Tracil: AI-Powered Traceability Tool Across CDISC Standards

Kexin Guan, Scientist, Statistical Programming, Merck & Co., Inc.
Junze Zhang, Scientist, Statistical Programming, Merck & Co., Inc.
Anthony Chow, Executive Director, CDISC

Meet the Speakers

Kexin Guan

Title: Scientist, Statistical Programming

Organization: Merck & Co., Inc.

Kexin Guan is a Statistical Programmer at Merck & Co. Inc., where she has been part of the Oncology Early Development group since December 2022. She holds a Master's degree in Biostatistics and Bachelor's degree in Applied Mathematics and Statistics.

Junze Zhang

Title: Scientist, Statistical Programming

Organization: Merck & Co., Inc

Junze Zhang is a Scientist at Merck & Co. Inc., supporting early oncology statistical programming. He earned his Master's in Computer Engineering from NYU and Bachelor's in Computer Science from Oregon State University.

Disclaimer and Disclosures

- *The views and opinions expressed in this presentation are those of the author(s) and do not necessarily reflect the official policy or position of CDISC.*
- *The views and opinions expressed in this presentation are those of the author(s) and do not necessarily reflect the official policy or position of Merck & Co., Inc.*

Agenda

1. Background & Motivation
2. Demo
3. App Structure
4. Backend AI Workflow
5. Summary, Lessons Learned & Future Steps

Background & Motivation

The Traceability Problem

- Clinical data flow is **complex,** spanning **multiple** systems and **silos**.
- Each layer (Protocol → CRF → SDTM → ADaM → TLFs) adds transformation logic.
- Manual tracing lineage = **heavy review time**.

Our Project Goals

- **Automate Lineage Inference**
  - Extract relationships from existing metadata without manual mapping.
- **Support CDISC Standards End-to-End**
  - Protocol (USDM) ↔ CRF ↔ SDTM (Define.xml) ↔ ADaM (Define.xml) ↔ TLF (ARS)
- **Provide Explainable AI Results**
  - Every link comes with a natural-language justification.
- **Deliver Human-Friendly Visualization All in One Place**
  - Interactive graphs for regulators, programmers, and statisticians.
- **Explore Data with Natural Language**

User Flow

Inputs

- Protocol (USDM / PDF)
- CRF (annotated CRF)
- SDTM / ADaM (Define.xml / spec.xlsx)
- TLF (RTF or ARS/ARD)

