Simon Biggs, M.S.
Stuart Swerdloff, PhD

An open source PyMedPhys harness for interfacing MOSAIQ® with Claude®

Utilising an LLM to interrogate an Oncology Information System

Disclosures of pecuniary interests:

Stuart Swerdloff is an employee of ELEKTA

Simon Biggs is an employee of Anthropic

We are both maintainers of PyMedPhys

Anthropic provided credits that went into this work (I used about $50 worth)

Icons from amazon.com, freeicon.com, kindpng.com and vecteezy.com

Any opinions expressed are those of the authors and don't necessarily reflect that of our employers

Disclaimer/Reference: This is a non-clinical work for research purposes only. It is not a commercial product nor a commercial WIP. Access to copies of a MOSAIQ database and Claude.ai plans are not part of PyMedPhys.

Simon is from Australia. I live in New Zealand.

NZ has better ski fields.

Oz: Wagga Wagga

NZ: Cardrona (lots more snow now!)

NZ: Wanaka

Disclaimer/Reference

Motivation

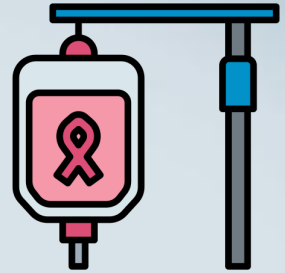
To provide a starting point for those who want to integrate LLMs into OISs (or really, any SQL database)

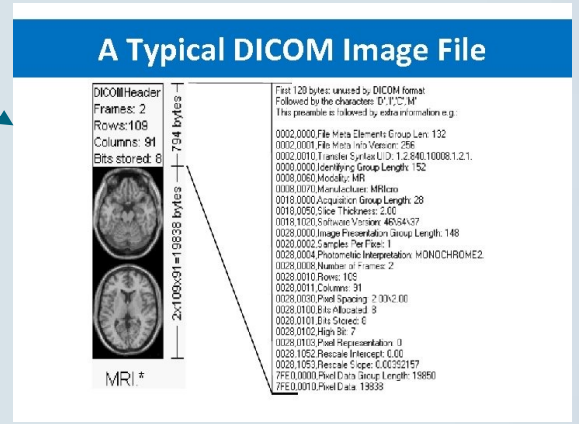
To provide a simple demonstration of sub-agents and tool use with accessible abstractions

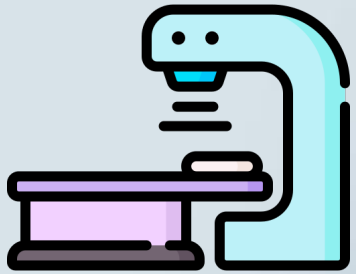
What is (in) MOSAIQ?

Oncology Information System:

- Relational Database + documents and DICOM objects
- Patient Demographics, Diagnosis, Prescription, Treatments, Schedule, History, and Outcomes





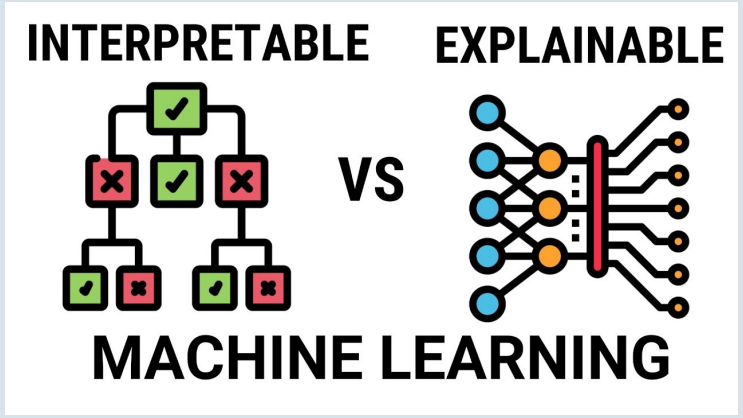


Disclaimer/Reference: This is a non-clinical work for research purposes only. It is not a commercial product nor a commercial WIP. Access to copies of a MOSAIQ database and Claude.ai plans are not part of PyMedPhys.

What is Claude?

