left of the article section. The article text begins with 'Model Context Protocol (MCP) is a powerful protocol from Anthropic that defines how to connect large language models (LLMs) to external tools. It has quickly gained traction due to its ease of use and the benefits it adds in our use of AI. In this article we'll cover some of the potential security risks you'll encounter with MCP and how you can approach mitigating them.'

Red Hat AI Hybrid cloud Products Training Learn Partners

Red Hat Blog By product By topic Podcasts More blogs

Model Context Protocol (MCP): Understanding security risks and controls

July 1, 2025 Florencio Cano Gabarda 6-minute read

Artificial intelligence Security

< Back to all posts

Model Context Protocol (MCP) is a powerful protocol from Anthropic that defines how to connect large language models (LLMs) to external tools. It has quickly gained traction due to its ease of use and the benefits it adds in our use of AI. In this article we'll cover some of the potential security risks you'll encounter with MCP and how you can approach mitigating them.

<https://www.redhat.com/en/blog/model-context-protocol-mcp-understanding-security-risks-and-controls>

Some use cases

Getting data from national repositories through APIs

Recent events

```
#title: Recent events
SELECT ?event ?eventLabel ?date
WITH {
  SELECT DISTINCT ?event ?date
  WHERE {
    # find events
    ?event wdt:P31/wdt:P279* wd:Q1190554.
    # with a point in time or start date
    OPTIONAL { ?event wdt:P585 ?date. }
    OPTIONAL { ?event wdt:P580 ?date. }
    # but at least one of those
    FILTER(BOUND(?date) && DATATYPE(?date) = xsd:dateTime).
    # not in the future, and not more than 31 days ago
    BIND(NOW() - ?date AS ?distance).
    FILTER(0 <= ?distance && ?distance < 31).
  }
  LIMIT 150
} AS %i
WHERE {
  INCLUDE %i
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],mul,en" . }
}
```

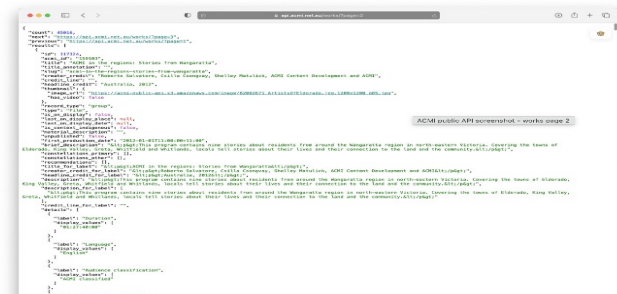
Wikidata query service: recent events as structured data



To get started, let's have a look at all of the Works in our collection:

<https://api.acmi.net.au/works/>

You'll notice that what you see is a page of text, but don't worry, a JSON viewer browser plugin will help you make sense of it all. Here's what it looks like when it's formatted nicely using JSON Peep:



ACMI public collection API

Domain-specific tools for working and exploring data

Can you show me the notation for Bach BWV 772?

Loaded tools, ran a command

Found tools

Bash

E Encoding music MCP show_notation

</>

Invention No. 1 in C major, BWV 772



← Prev

Page 1 of 7

Next →

Count and rank the most frequent melodic 4-grams in Bach_BWV_0772.mel

Used encoding-music-mcp integration, loaded tools >

E encoding-music-mcp plot_melodic_ngram_heatmap

</>

Melodic N-gram Heatmap

1 score

- Bach, Johann Sebastian | Invention No. 1 in C major, BWV 772

n=4 | kind=d | top 5 patterns across all selected scores.



| Available Tools



list_available_mei_files

DISCOVERY

Lists all built-in MEI files available for analysis (Bach, Bartók, Morley, CRIM Mass)

