

Data2Evidence Chat Assistant: An AI-Powered Code Assistant for OHDSI Strategus Analysis

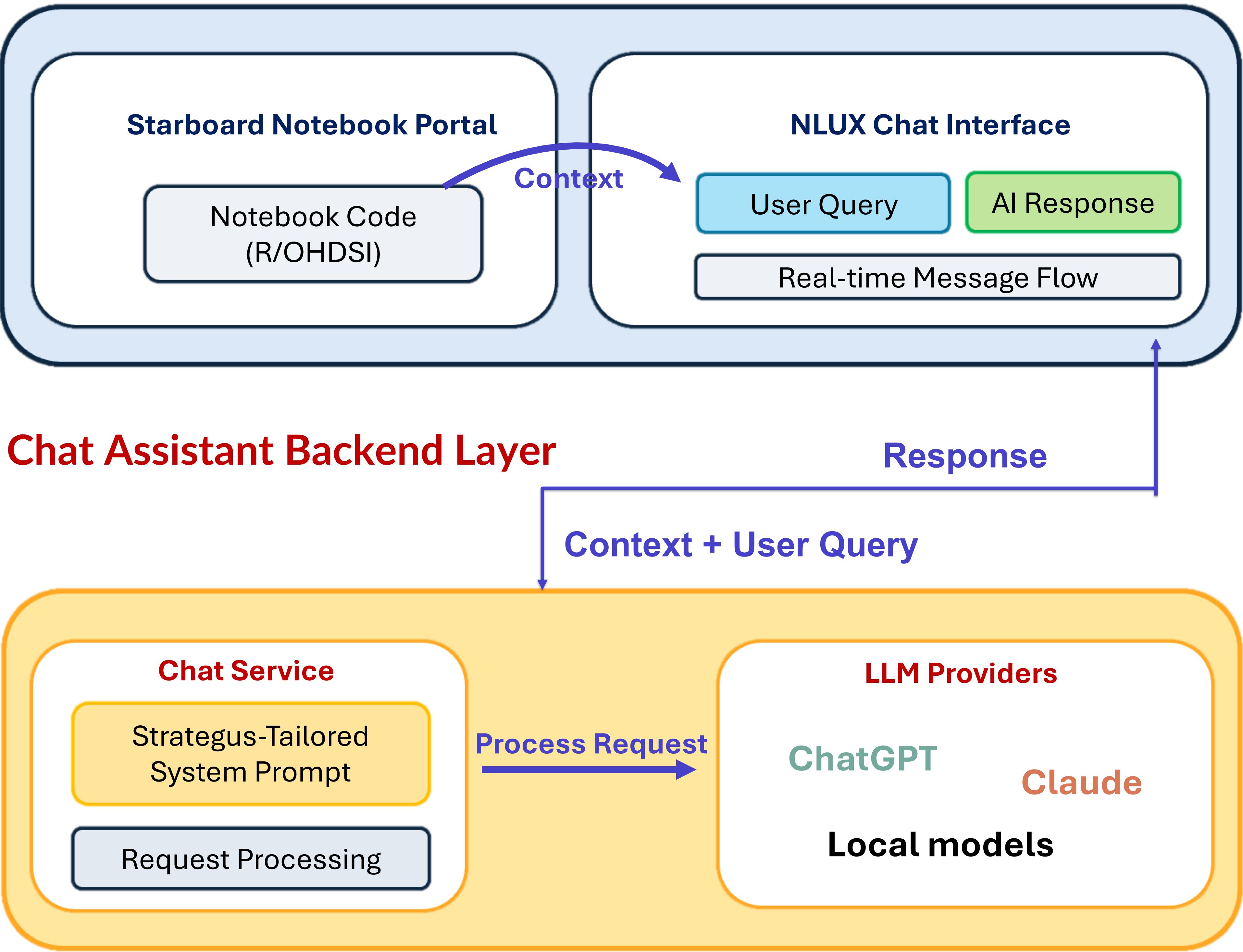
Background

The complexity of healthcare data analysis using the OHDSI Common Data Model (CDM) and Strategus framework presents significant challenges for researchers conducting network studies, requiring extensive knowledge and coding ability of R. The pain of coding can be dramatically relieved using modern large language models (LLMs) which have demonstrated remarkable capabilities in code generation and analysis, but their application to specialized healthcare domains requires careful architectural design and domain-specific prompting strategies.