Large Language Model with foundational goals of being:

- Reliable
- Interpretable
- Steerable

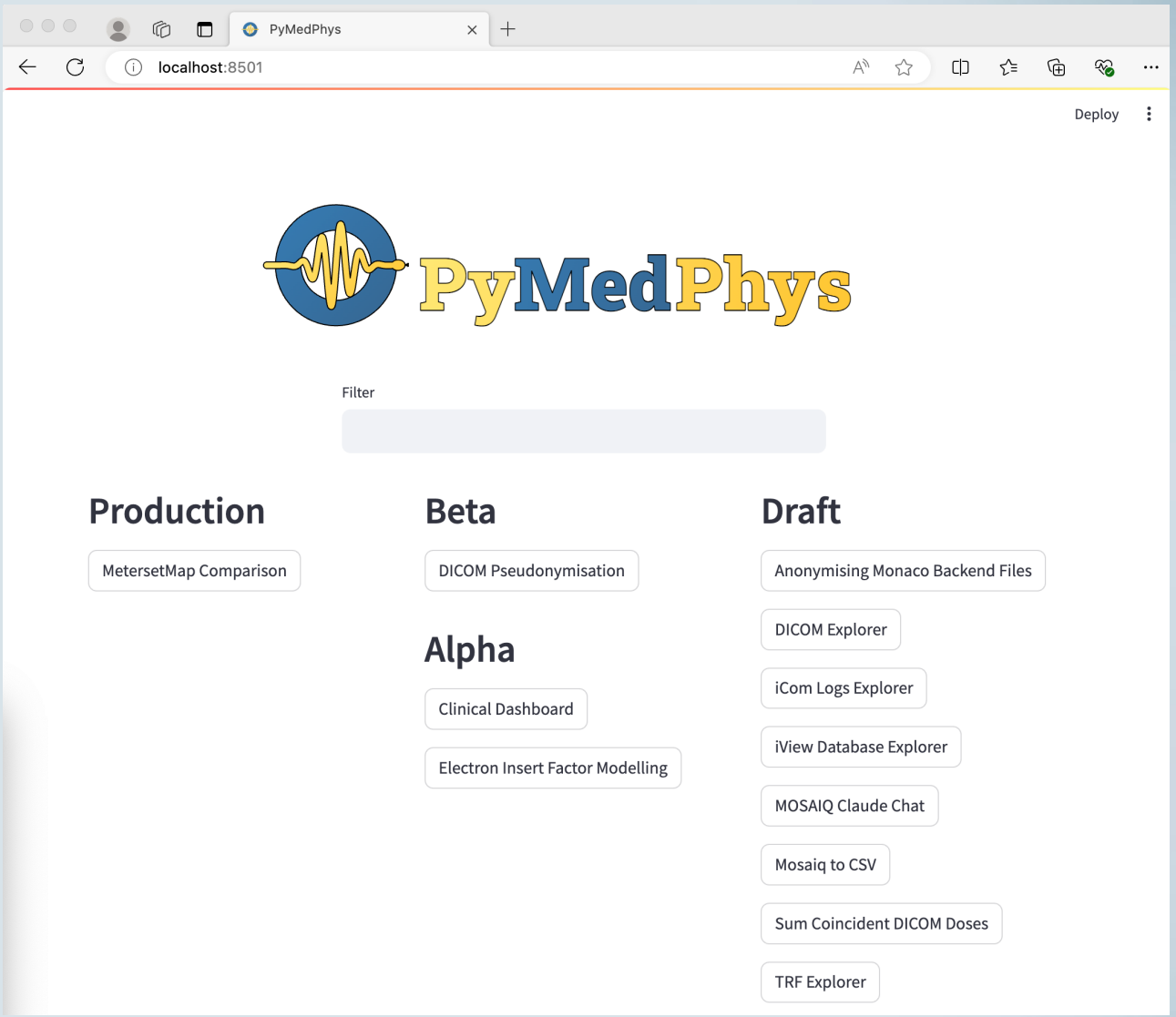


Disclaimer/Reference: This is a non-clinical work for research purposes only. It is not a commercial product nor a commercial WIP. Access to copies of a MOSAIQ database and Claude.ai plans are not part of PyMedPhys.

What is PyMedPhys

A community effort to develop an open standard library for Medical Physics in Python. Building quality transparent software together via peer review and open-source distribution. Open code is better science.