Returns: `List[str]`



get_mei_metadata

METADATA

Extracts metadata from MEI files: title, composer, editors, publication date, availability

filename: `str`



analyze_key

KEY ANALYSIS

Detects key signature and tonal analysis using music21 library

filename: `str`



get_notes

INTERVALS

Extracts all notes from an MEI file with pitch, octave, measure, and beat information

filename: `str`



get_melodic_intervals

INTERVALS

Calculates melodic intervals between consecutive notes. Supports diatonic, chromatic, and quality modes

filename: `str`
kind: `str` ("d"|"c"|"q"|"z")



get_harmonic_intervals

INTERVALS

Calculates harmonic intervals between simultaneous notes in chords

filename: `str`
kind: `str` ("d"|"c"|"q"|"z")



get_melodic_ngrams

INTERVALS

Finds recurring melodic patterns (n-grams) of specified length across the piece

filename: `str`
n: `int` (2-5)



get_cadences

INTERVALS

Detects harmonic cadences (authentic, plagal, half, deceptive) in the score

filename: `str`



show_notation

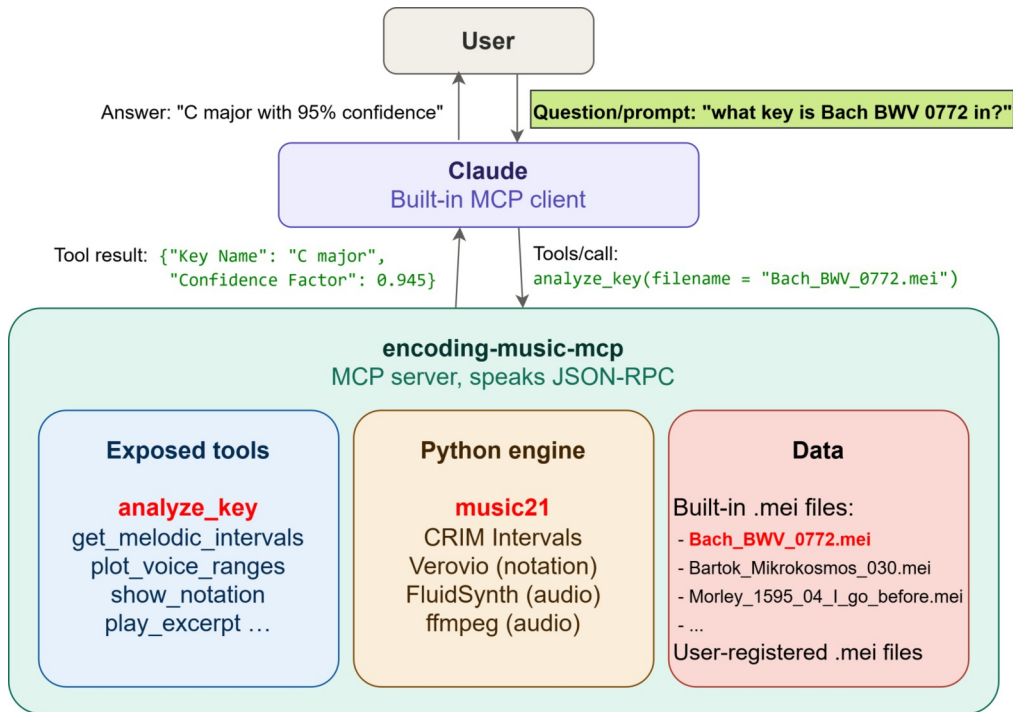
NOTATION

Renders MEI notation as SVG using Verovio and displays in interactive viewer

filename: `str`

How does it work?

User asks questions in natural language



Bach BWV 772 key identification ▾

what key is Bach BWV 0772 in?

Loaded tools, used encoding-music-mcp integration ▾

I know Bach's Invention No. 1 in C major is BWV 772, but let me verify this using the music analysis tool I have access to since there's an MEI file available for the piece.

Loading tools

Result

Analyze key

Result

Done

Bach's Invention No. 1, BWV 772, is in **C major** — analysis of the score gives this with very high confidence (~0.95).

📄 👍 💬 ↺



Write a message...