Processing

- Preprocessing
- AI model analysis
- Lineage image generation

Outputs

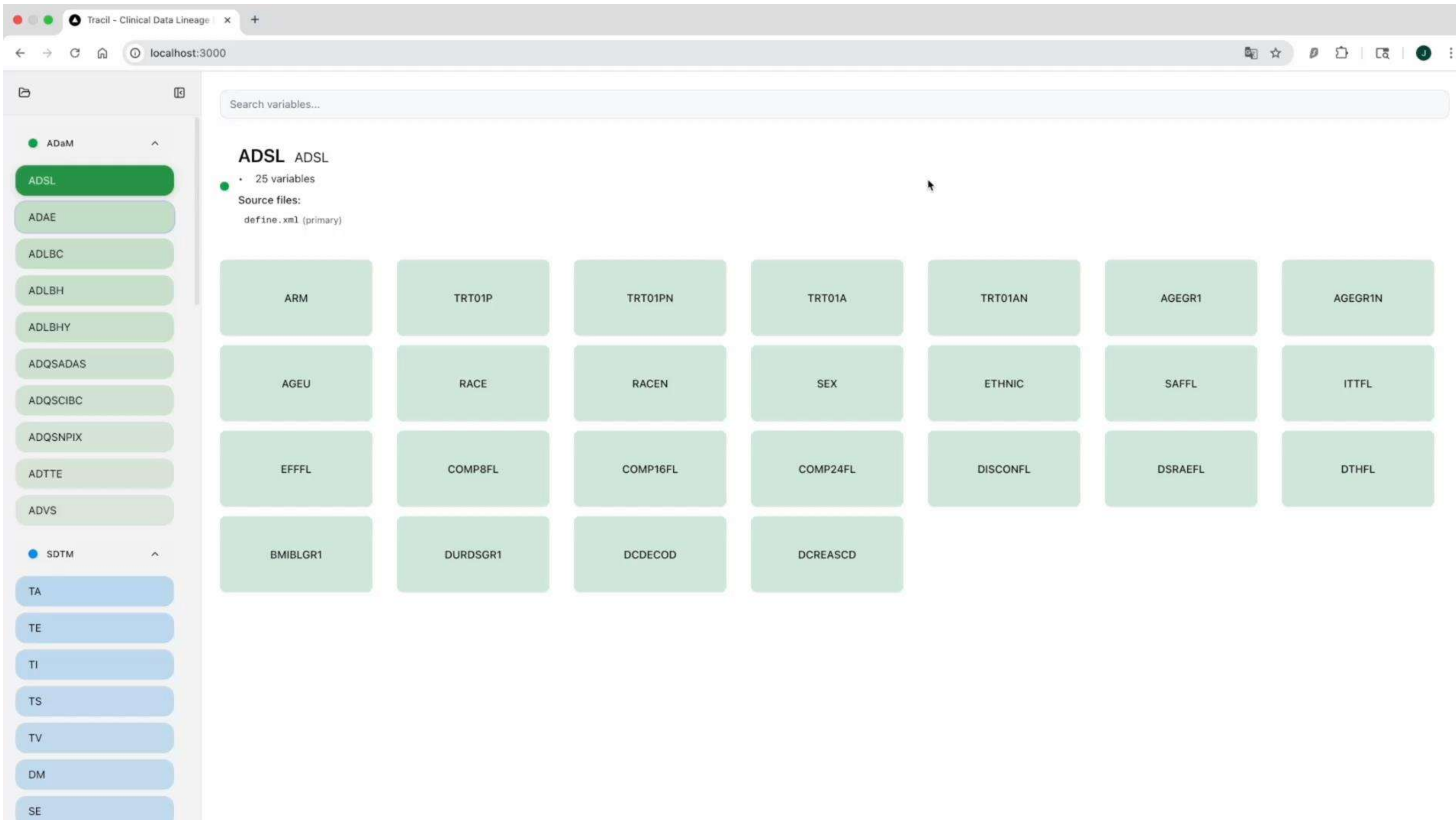
- Interactive Lineage Graph

Demo

Upload & General UI

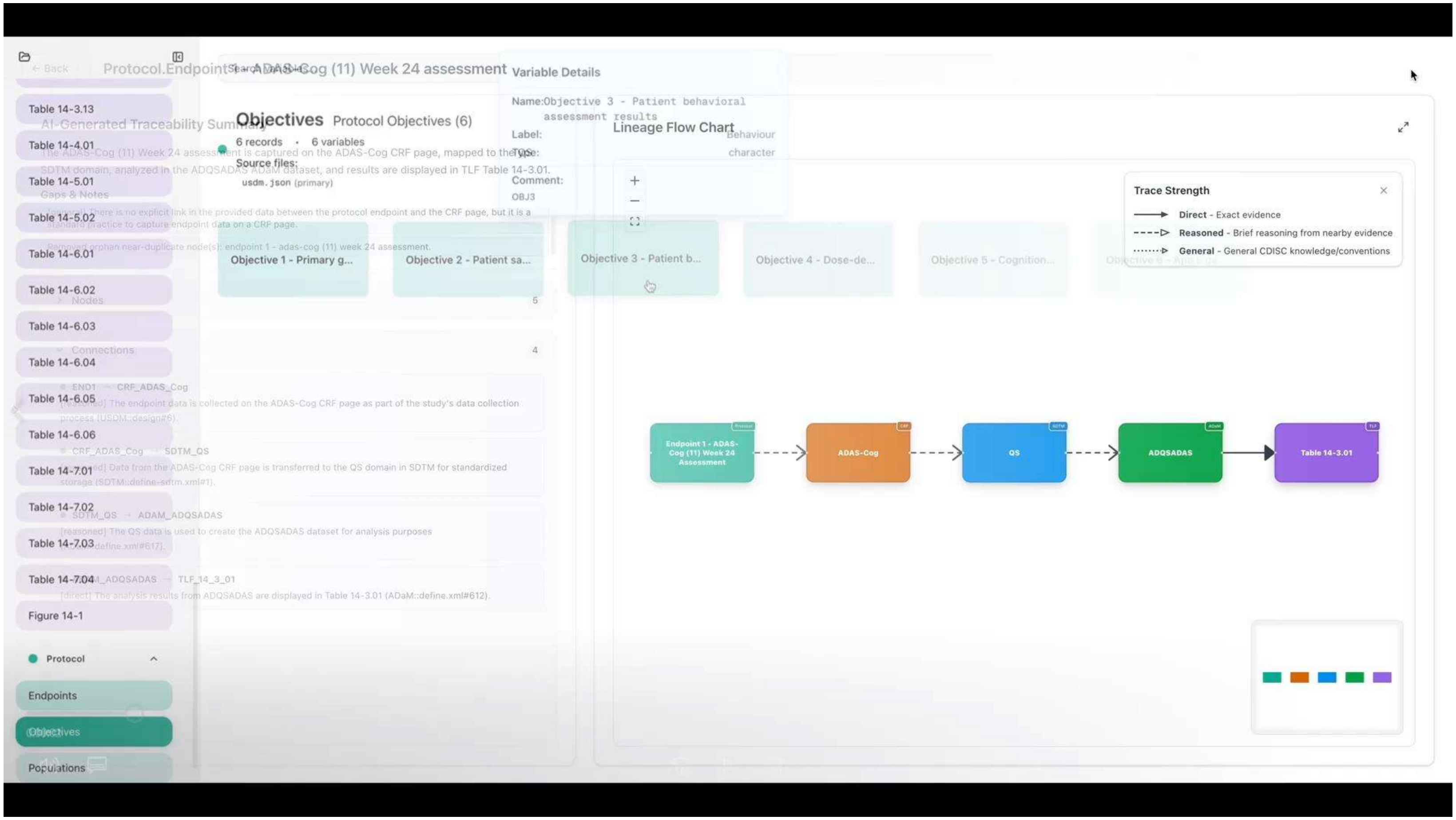
DEMOS