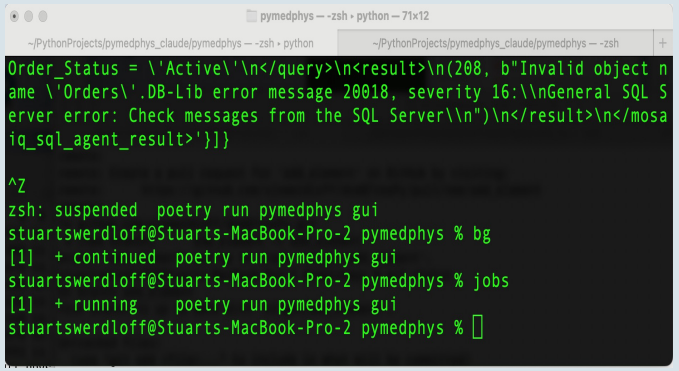
Repository: pymedphys/pymedphys ·
Tag: iccr_2024 · Commit: 5c214c7 ·
Released by: SimonBiggs

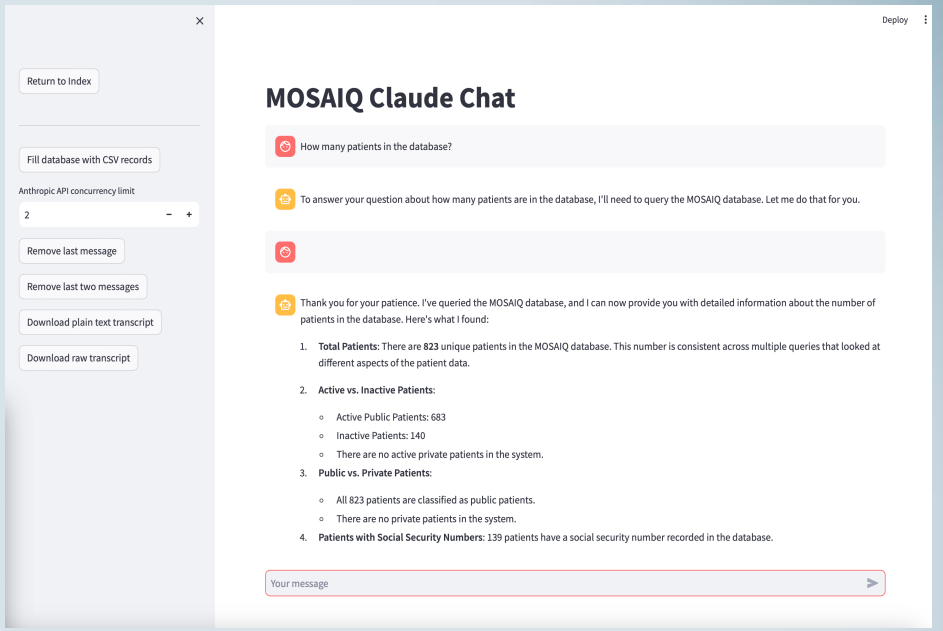


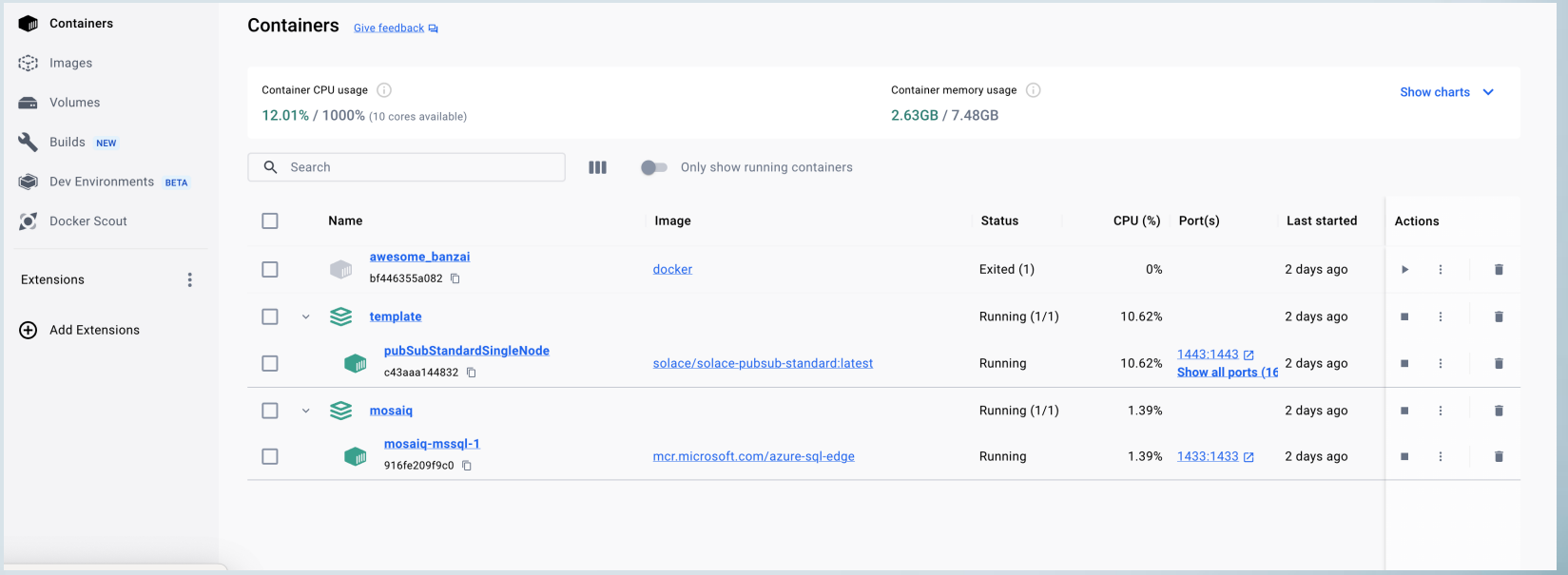
Disclaimer/Reference

High Level View

- Python using Streamlit GUI.
- MOSAIQ 3.x
- Claude 3.0 and 3.5 (was 2 at time of abstract).
- Docker container of azure-sql-server (podman can be used)







Disclaimer/Reference

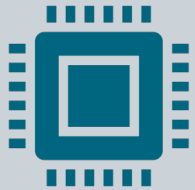
OS/Platform support

MS Windows 10/11

Linux (e.g. Ubuntu 2x)

MacOS on Apple Silicon is a bit touchy (pymssql 2.3.0 has bugs, dropping back to < 2.3.0 requires minor code change due to API change in pymssql).

Python 3.11



Pymedphys runs on Windows, Linux, and MacOS

Azure-sql-server container is Linux (host OS needs to be able to run docker…Win11 ARM as a VM on MacOS… not so much).

