

YOUR CODEBASE

Understand Anything

We don't just map your code. We teach it.

YOUR GUIDED TOUR



1 · SCAN

2 · MAP

3 · TEACH

1

2

3



CODE STRUCTURE



BUSINESS LOGIC



KNOWLEDGE BASE

Guided Tours

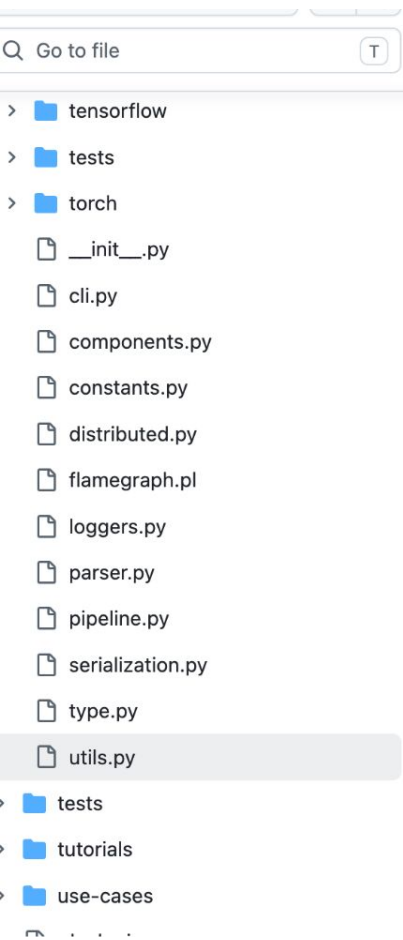
Semantic Search

Diff Impact

Layer View

Works with Claude Code · Codex · Cursor · Copilot · Gemini CLI · OpenCode · and more

20k lines, 100 files, 0 idea where to start



```
69 def load_yaml(path: str) -> Dict:
70     """
71     path (str): path to yaml file.
72     """
73
74     Raises:
75         yaml.YAMLError: for loading/parsing errors.
76
77     Returns:
78         Dict: nested dict representation of parsed YAML file.
79     """
80
81     with open(path, encoding="utf-8") as yaml_file:
82         loaded_config = yaml.safe_load(yaml_file)
83     return loaded_config
84
85
86 def dynamically_import_class(name: str) -> Type:
87     """
88     Dynamically import class by module path.
89     Adapted from https://stackoverflow.com/a/547867
90
91     Args:
92         name (str): path to the class (e.g., mypackage.mymodule.MyClass)
93
94     Returns:
95         __class__: class type.
96     """
97     try:
98         module, class_name = name.rsplit(".", 1)
99         mod = __import__(module, fromlist=[class_name])
100         class_ = getattr(mod, class_name)
101     except ModuleNotFoundError as err:
102         py_logger.error(
103             f"Module not found when trying to dynamically import '{name}'. "
104             "Make sure that the module's file is reachable from your current "
```

or in the code.

```
Filter symbols
```

- const py_logger
- func normalize_tracking_uri
- func generate_random_name
- func load_yaml
- func dynamically_import_class
- func flatten_dict
- class SignatureInspector
 - const INFTY
 - func __init__
 - func has_varargs
 - func has_kwargs
 - func required_params
 - func min_params_num
 - func max_params_num
 - func clear_key
 - func make_config_paths_abso...
 - func get_root_cause
 - func to_uri

Understand Anything in the code~

Understand Anything

Turn any codebase, knowledge base, or docs into an interactive knowledge graph you can explore, search, and ask questions about.

Works with Claude Code, Codex, Cursor, Copilot, Gemini CLI, and more.



[English](#) | [简体中文](#) | [繁體中文](#) | [日本語](#) | [한국어](#) | [Español](#) | [Türkçe](#) | [Русский](#)

[Quick Start](#) [License](#) [MIT](#) [Claude Code](#) [Codex](#) [Copilot](#) [Copilot CLI](#) [Gemini CLI](#) [OpenCode](#) [Vibe CLI](#) [Trae](#) [Homepage](#) [Live Demo](#)

YOUR CODEBASE

Understand Anything

We don't just map your code. We teach it.