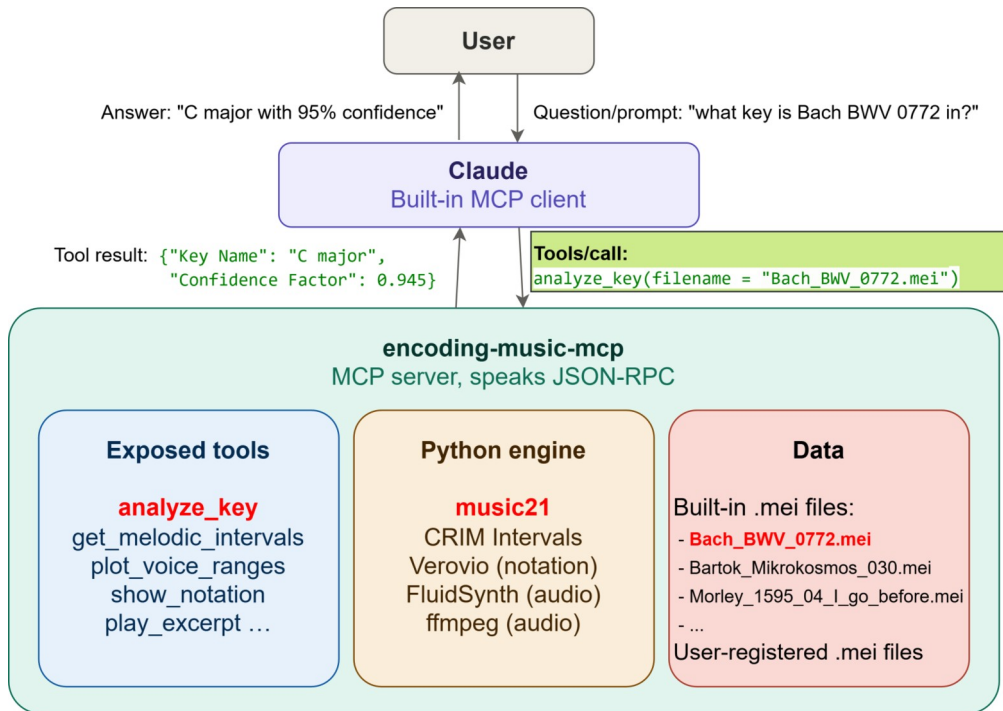
+

Opus 4.7 Adaptive ▾



Claude is AI and can make mistakes. Please double-check responses.

MCP client decides which tool(s) to call



Bach BWV 772 key identification ▾



what key is Bach BWV 0772 in?

Loaded tools, used encoding-music-mcp integration ▾

I know Bach's Invention No. 1 in C major is BWV 772, but let me verify this using the music analysis tool I have access to since there's an MEI file available for the piece.

Loading tools

Result

E Analyze key

Request

```
{
  "filename": "Bach_BWV_0772.mei"
}
```

Response

```
{"Key Name": "C major", "Confidence Factor": 0.9450617236680577}
```

Write a message...