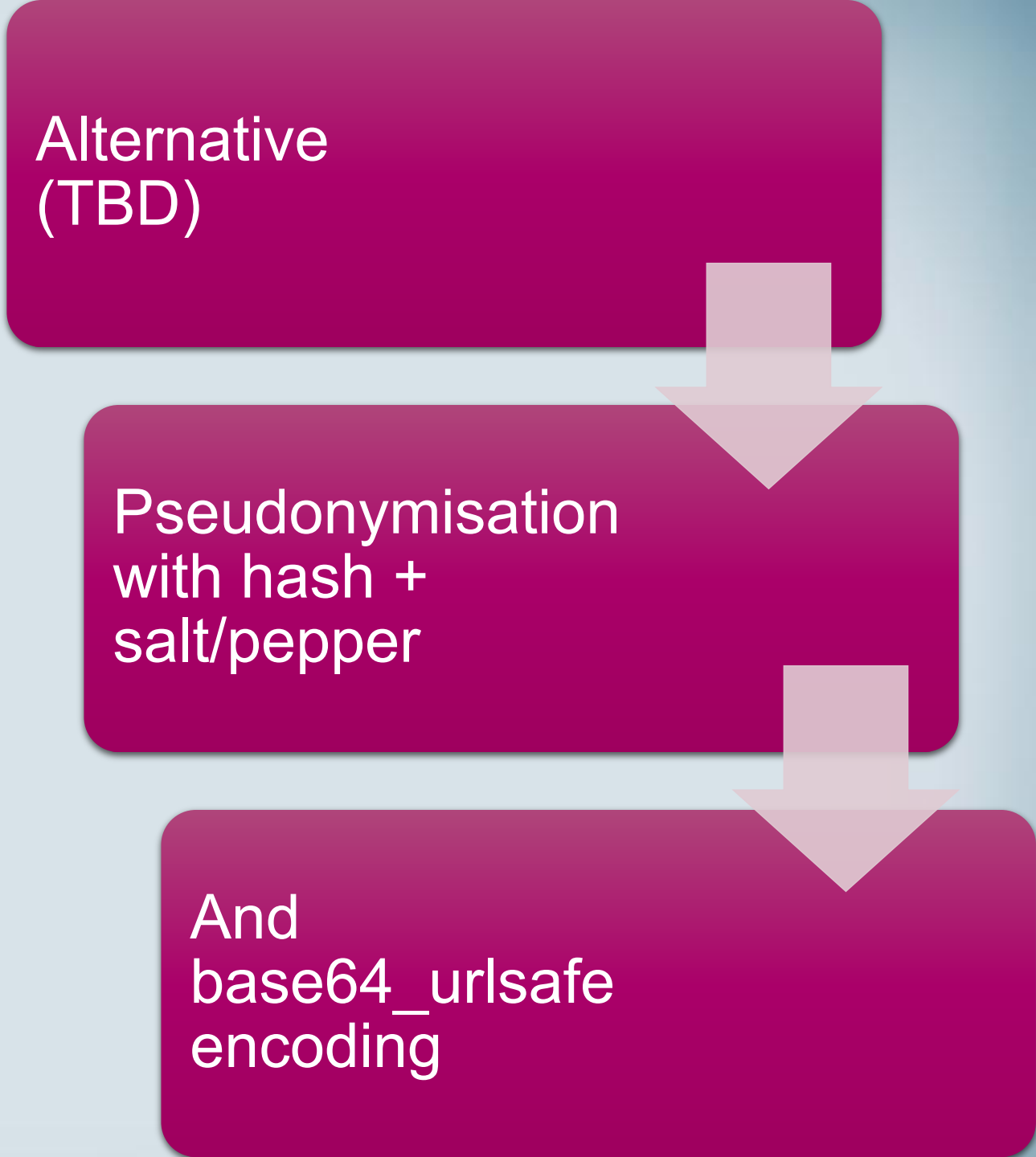
Private Health Information

Pynonymizer > 2.2

Flexible database anonymization tool good for initial prototyping with realistic substitutions