YOUR GUIDED TOUR

1 - SCAN 2 - MAP 3 - TEACH

CODE STRUCTURE

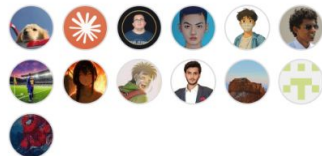
BUSINESS LOGIC

KNOWLEDGE BASE

Guided Tours • Semantic Search • Diff Impact • Layer View

Works with Claude Code • Codex • Cursor • Copilot • Gemini CLI • OpenCode • and more

Contributors 40

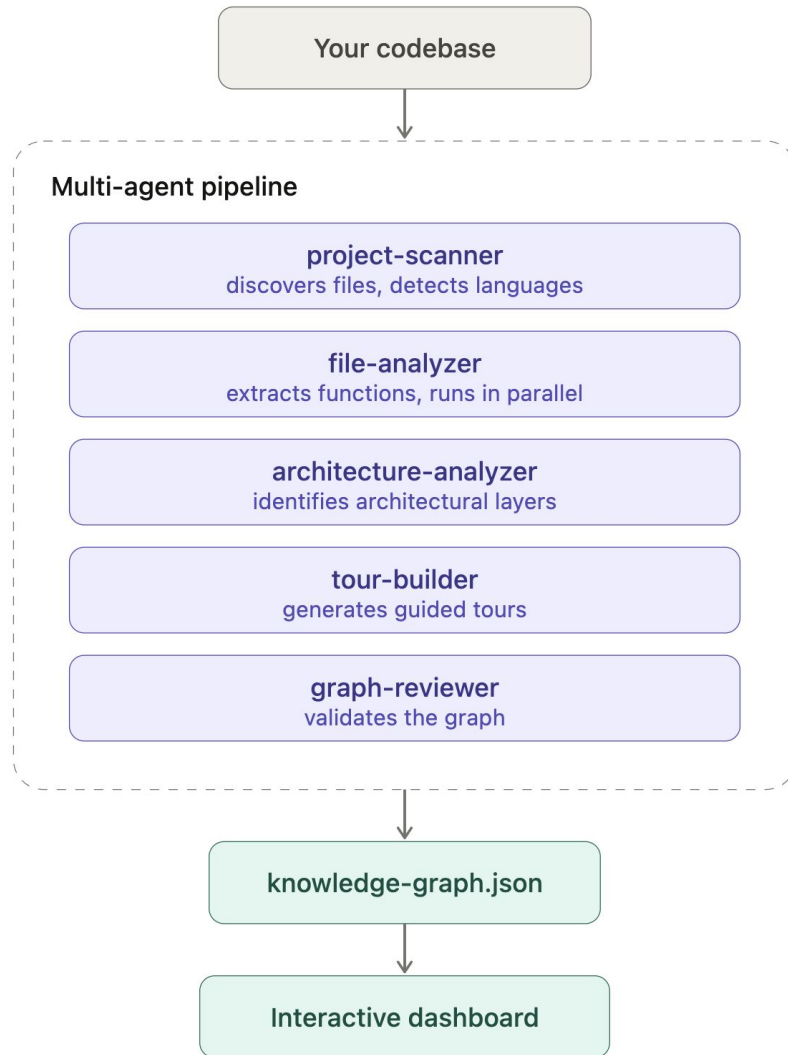


[+ 26 contributors](#)

Languages



- TypeScript 70.4%
- JavaScript 15.7%
- Astro 2.7%
- PowerShell 0.5%
- Python 9.7%
- CSS 0.5%
- Other 0.5%



Understand Anything in the code~

+ / - ?