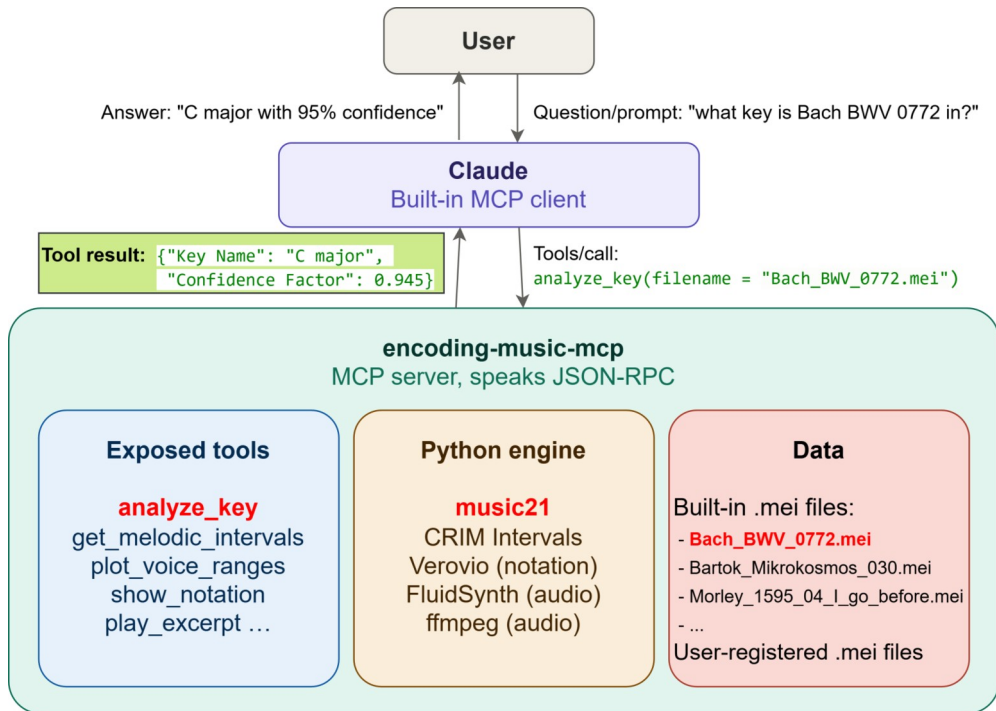


Opus 4.7 Adaptive ▾



Claude is AI and can make mistakes. Please double-check responses.

MCP server returns the tool result



Bach BWV 772 key identification ▾

what key is Bach BWV 0772 in?

Loaded tools, used encoding-music-mcp integration ▾

I know Bach's Invention No. 1 in C major is BWV 772, but let me verify this using the music analysis tool I have access to since there's an MEI file available for the piece.

Loading tools

Result

E Analyze key

Request

```
{
  "filename": "Bach_BWV_0772.mei"
}
```

Response

```
{"Key Name": "C major", "Confidence Factor": 0.9450617236680577}
```

Write a message...

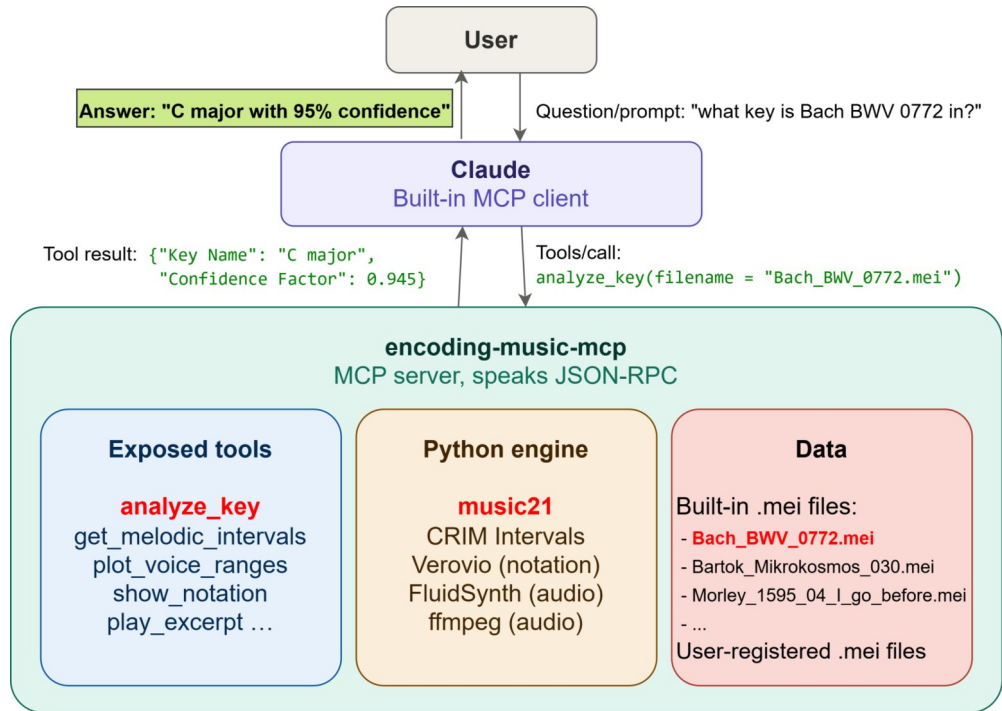
+

Opus 4.7 Adaptive ▾

🎤

Claude is AI and can make mistakes. Please double-check responses.

MCP client interprets the result



Bach's BWV 772 — the first of his Two-Part Inventions — is in C major. The music21 key analysis on the score confirms it with a confidence of about 0.95.

Hansard case study

Semantic search of the database in Claude Desktop via MCP

Prompt: Find me the top 5 quotes from the 1850s in Victoria about democracy as a problem or danger. This might relate to perceptions of the fickleness of 'the people', their irrationality, lack of education, or the way the propertyless could not be trusted to govern responsibly.