MOSAIQ specific strategy file

But… not optimal for large patient database

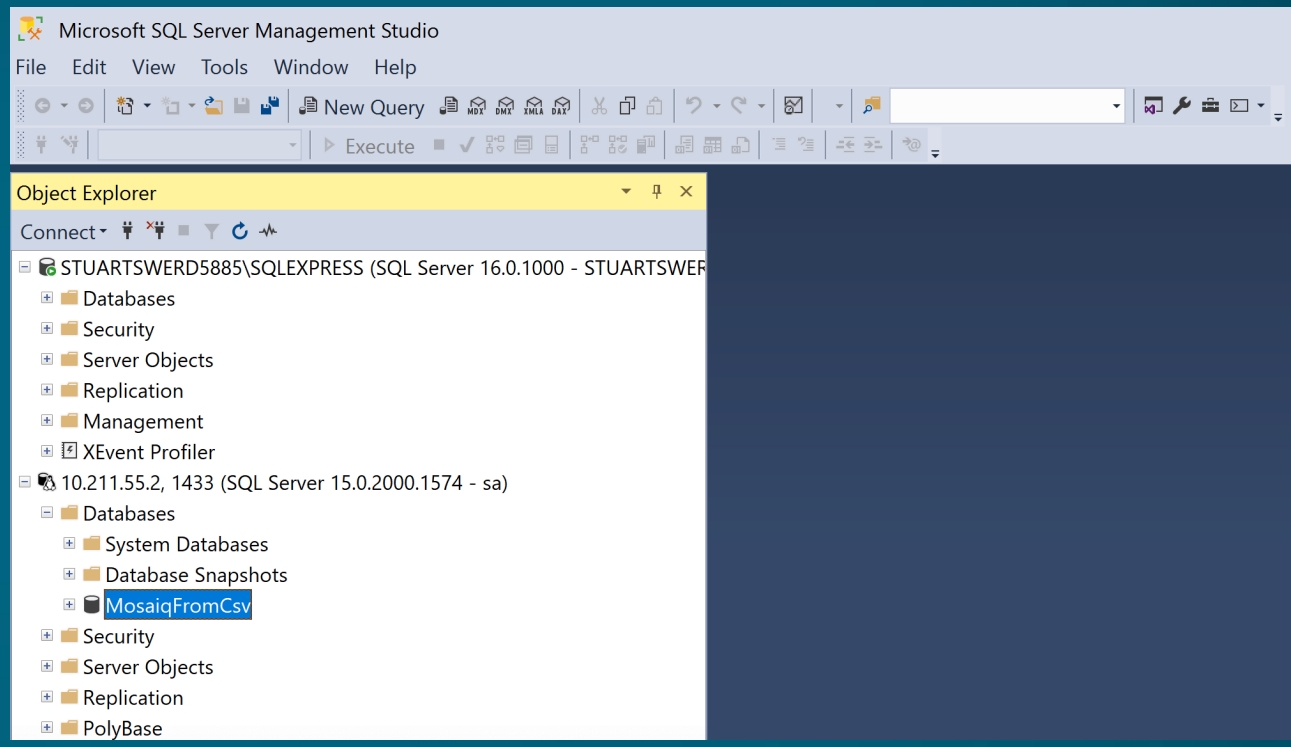


Test Data Set ("includes batteries")

Engineering (test) Database (intrinsically no PHI)

Subset extracted to CSV files for automated regression testing and "initial out of the box" demo.