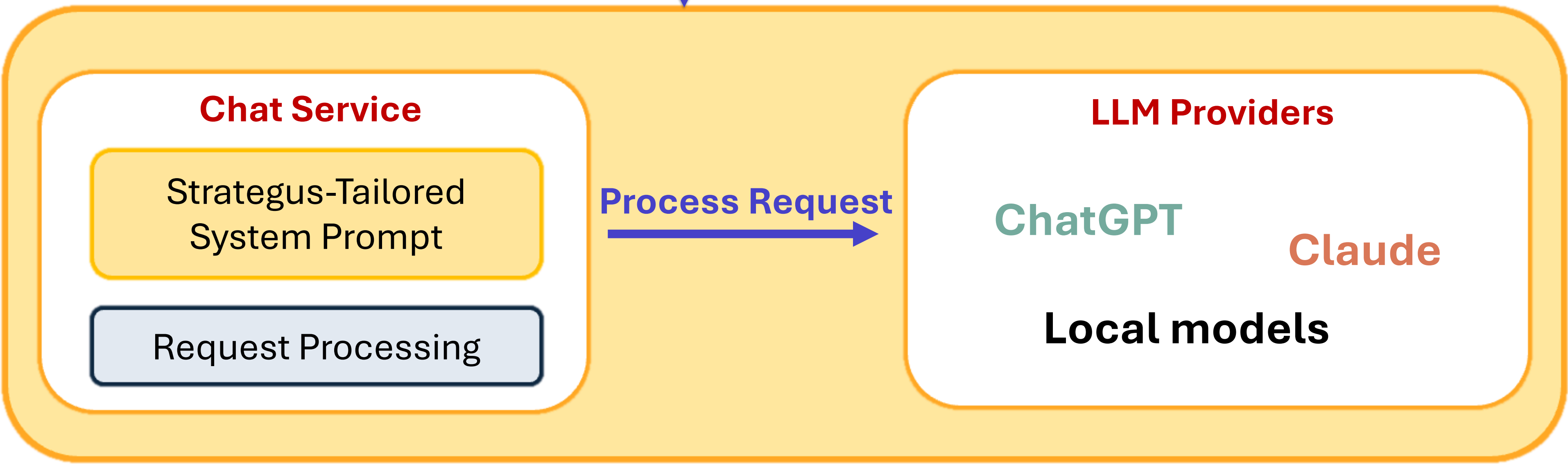
Methods and Results

- Apply NLUX AI framework as interaction layer
- Access third-party LLM models and local model through llama.cpp library
- Implement domain-specific prompt engineering

Chat Assistant Frontend Layer



Chat Assistant Backend Layer



```
cdmdefault Dataset Concepts Cohorts Notebooks Analysis Account
Share notebook Export Notebook Import Notebook New Notebook Save Delete Sync from Remo

example of strategus study for type2 diabetes

# Set-up for a Strategus study on type 2 diabetes
library(rD2E)
library(Strategus)

# Step 1: Define the cohorts
diabetesCohorts <- c(101, 102, 103) # Here, replace with actual cohort IDs for Type 2 Diabetes defini
cohortDefinitionSet <- rD2E::get_cohort_definition_set(diabetesCohorts)

# Step 2: Create shared resources and module specifications

## CohortGenerator (required for creating cohorts)
cgModule <- CohortGeneratorModule$new()
cohortDefinitionSharedResource <- cgModule$createCohortSharedResourceSpecifications(cohortDefinitionS
cohortGeneratorModuleSpecifications <- cgModule$createModuleSpecifications(generateStats = TRUE)

## CohortDiagnostics (can analyze differences between cohorts)
cdModule <- CohortDiagnosticsModule$new()
cohortDiagnosticsModuleSpecifications <- cdModule$createModuleSpecifications(
  runInclusionStatistics = TRUE,
  runIncludedSourceConcepts = TRUE,
  runOrphanConcepts = TRUE,
  runTimeSeries = TRUE,
  runVisitContext = TRUE,
  runBreakdownIndexEvents = TRUE,
  runIncidenceRate = TRUE,
  runCohortRelationship = TRUE,
  runTemporalCohortCharacterization = TRUE
)

## Characterization (to describe population)
chModule <- CharacterizationModule$new()
characterizationModuleSpecifications <- chModule$createModuleSpecifications(
  targetIds = c(101, 102), # Replace with cohort IDs representing target groups for characterization
  outcomeIds = c(103) # Replace with cohort ID representing outcome group (e.g., diabetes progre
)

## CohortMethod (for comparative cohort analysis)
cmModule <- CohortMethodModule$new()
cmAnalysisList <- list(CohortMethod::createCmAnalysis(analysisId = 1)) # Adjust settings as needed
targetComparatorOutcomesList <- list(
  CohortMethod::createTargetComparatorOutcomes(
    targetId = 101, # Replace with actual target cohort ID
    comparatorId = 102, # Replace with actual comparator cohort ID
    outcomeIds = c(103) # Replace with diabetes progression outcome ID
  )
)
cohortMethodModuleSpecifications <- cmModule$createModuleSpecifications(
```

Future work: Enhance the chat assistant's capabilities through MCP servers

- Generate specified cohort definition through interaction with users
- Direct cohort management through chat assistant