Here are the five strongest passages from the 1850s Victorian debates on this theme — all found via semantic search, so quoted with the corpus's own OCR artefacts intact rather than silently corrected:

1. On the fickleness and manipulability of "the people" (Vic, 1857)

"There is in every country a class of persons who are pleased to designate themselves 'The People,' and to deny that appellation to all who do not adopt their own peculiar views... The mob orator of public meetings, like the drum in a regimental band, though very empty, makes a very great noise. Instead of argument, he uses abuse; and instead of appeals to the judgment and understanding of his audience, he appeals to their worst passions; and when every thing that can prevent their forming a correct opinion has been insidiously laid before them, they are made, like puppets, to pass resolutions condemnatory of every body and every thing."

2. On the ignorant voter as a manipulable child (Vic, 1857 — Mr. Moore, in franchise debate)

"an ignorant man would have to vote under the direction of the returning officer, or some other individual at the polling booth, and might thus ↓ e out the name of the

How democracy was discussed in parliament in 1850s Victoria

I asked Sonnet: 'Reverting to the Victorian discussion of democracy through the 1850s - what adjectives, metaphors or similes are used to describe "the people" in these debates? Group by pro and anti democracy speakers.' The answer was very revealing. It pointed to comments about the **animality** of the mob; ordinary people as **puppets** able to be manipulated by the rich and powerful; the **childishness** of the ignorant and uneducated; **democracy itself** viewed as an organic and threatening force. The people were cast as "ignorant," "besotted," "degraded," "rash," "speculative," "unsettled" (of goldfields population), "incendiary" (of adventurer-politicians), "presumptuous," "insidious."

On the other (pro-democracy) side, it found, the language was much less vivid and metaphoric. Ordinary and humble people had **common sense**, **respect for property and the law**, **conservative instincts**, they were **industrious and intelligent**.

Then I gave the same prompt to Opus, the higher model. It sliced the material in a different way – by speaker, usefully, helping me see the ways certain speakers ran one or other of these arguments.

I'll do this more systematically this time — keyword searches on the period's actual vocabulary for "the people," rather than relying on semantic paraphrase alone.

Used hansard-search integration, loaded tools, read memory · 1 note

This time I ran keyword searches on the period's own nouns for the people (mob, rabble, populace, multitude, masses, demagogue) rather than relying on semantic paraphrase, and chased speaker attributions through chunk_context. That changes the picture in one important way, which I'll come to at the end.

Anti-democracy / democracy-wary speakers

John Pascoe Fawkner — Legislative Council, Reform Bill, 1858.

The single richest cluster in the corpus, all in one speech:

- Quotes verse at the House: "May the rude rabble's insolence despise, / Their senseless clamor and tumultuous cries." (rude, insolent, senseless, tumultuous)
- "The rowdy mob are very few in number, but they put forth a penny whistle and sound a penny trumpet, and think that it is the note of a trumpet of war and can be heard for miles throughout the colony, to the ends of the earth" — miniaturised-noise simile ↓

World Bank Data360

Analysis and visualization

TOOL	DESCRIPTION
<code>data360_summarize_data</code>	Summary statistics grouped by dimensions—use for trend or change-over-time questions without pulling full series.
<code>data360_rank_countries</code>	Rank countries by indicator value for a year (top/bottom performers, leaderboards).
<code>data360_compare_countries</code>	Compare 2–8 countries on one indicator (snapshot or aligned time series).
<code>data360_get_viz_spec</code>	Vega-Lite spec and chart URL for one indicator. Call <code>get_disaggregation</code> first. Fetches its own data; does not consume <code>get_data</code> output.
<code>data360_get_multi_indicator_viz_spec</code>	Chart comparing 2–4 indicators (scatter, dual-axis line, etc.).
<code>data360_get_supported_chart_types</code>	List supported chart types and data requirements.