- Took 300k Claude tokens to analyse the repository
- AI repositories often have similar structure
- Useful for the onboarding

From Query to Plot: Implementing a Tool-Based LLM Framework for ATLAS Analysis

G. Watts (UW/Seattle)

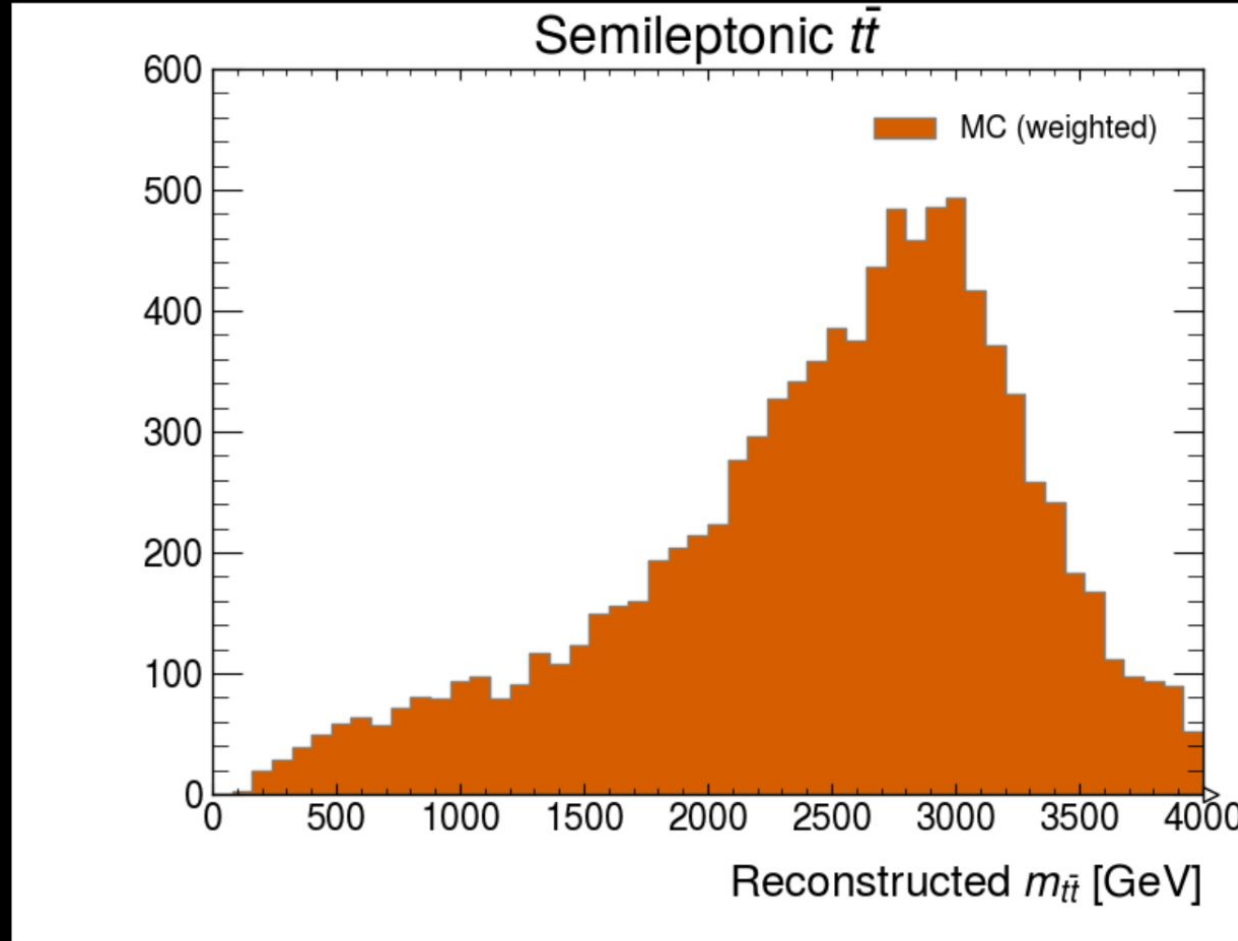


CHEP 2026

2026-05-28



On behalf of the ATLAS Collaboration



“Plot the p_T of all jets with $|\eta| < 1$ in
the rucio data set `<rucio-did>`”

