Design Document

1. Overview

Starfire is an AI powered biomedical insights app designed to help healthcare professionals uncover insights from their data. It supports data analysis and visualization as directed by free-text user queries.

1.1 Purpose

Provide a clear, maintainable architecture that can scale to more complex tasks and datasets. Build a framework that is intuitive, reliable and delivers accurate insights.

1.2 Scope

- Streamlit frontend
- Python backend

2. Goals and Non-Goals

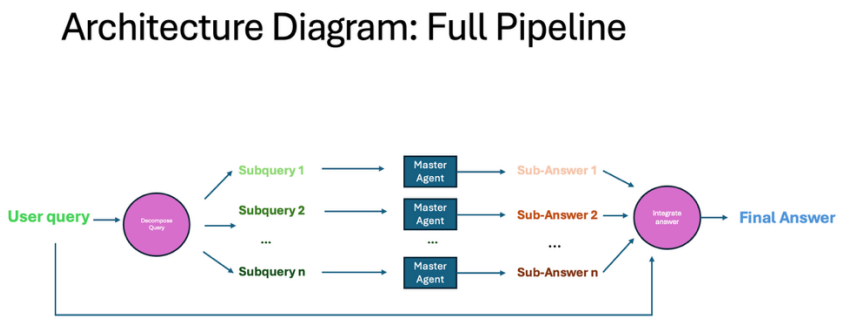
2.1 Goals

- Allow users to receive insightful and correct responses to their data analysis queries
- Provide real time updates on analysis progress in front end app
- Offer internal analytics on task accuracy and runtime

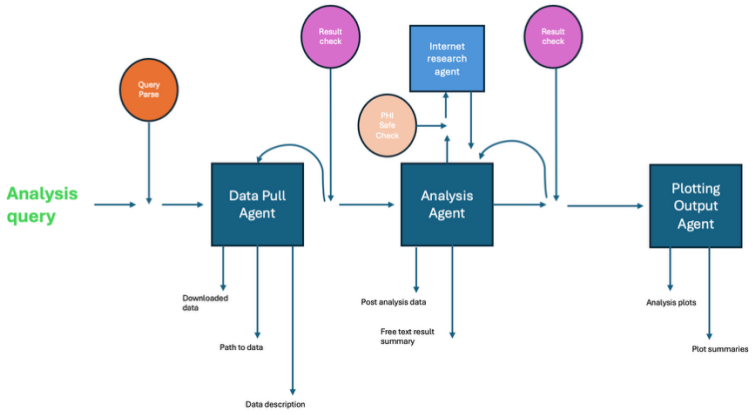
2.2 Non-Goals

- Non-local implementation
- Management of sensitive data (for prototype)

3. Architecture



Architecture Diagram: Master Agent



Tech Stack

- **Frontend:** Streamlit
- **Backend:** Python
- **LLM:**
  - inference: huggingface LLAMA 3.3
  - agentic: smolagents LLAMA 3.3
- **Deployment:** local macbook

4. Components

Frontend

- **Views:**
  - Assist mode: step by step analysis with intermediate visualization
  - Autonomous mode: return an answer to a complex query without intermediate user input