Compare population growth across Australia, New Zealand and Canada using the compare countries tool

Data360 compare countries ▾

Data360 compare countries ▾

Request

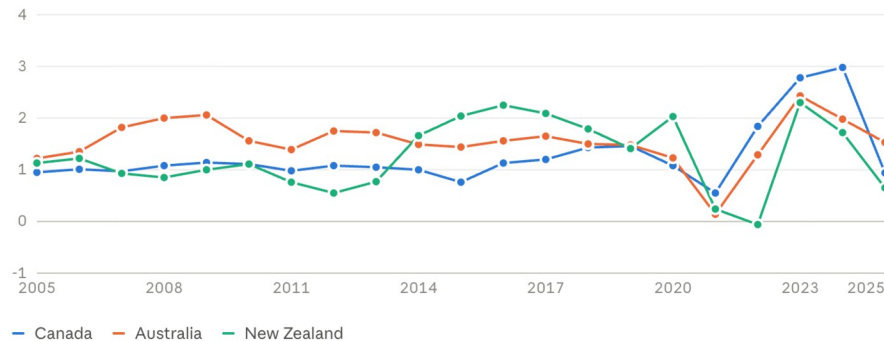
```
{
  "country_codes": "AUS;NZL;CAN",
  "database_id": "WB_WDI",
  "end_year": 2025,
  "include_time_series": true,
  "indicator_id": "WB_WDI_SP_POP_GROW",
  "start_year": 2005
}
```

~

Comparing trends in data across multiple countries.

Population growth, annual %, 2005–2025

Annual % change



PubMed

Biomedical literature search, in two steps: a search returns article IDs, a second call fetches the detail.

Anthropic itself provides a set of example use cases as part of its training: <https://academy.claude.com/tutorials/using-the-pubmed-connector-in-ai>

Can you find recent articles that explore FAIR principles in medical data contexts?

Used PubMed integration ▾

Search articles FAIR principles health data ▸

Checking search results against the intended date filter.

Search articles (FAIR data principles[Title/Abstract]) AND (health OR clinical OR biomedical OR medic... ▾

Request

```
{
  "max_results": 20,
  "query": "(FAIR data principles[Title/Abstract]) AND (health OR clinical OR biomedical OR medical) AND (\"2024\"[Date - Publication] : \"2026\"[Date - Publication])",
  "sort": "pub_date"
}
```

Response

```
{
```

Get article metadata ▾

Request

```
{
  "pmids": [
    "42704371",
    "42598558",
    "42555930",
    "42552320",
    "42626911",
    "42288436",
    "42523250",
    "42494821",
    ...
  ]
}
```

Filtering sources to find studies on FAIR in medical data.

Based on articles retrieved from PubMed — 70 results for FAIR data principles in health contexts published 2024 onward. The most relevant, grouped by what they actually contribute:

Concrete infrastructure implementations

The bioimaging workflow paper is probably the most directly useful if you're interested in how FAIR gets operationalised rather than advocated. It chains OMERO, eLabFTW, Adamant, and Micro-Meta App through an open-source Jupyter notebook, and — unusually — reports a structured FAIR assessment of the result, landing at 50–60% compliance. That willingness to publish a number rather than claim success is rare. [10.1038/s41597-026-07982-x](https://doi.org/10.1038/s41597-026-07982-x)

Mapbox

Search and geocoding (6)

Tool	Purpose
<code>search_and_geocode_tool</code>	POIs, brands, chains, cities, addresses
<code>category_search_tool</code>	Places by category, industry, amenity, or NAICS-style code
<code>category_list_tool</code>	Deprecated — use <code>resource_reader_tool</code> with <code>mapbox://categories</code>
<code>reverse_geocode_tool</code>	Coordinates to address or place name
<code>place_details_tool</code>	Full detail for a place by Mapbox ID
<code>ground_location_tool</code>	Combined: reverse geocode + nearby POIs + isochrone in one call

Routing and travel time (5)

Tool	Purpose
<code>directions_tool</code>	Turn-by-turn directions between coordinates
<code>isochrone_tool</code>	Reachable areas within time or distance thresholds
<code>matrix_tool</code>	Travel times and distances between many points
<code>optimization_tool</code>	Shortest route through 2-12 stops
<code>map_matching_tool</code>	Snap GPS traces to the road network

Geometry and spatial analysis (17)

Measurement: `area_tool`, `length_tool`, `distance_tool`, `bearing_tool`, `bbox_tool`.

Construction: `buffer_tool`, `centroid_tool`, `midpoint_tool`, `destination_tool` (point from origin + bearing + distance), `convex_tool` (convex hull).