ADaM

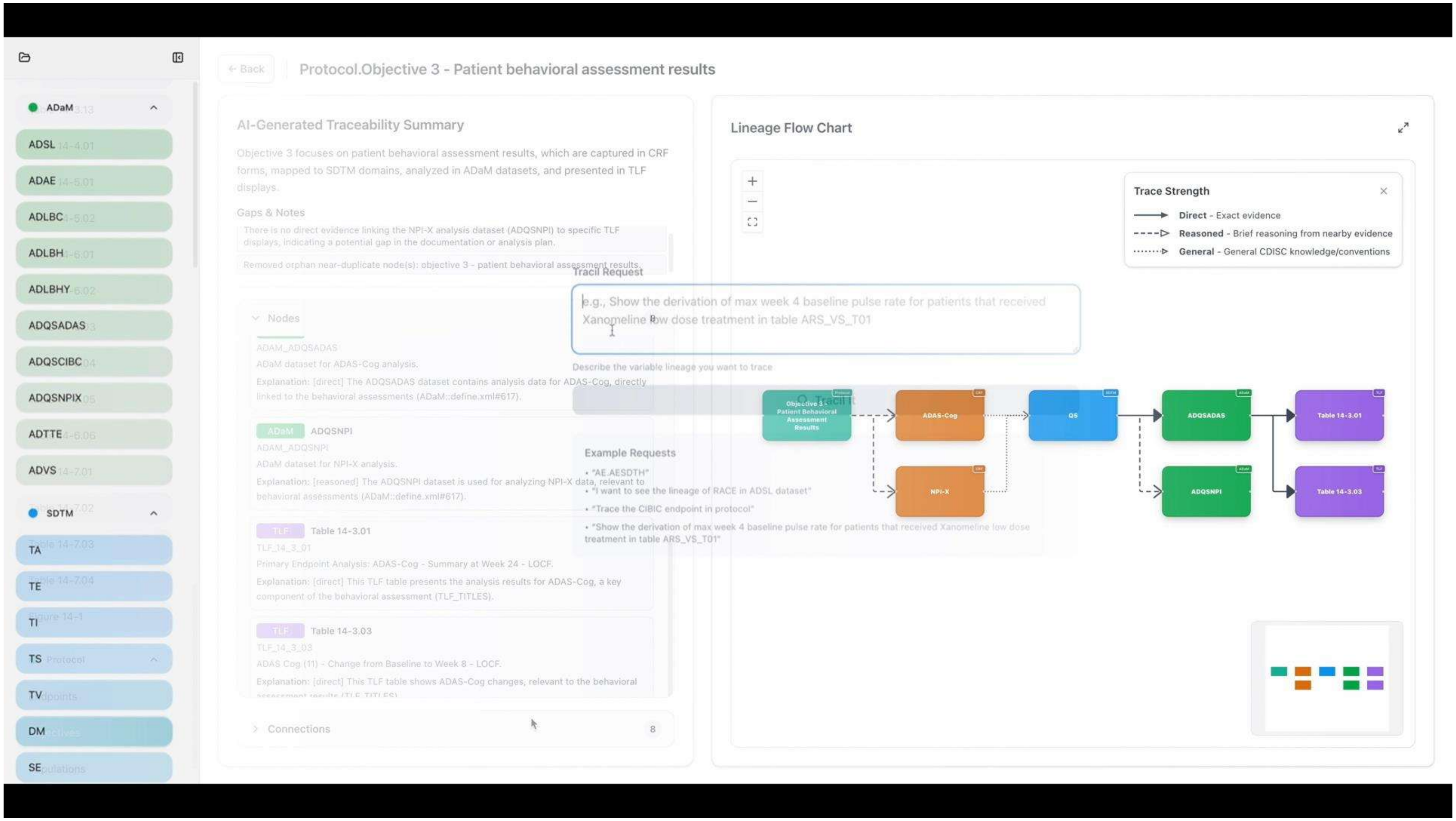


Tracil - Clinical Data Lineage
localhost:3000
Search variables...
ADaM
ADSL
ADAE
ADLBC
ADLBH
ADLBHY
ADQSADAS
ADQSCIBC
ADQSNPIX
ADTTE
ADVS
SDTM
TA
TE
TI
TS
TV
DM
SE
ADSL ADSL
25 variables
Source files:
define.xml (primary)
ARM
TRT01P
TRT01PN
TRT01A
TRT01AN
AGEGR1
AGEGR1N
AGEU
RACE
RACEN
SEX
ETHNIC
SAFFL
ITTFL
EFFFL
COMP8FL
COMP16FL
COMP24FL
DISCONFL
DSRAEFL
DTHFL
BMIBLGR1
DURDSGR1
DCDECOD
DCREASCD