You can get to the DB via port passthrough… (per docker file)



Replacing Dataset

A little bit of command line work to get your own dataset in place.

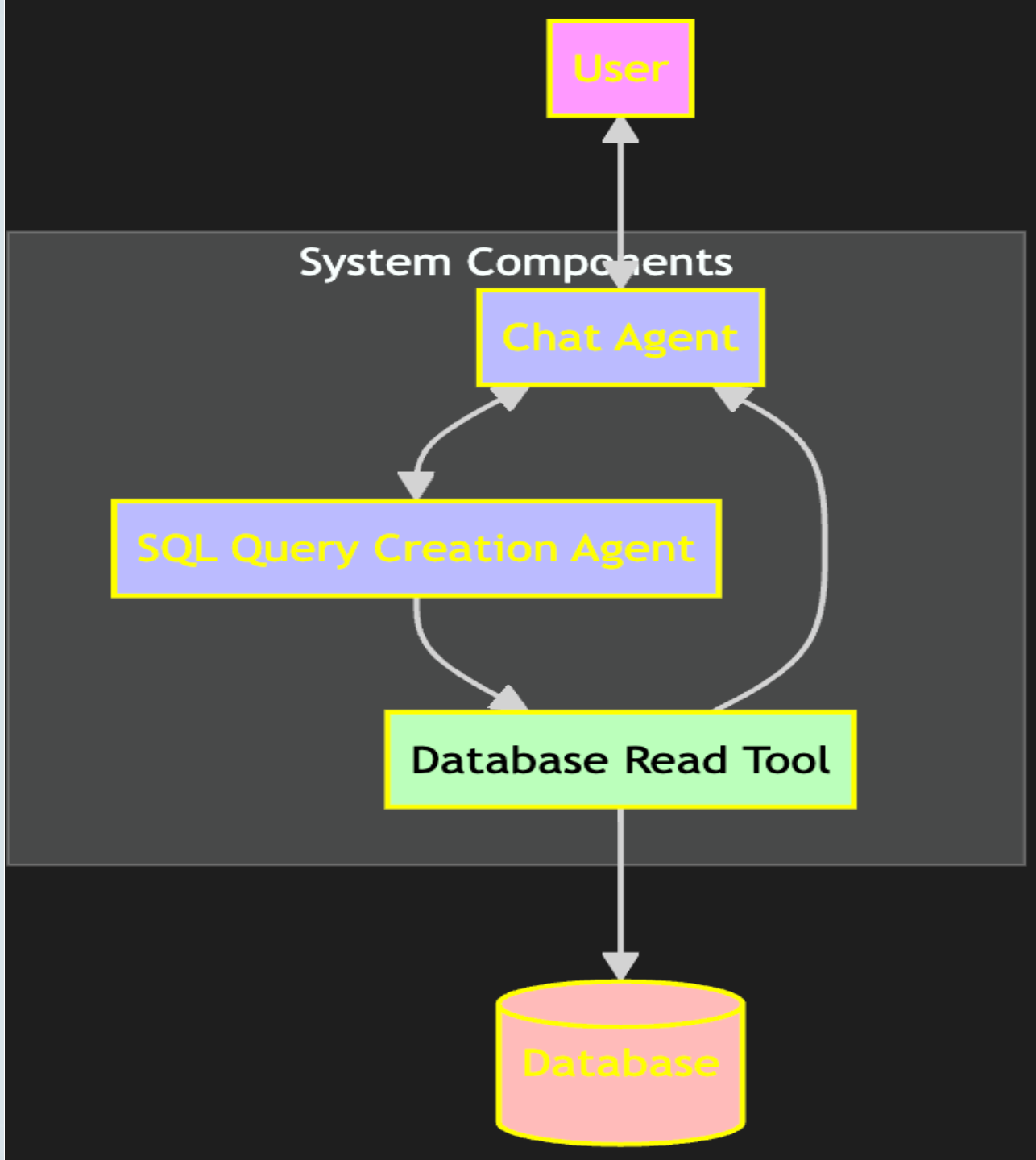
Create anonymized copy of your database.