Set operations: `union_tool`, `intersect_tool`, `difference_tool`.

Point operations: `nearest_point_tool`, `nearest_point_on_line_tool`, `points_within_polygon_tool`.

Simplification: `simplify_tool`.

Rendering (2)

`render_map_tool` for the interactive Mapbox GL JS map, `static_map_image_tool` for a static image.

Show me locations within 10, 20 and 30 minutes of Melbourne Connect by foot

2 minutes ago ↺ ↻ ↗ < 2/2 >

Isochrone Tool ▾

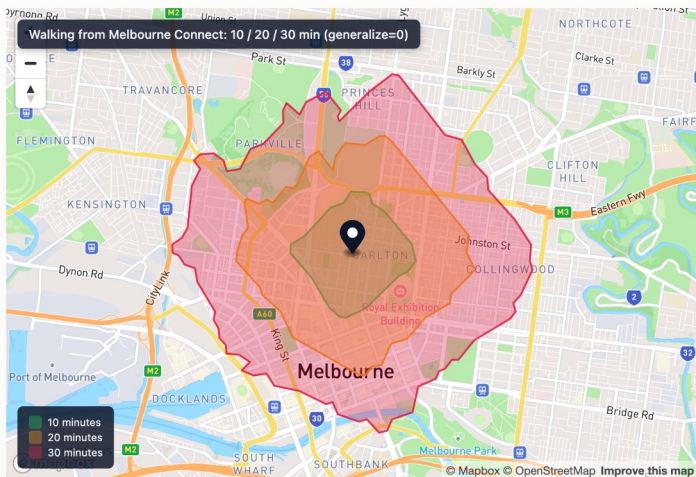
Adjusting settings for lower generalization and richer color variety.

Isochrone Tool ▾

Request

```
{
  "contours_colors": [
    "22c55e",
    "f59e0b",
    "e11d48"
  ],
  "contours_minutes": [
    10,
    20,
    30
  ]
}
```

Mapbox render_map_tool



Draw.io and Wikidata

Can you use the DrawIO MCP to create a periodic table? Be sure to validate all of the information before creating the diagram. You could use the Wikidata MCP for this.

Building a periodic table diagram checked against Wikidata.

Good idea — let me pull the element data from Wikidata first rather than writing it from memory.

Used Wikidata integration, loaded tools ▾

Searched available tools Wikidata SPARQL query properties search ▾

Search properties atomic number element symbol ▾

Search properties period group of periodic table ▾

Searching for periodic table position data on Wikidata.

Search items chemical element class of substance ▾

Execute sparql ▾

```
{
  "K": 130,
  "sparql": "SELECT ?number ?symbol ?elementLabel WHERE {
    ?element wdt:P31 wd:Q11344 ;
    wdt:P1086 ?number ;
    wdt:P246 ?symbol .
  } SERVICE wikibase:label { bd:serviceParam
    wikibase:language \"en\". }
  }\\n\\nORDER BY ?number
}"
```

Response

```
;symbol;number;elementLabel
0;H;1;hydrogen
1;He;2;helium
2;...;...;...
```