Backend

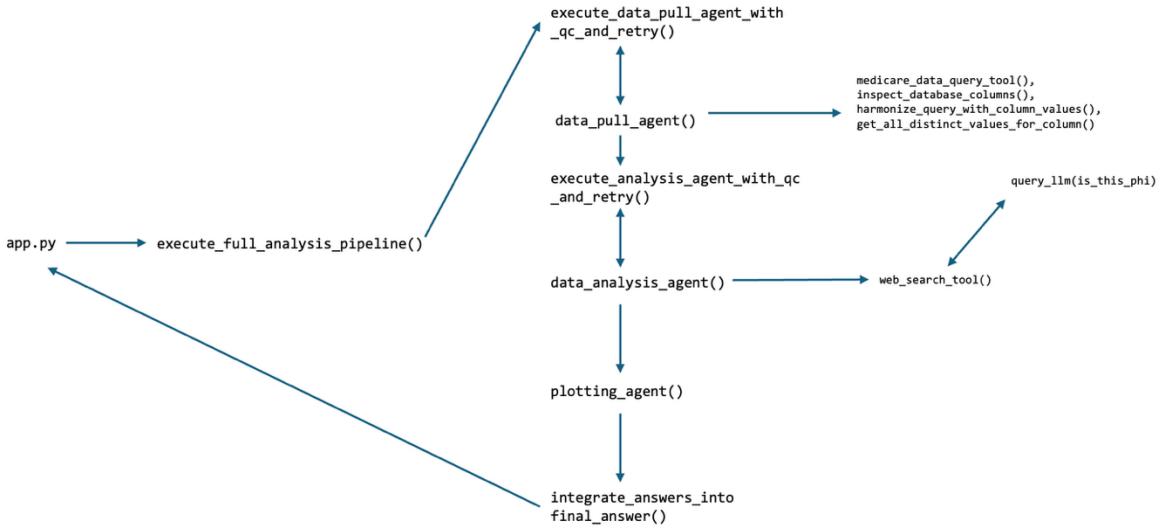
- **Agents:** smolagent code agents with optional tool functionality
- **Master process:** button triggered pipelines
  - Co-pilot mode: Trigger modular pipeline then return to application
  - Autonomous mode: Trigger entire pipeline and run until completion

5. Data Model

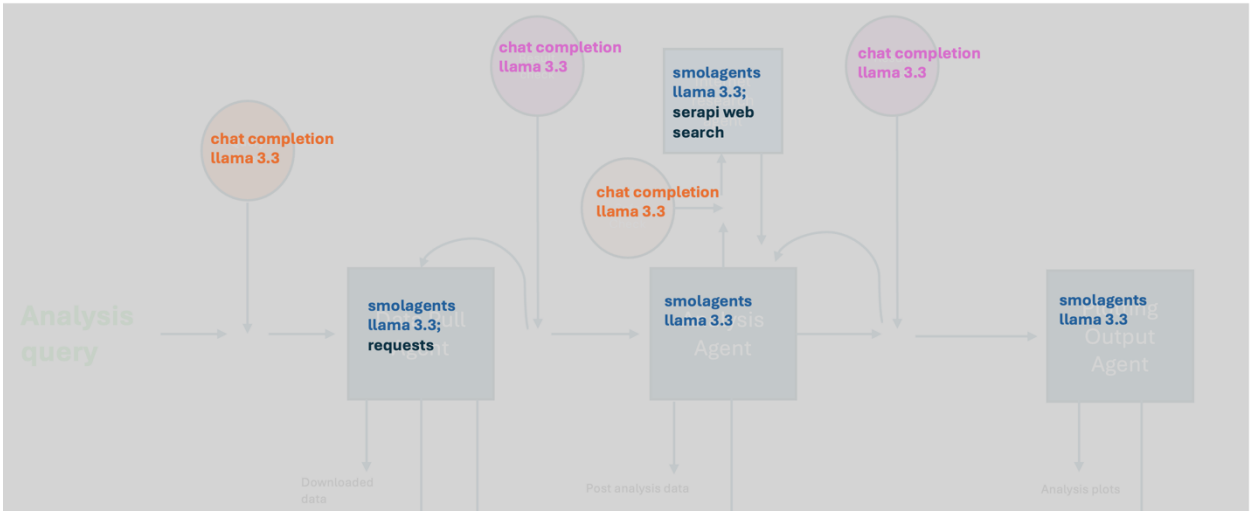
- Data is saved locally in subdirectories created by LLM
- Data is erased upon program termination
- Outputs are rendered to the app and can be saved

6. Sequence Diagram

Appendix: code module flow (autonomous mode)



Appendix: Component Services Diagram



8. Deployment

- Runs locally on a macbook with python 3.9.12
- Code on github

- Inference via Noah Friedman's hugging face account

9. Security Considerations

- Ensure that LLM API tokens are not exposed to user
- Ensure that improper data is not pulled
- Ensure that PHI data is not sent to non-secure APIs
- Ensure that PHI data is not persisted or saved where it shouldn't be

Core Library Documentation

- Smolagents
- Serapi
- Streamlit