Docker copy from host to container

Use SQL Server Management Studio (or sqlcmd) to overwrite the test database

Initial LLM harness architecture

- Chat agent for user interaction
- SQL query creation agent which returns queries based on a schema and the conversation history
- A standard database read tool which accepts the queries provided by the SQL query creation agent executes them, and then sends the results to the chat agent in order to converse with the user
- Expensive approach because all tables/columns re-sent each time (huge context).



Control and Observability options

- Number of Concurrent LLM API interactions
- Conversation is re-sent to Claude with each incremental message. ($$ implications)
- Removal and *Rollback* of messages
- Text Transcript of external agent dialog with user
- Raw (JSON) Transcript (useful for development replay) contains Sonnet conversation tool use. Includes SQL.
- Command line is printing details of agent conversations to screen.

Disclaimer/Reference

System prompt:

You are an MSSQL AI agent. You respond only with valid Microsoft SQL Queries encompassed within <query> tags.

You always provide 3 unique and diverse queries.

Some queries that you request may not return a result, and some tables within the schema may not be relevant to your needs. So make sure that each of your queries targets different tables within the database so that you get good coverage over possible solutions and are able to return something meaningful.

…

All queries assume the following database schema: schema

Questions used in our test

1. Please find the Patient ID that has the most radiotherapy treatments.

2. Please find the most common cancer diagnosis that patients have had

3. What is the most common number of fractions used to treat patients? Do not give a range, instead, find exactly the most common fraction number.

Results (note that this is a test database, not clinically realistic)

MOSAIQ database “expert” (me):

Patient ID that has the most radiotherapy treatments:
10003
Interpretation: Which patient received the most treatment fractions to a prescription site. Use PAT_ID1 to identify.
Confidence of expert: Very High

Most Common Cancer Diagnosis:
Prostate
Interpretation: Category in Medical table
Confidence of expert: Medium (there are other tables containing relevant information)

Most common number of fractions used to treat patients:
10
Interpretation: Number of Fractions in prescription (Site table)
Confidence of expert: Very High

Initial results (Claude 2 and initial harness)

The agent was only given one shot in each attempt and was not given the opportunity to refine or further verify its answer. For all testing we used the publicly available version of Claude, version 2.1 via the Anthropic Python PyPI pack- age version 0.14.0. The Mosaiq DB utilised was a testing instance that corresponds to the same schema as that used by MOSAIQ version 3.1.3.