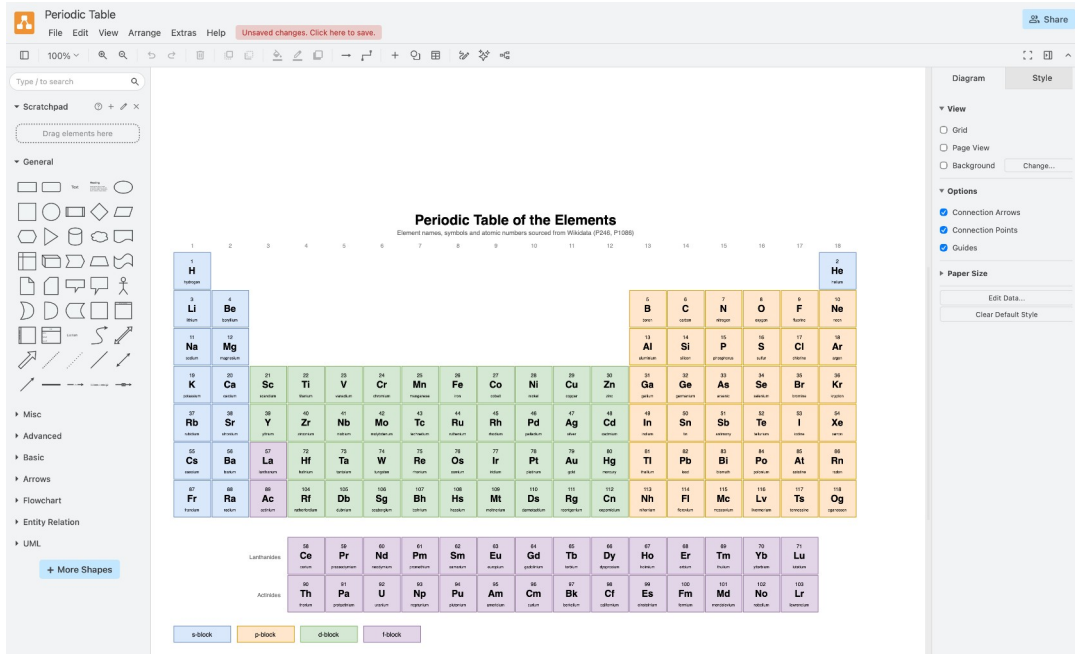
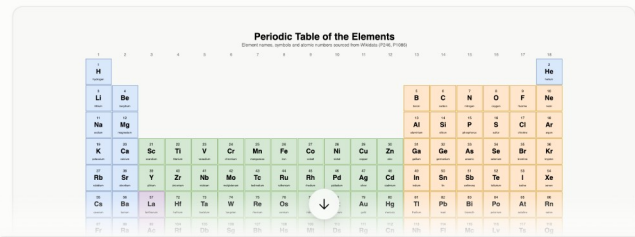
Clean pull — 1 through 118 all present with symbols and names. Now the diagram tool.

Searched available tools DrawIO create diagram ▾

Drafting the full markup layout for each periodic cell. ▾

Wikidata returned all 118 cleanly, plus systematic placeholder names for 119-130 (ununennium onwards) which I've excluded since those are unsynthesised.

 DrawIO



Activity

Things to try

- Connect to an MCP server (if you haven't already) and confirm access in a chat: "Do you have access to the [server name] MCP server?"
- Ask what tools the server provides and how they work
- Ask questions that use the tools and evaluate the quality of what comes back
- *For those who don't want to sign in to claude.ai, there is a pre-baked conversation with tools you can follow.*



Workshop resources:
go.unimelb.edu.au/vh4h

Things to critique

- Is the selection of tools provided by a given MCP server useful?
- Is it clear which tools were used and why? What arguments were passed in and what response did the LLM get back?
- Can you easily distinguish between a deterministic tool response and an LLM summary response?
- Did the LLM give an accurate account of what the tool gave back?
- What are some things that could go wrong with these kinds of Natural Language Interfaces to tools?
- How would the LLM perform an equivalent task without access to specialised tools?

Things to watch out for

- Giving an LLM access to a tool does not make the entire interaction trustworthy. Use discretion.
- There is a big difference between an LLM getting data from a tool and knowing what its significance is. Interpretation is often best left to humans.
- Biases that flow through to tool use: selection or omission biases.
- Security.
- Always maintain a critical eye and be careful about common reasoning errors, both machine and human.

Questions?

daniel.russo@unimelb.edu.au

Handy MCP links

- List of reference and third-party MCP servers, as well as MCP frameworks: <https://github.com/modelcontextprotocol/servers>
- Official Python MCP SDK: <https://github.com/modelcontextprotocol/python-sdk>
- FastMCP Framework (includes additional features): <https://github.com/jlowin/fastmcp>
- TypeScript MCP SDK: <https://github.com/modelcontextprotocol/typescript-sdk>
- “The Shift from Models to Compound AI Systems”: <https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/>
- LangGraph for compound AI systems:
 - <https://docs.langchain.com/oss/python/langgraph/thinking-in-langgraph>
 - <https://docs.langchain.com/oss/python/langgraph/workflows-agents>
- RedHat post on security challenges: <https://www.redhat.com/en/blog/model-context-protocol-mcp-understanding-security-risks-and-controls>