MANNA: MCP Architecture for NRAO and NOIRLab Archives

Summer Team for Astronomical Benchmarking and LLM Engineering (STABLE)

Dan Gause[1], Zoie Telkamp[2], Adele Plunkett[1], Brian Mason[1], Robert Nikutta[2], Stephanie Juneau[3], Adam Zacharia Anil, Ryan Loomis[1], Stella Offner[4], Greg Durrett[5], Paul Torrey[2], Eric Murphy[1]

1 National Radio Astronomy Observatory 2 University of Virginia 3 NSF NOIRLab 4 University of Texas at Austin 5 New York University

dgause@nrao.edu

01 The problem

Astronomers increasingly need data from archives outside their own specialty. Studying a single galaxy across wavelengths might mean querying NOIRLab's Astro Data Lab and the NRAO archive, both with a different query language and interface. Learning them takes time, and small mistakes are easy to make. We wanted a way to ask any of these archives for data in plain language.

Our Solution MANNA is an MCP server that connects a language model to astronomical archives through IVOA standards. From a Jupyter notebook a researcher asks in plain language; the model chooses the appropriate tools, runs the queries, and returns the results into the notebook — so the researcher does not have to write ADQL or learn each archive's schema.

02 What is an MCP server, and why does it exist?

A language model can write a query, but it cannot run one. By itself it has no way to reach a database, download an image, or call a service. The **Model Context Protocol** (MCP) is an open standard, released by Anthropic in 2024, that gives a model that access. It defines a single, consistent way for a model to call external tools, so each tool no longer needs its own custom integration.

An **MCP server** is a program that provides those tools. Any MCP-capable assistant can use it. Ours provides tools for astronomical archives and holds the knowledge of how each archive works, so the model does not have to. You ask a question, the model chooses a tool, and the server runs the query and returns the result.

03 What are IVOA standards?

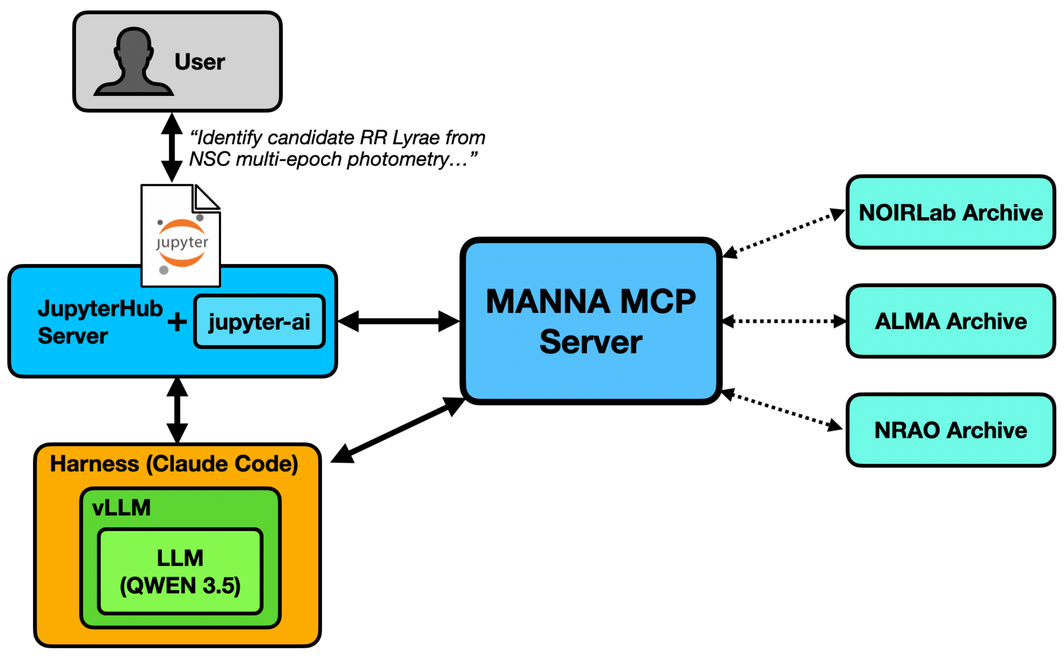
The International Virtual Observatory Alliance (IVOA) is a global consortium of national virtual-observatory programs. It defines the shared standards that let astronomical archives be searched the same way, no matter who operates them. Instead of a separate interface for every archive, a client speaks a handful of common protocols and any compliant archive answers. Our server is built on these standards — which is why one set of tools reaches Astro Data Lab, NRAO, ALMA, and beyond.

- **TAP (Table Access Protocol)** — query catalogs and observation tables with ADQL, a SQL-like language.
- **SIA (Simple Image Access)** — find and retrieve images by sky position and wavelength.
- **Cone Search (SCS)** — the simplest positional query: everything within a radius of a point.

04 What MANNA is made of

- **Built on FastMCP** — an open python toolkit for making MCP servers. It's what lets any assistant connect and use the tools.
- **A handful of tools** — about a dozen, one per job: find an object's coordinates, run a query, look up a table, search and fetch images.
- **One way in to every archive** — each archive's standard service is wired up once, so every tool behaves the same across NOIRLab, ALMA, and NRAO.
- **Handles big jobs** — large queries run in the background; the server tracks them and returns the results when they're ready.
- **Knows each archive's quirks** — a small built-in knowledge base tells the model how each archive and table behaves before it queries.

05 Jupyter Deployment Architecture




```python
import pyvo
from mcp.server.fastmcp import FastMCP

mcp = FastMCP("astro-archives")

@mcp.tool()
def vo_tap_query(endpoint: str, adql: str, maxrec: int = 10_000) -> dict:
    """Run an ADQL query against any IVOA-compliant TAP service.

    For an unfamiliar archive, call vo_archive_list first for its usage
    notes (non-standard tables, async-only routing, ADQL quirks).
    """
    service = pyvo.dal.TAPService(endpoint)                # connect via pyvo
    table = service.search(adql, maxrec=maxrec).to_table()  # type: ignore
    return {"columns": table.colnames,
            "rows": [tuple(row) for row in table],
            "row_count": len(table)}
```

06 A query, start to finish

A workflow in a jupyter-ai notebook: an agent (Claude Code) drives a local model and calls MANNA's tools, which query the archives.

"Get g and r photometry for stars around the cluster M5 and plot a color-magnitude diagram"

1. Resolve the name **M5** to sky coordinates.
2. Query NOIRLab's **Astro Data Lab** for g and r magnitudes of stars in the field
3. Load the returned catalog into the **notebook** as a table
4. Compute **g - r** and plot g against g - r
5. The notebook renders the color-magnitude diagram – the cluster's main sequence and red-giant branch.

MANNA GitHub