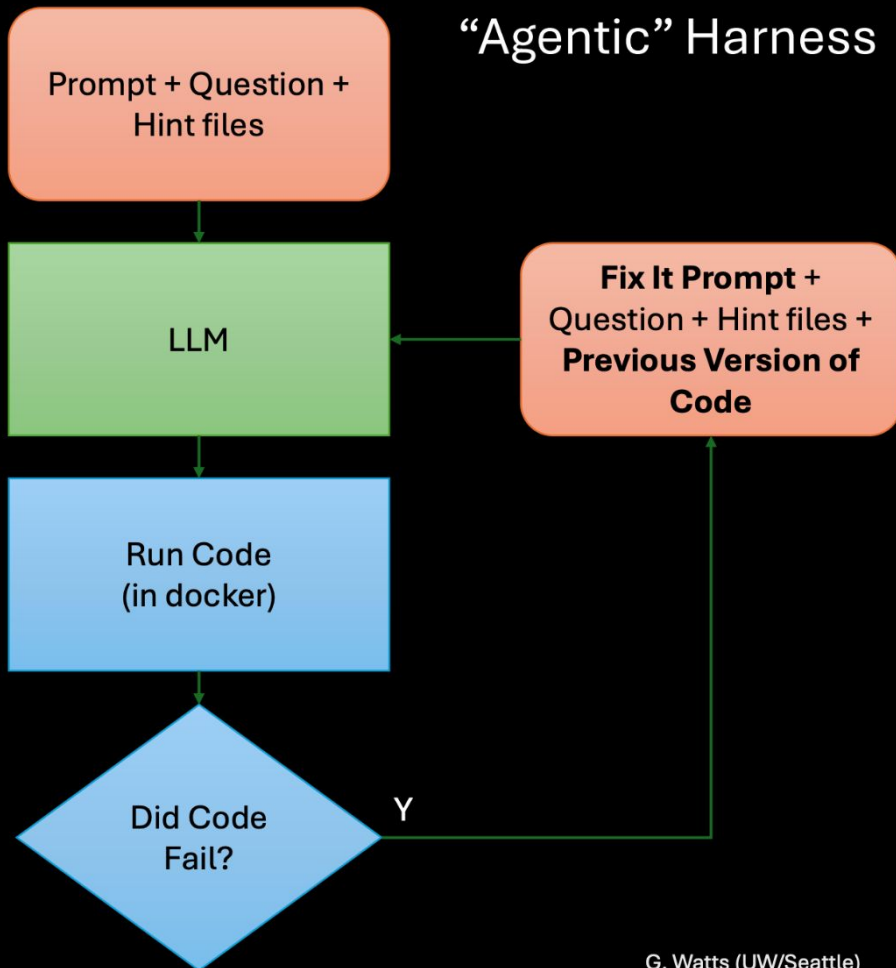
“Plot **pT of the trijet system** with the mass closest to 172.5 GeV in each event and plot the **maximum b-tagging discriminant** value among the jets in the triplet in the rucio data set <rucio-dataset>.”

Can We Make This “realistic”

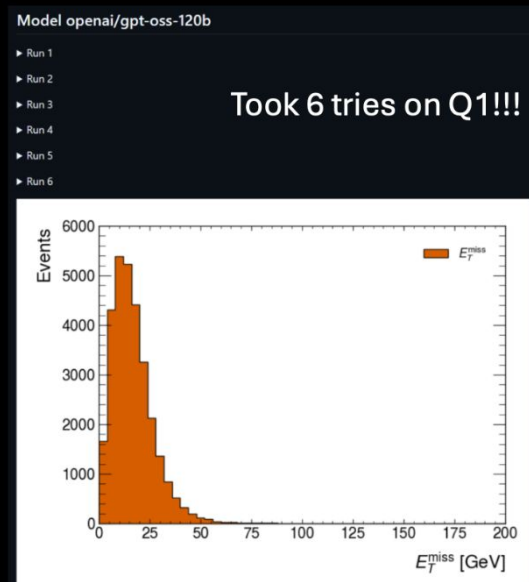
What do **agents** need to:

1. Turn that into actual python code
2. Access experiment resources
 - GRID data!
3. Produce the plot

“Agentic” Harness

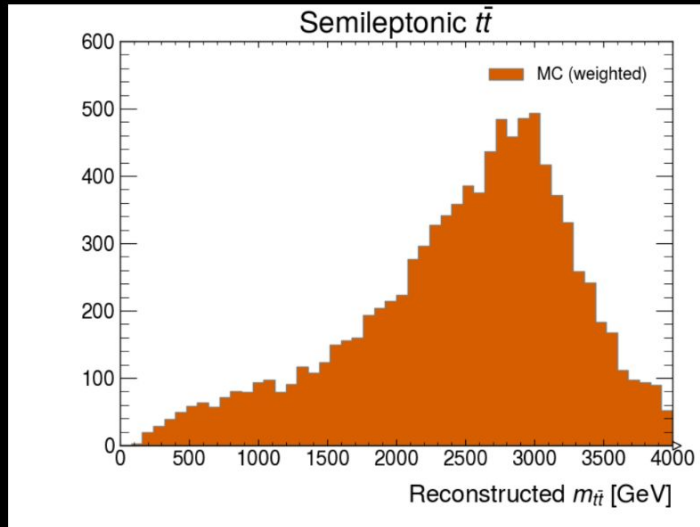


- Fix It Prompt is very similar to first prompt
 - Includes old code
 - stderr, stdout
- Automatic Iterations are allowed up to some limit



3 TeV Z-Prime

Plot the reconstructed top quark pair mass in events with one charged lepton in the rucio dataset <z-prime-mc-3000-gev>



- “Not unreasonable” – after examining the code
- Could have used b-tagging to improve further
- Used m_W mass constraint to choose the smaller neutrino p_z
- GPT-5 made two coding errors
 - That it had never made in other questions
 - But many other models had
 - Getting to the limits of what it can do in one shot?
- BUT... injects preconceptions
 - Initial Object Selection Cuts, etc.
 - So plots aren't identical to reference
 - But still is getting “physics” right
 - Reproducible? Not really!

Demo, Claude Code

Scenario: A physicist asks Claude:

“What branches are in this file? Show me the muon p_T distribution and fit it with a Gaussian.”

What Claude does — automatically:

1. `list_branches`: discovers tree structure
2. `get_branch_stats`: quick summary stats
3. `fit_histogram`: Gaussian fit on p_T
4. `plot_histogram_1d`: saves PNG plot
5. Summarises results in plain language

No script written. No copy-paste. No debug cycle.

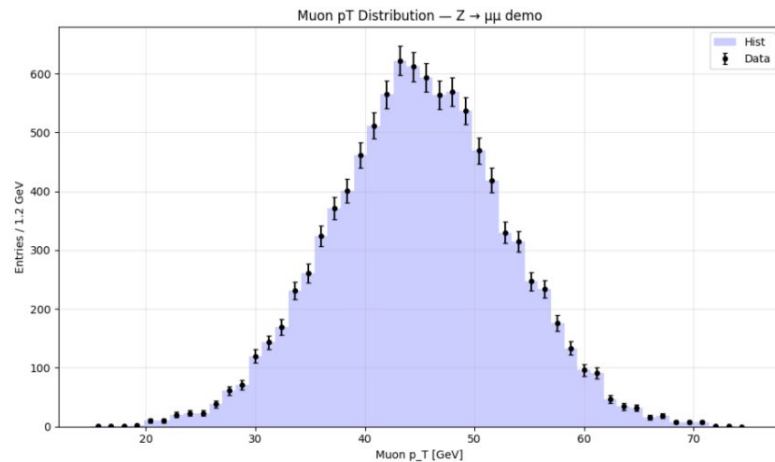


Figure: The Distribution and fit for our sample data

The Problem: LLMs Are Blind to Your Data

Today's Workflow

- 1 Describe problem to LLM in words
- 2 LLM writes a Python/C++ script
- 3 Copy-paste, debug, run locally
- 4 Interpret results yourself
- 5 Repeat ...