What is the most common number of fractions used to treat patients?	Count
Agent said it couldn't find a result	20
1 fractions	4
10 fractions	3
0 fractions	1
14 fractions	1
23 fractions	1
152 fractions	1
Answered wrong question	1

Disclaimer/Reference

Patient ID that has the most radiotherapy treatments	Count
Agent said it couldn't find a result	18
10003	3
31587	2
31215	1

The most common cancer diagnosis that patients have had	Count
Malignant neoplasm of prostate	26
Agent said it couldn't find a result	21
The "NULL" diagnosis (no diagnosis entered)	3
Provided diagnosis codes	1
Prostate cancer	1
Bladder cancer	1
Malignant neoplasm	1
Answered wrong question	1

A little more work and a new version of Claude

- Claude Sonnet updated to 3.5 (anthropic 0.29.0)
- Use 3 Unique/Diverse Queries (to improve 3.0 Haiku results) per agent.
- 6 Creation and 4 Voter Agents
- Decrease from all table names to top 5

What is the most common number of fractions used to treat patients?	Count
1 fraction	28
6 fractions	1
Software exception/error	1

Patient ID that has the most radiotherapy treatments	Count
10106	20
10003	9
10027	1

The most common cancer diagnosis that patients have had	Count
Malignant neoplasm of prostate	17
Agent said it couldn't find a result	5
Other/Unspecified malignant neoplasm	3
Software exception/error	2
Base of tongue	1
Upper lip	1
Ac erythraemia	1

Reasons for Failures:

Names of database tables and columns don't map directly to natural language terminology (LLM can't find the appropriate tables).

Columns of same name (e.g. modality_enum) are in separate tables with enumeration values that are *not* the same.

Software Exceptions due to human coding blunders (use of various libraries).

Generated SQL is valid, but not always in T-SQL subset

The Fix:
Providing brief statement to Claude regarding name/description of relevant tables and distinct enumerated values improves results markedly and immediately. More precise questions can also make for a big improvement…

Modify input a bit

The Dose_Hst table has description: Field treatment or portal image history; The Site table has description: Prescription Site; The Medical table contains diagnosis information. The Topog table contains Topography information. The Medical table is linked to the Topog table.

I will ask three questions, I would like you to format your answer as a table containing two columns, the first column containing the number of the question and the second column containing the answer, without including additional detail or commentary.

Question 1: Please find the patient ID that has had the most radiotherapy treatments.

Question 2: Please find the most common type of cancer that patients have had.

Question 3: What is the most common number of fractions prescribed to treat patients? Do not give a range, instead, find exactly the most common fraction number.

Patient ID that has the most radiotherapy treatments	Count
10003	30

The most common type of cancer that patients have had	Count
Malignant neoplasm of prostate	26
Upper Lip, NOS	3
Agent said it couldn't find a result	1

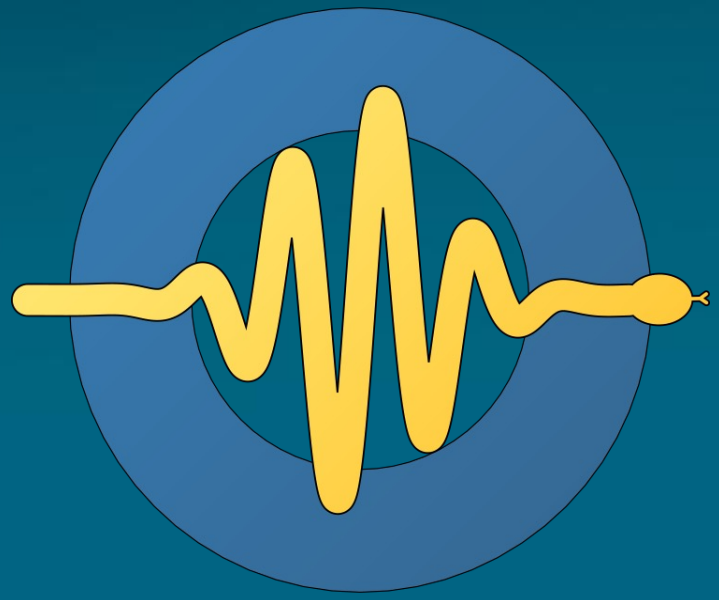
What is the most common number of fractions prescribed to treat patients?	Count
10	28
1	2

Updates to Claude motivate architecture change

- Cost performance changes
- Use Claude Sonnet 3.5 throughout
- Fuzzy search of tables and descriptions.
- Create and issue SQL queries

Motivations to not change the architecture

- Retain template for picking and choosing models (educational example)
- Encourage encoding of Table and column description in DB itself



https://github.com/pymedphys/pymedphys

Contributors welcome, but please read *The Contributors Guide*