Privacy Constraint

Sending `.root` files to a cloud API is often **not an option** (proprietary data, CERN/institutional policies, large file sizes).

What We Want

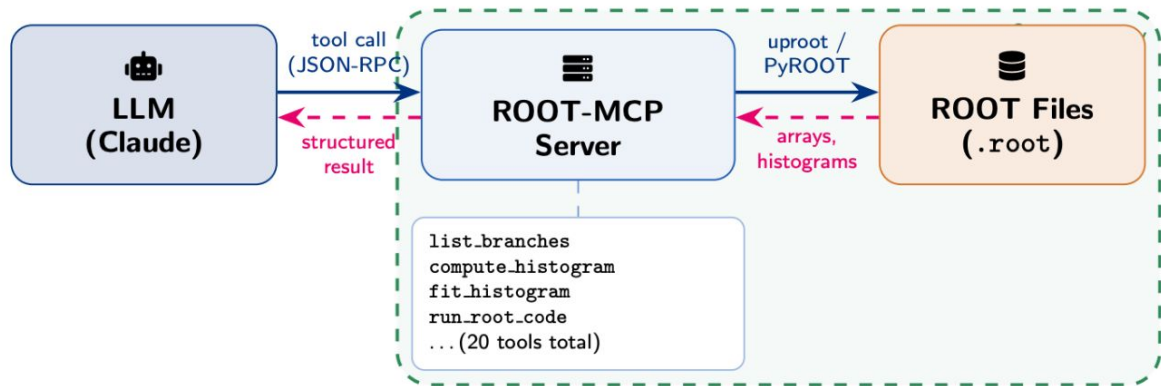
- 1 Ask the LLM a physics question
- 2 LLM calls local tools directly
- 3 Gets histogram, stats, plot **immediately**
- 4 Answers your question with evidence

Key Insight

The LLM should have **hands**, not just a mouth. Tools that run *locally* bridge the gap without ever uploading your data.

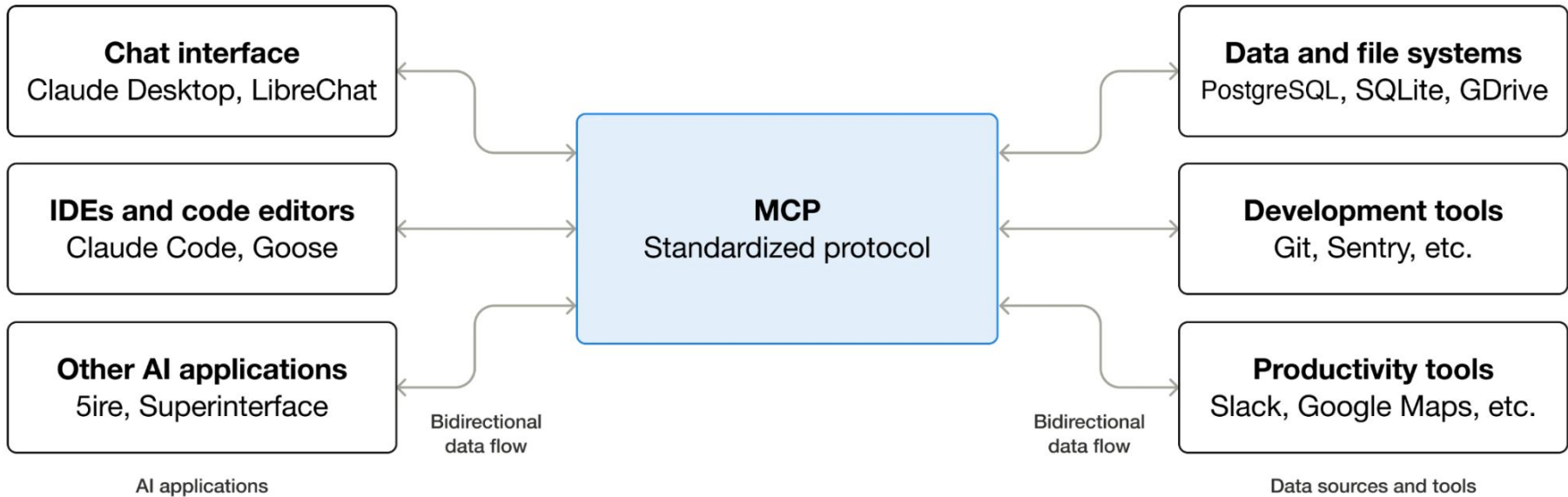
Model Context Protocol (MCP)

- ▶ Open standard from Anthropic (2024), now adopted broadly (Claude, Copilot, ...)
- ▶ Defines a **JSON-RPC contract** between an LLM agent and a local server
- ▶ Server exposes **Tools, Resources, and Prompts**
- ▶ Think: *USB-C for LLM capabilities*



★ Why It Matters

Data **never** leaves your machine.
The LLM only sees *structured results*,
not raw file bytes.



Under the Hood: MCP Protocol Exchange

LLM → Server (tool call):

```
{
  "name": "fit_histogram",
  "arguments": {
    "path": "/data/muons.root",
    "tree_name": "Events",
    "branch": "Muon_pt",
    "bins": 50,
    "range": [0, 150],
    "model": "gaussian",
    "selection": "Muon_pt > 20"
  }
}
```

Server → LLM (structured result):

```
{
  "model": "gaussian",
  "parameters": {
    "amplitude": 12543.2,
    "mean": 42.31,
    "sigma": 8.07
  },
  "chi2_ndf": 1.14,
  "entries": 98432,
  "fit_range": [0, 150]
}
```

The LLM never sees raw array data — only **structured physics-meaningful summaries**. This is what makes the interaction efficient: a complete fit result in a handful of tokens.

ROOT-MCP at a Glance

One-liner

An MCP server that lets LLMs *inspect*, *analyze*, and *visualise* CERN ROOT files, locally, safely, without writing a single script.

- ▶ **20 physics-aware tools** exposed via MCP
- ▶ **pip-installable**: `pip install root-mcp`
- ▶ **Zero config** to get started: `root-mcp --data-path /my/data`
- ▶ Works with **Claude Desktop**, Claude Code, and any MCP-compliant agent
- ▶ XRootD remote file support (`root://`)

20 Tools

3 Modes

140 Tests passed

Core deps: `uproot` · `awkward` · `numpy`
`pandas` · `scipy` · `matplotlib` · `mcp`

Tool Catalog

Discovery

`list_files`
`inspect_file`
`list_branches`
`validate_file`
`get_server_info`

Analysis

`compute_histogram`
`compute_histogram_2d`
`fit_histogram`
`compute_invariant_mass`
`compute_correlation`
`histogram_arithmetic`

Native ROOT (optional)

`run_root_code`
arbitrary PyROOT/Python
`run_rdataframe`
RDataFrame histograms
`run_root_macro`
C++ via gROOT.ProcessLine

Data Access

`read_branches`
`get_branch_stats`
`export_data` (JSON/CSV/Parquet)

Plotting

`plot_histogram_1d`
`plot_histogram_2d`

Control

`switch_mode`
core ↔ extended, live

All tools accept physics selection cuts (C++ syntax): `"pt > 20 && abs(eta) < 2.5"`