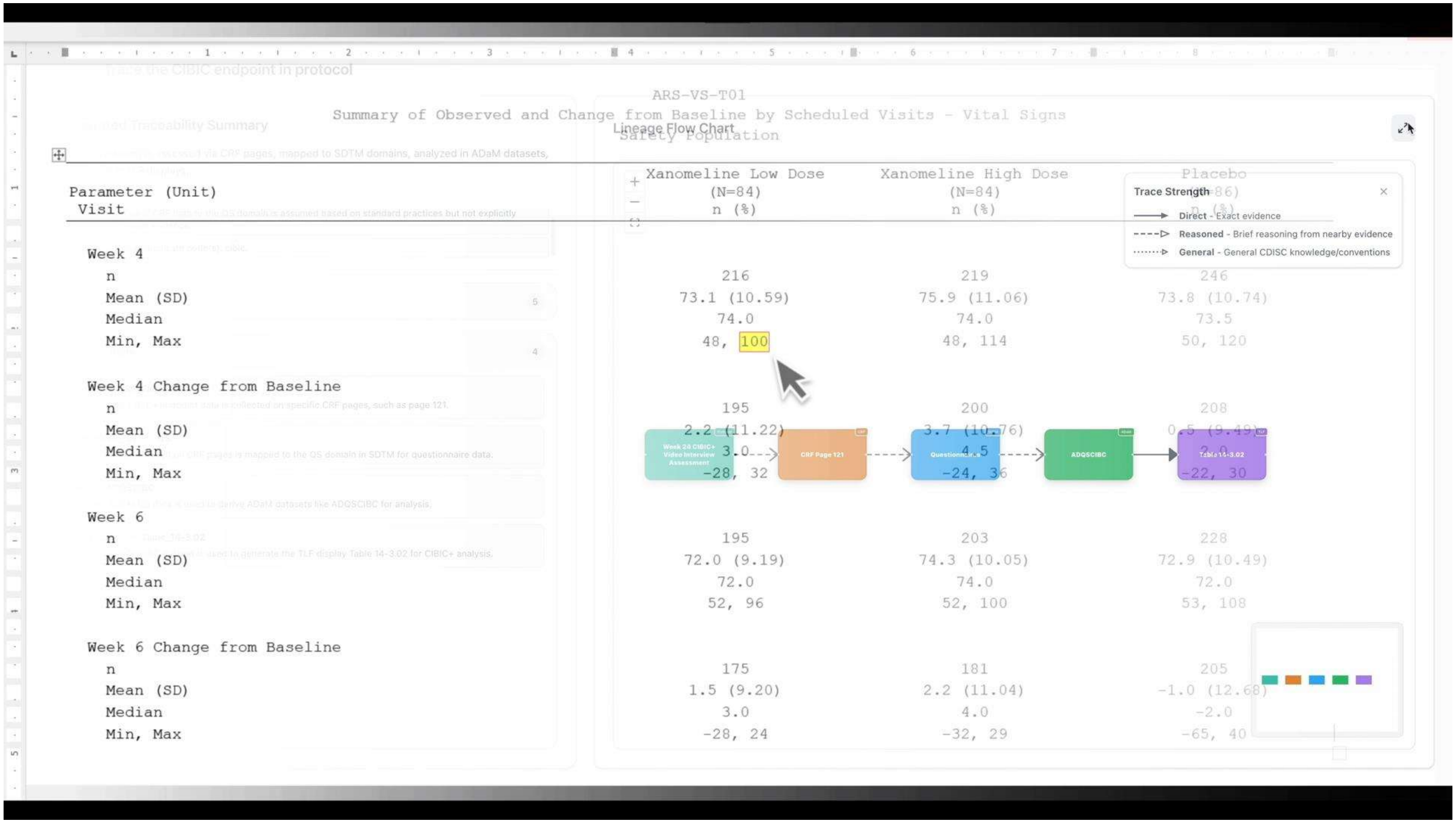
Objectives


Objectives Protocol Objectives (6)
6 records • 6 variables
Source files:
usdm.json (primary)
Variable Details
Name:Objective 3 - Patient behavioral assessment results
Lineage Flow Chart
Table 14-3.13
Table 14-4.01
Table 14-5.01
Table 14-5.02
Table 14-6.01
Table 14-6.02
Table 14-6.03
Table 14-6.04
Table 14-6.05
Table 14-6.06
Table 14-7.01
Table 14-7.02
Table 14-7.03
Table 14-7.04
Figure 14-1
Protocol
Endpoints
Objectives
Populations
Objective 1 - Primary g...
Objective 2 - Patient sa...
Objective 3 - Patient b...
Objective 4 - Dose-de...
Trace Strength
Direct - Exact evidence
Reasoned - Brief reasoning from nearby evidence
General - General CDISC knowledge/conventions
ADAS-Cog
QS
ADQSADAS
Table 14-3.01

Search With Natural Language



TLF Details with ARS



ARS-VS-T01
Summary of Observed and Change from Baseline by Scheduled Visits - Vital Signs
Safety Population

Parameter (Unit) Visit	Xanomeline Low Dose (N=84) n (%)	Xanomeline High Dose (N=84) n (%)	Placebo (N=86) n (%)
Week 4			
n	216	219	246
Mean (SD)	73.1 (10.59)	75.9 (11.06)	73.8 (10.74)
Median	74.0	74.0	73.5
Min, Max	48, 100	48, 114	50, 120
Week 4 Change from Baseline			
n	195	200	208
Mean (SD)	2.2 (11.22)	3.7 (10.76)	0.5 (9.49)
Median	3.0	4.5	2.0
Min, Max	-28, 32	-24, 36	-22, 30
Week 6			
n	195	203	228
Mean (SD)	72.0 (9.19)	74.3 (10.05)	72.9 (10.49)
Median	72.0	74.0	72.0
Min, Max	52, 96	52, 100	53, 108
Week 6 Change from Baseline			
n	175	181	205
Mean (SD)	1.5 (9.20)	2.2 (11.04)	-1.0 (12.68)
Median	3.0	4.0	-2.0
Min, Max	-28, 24	-32, 29	-65, 40

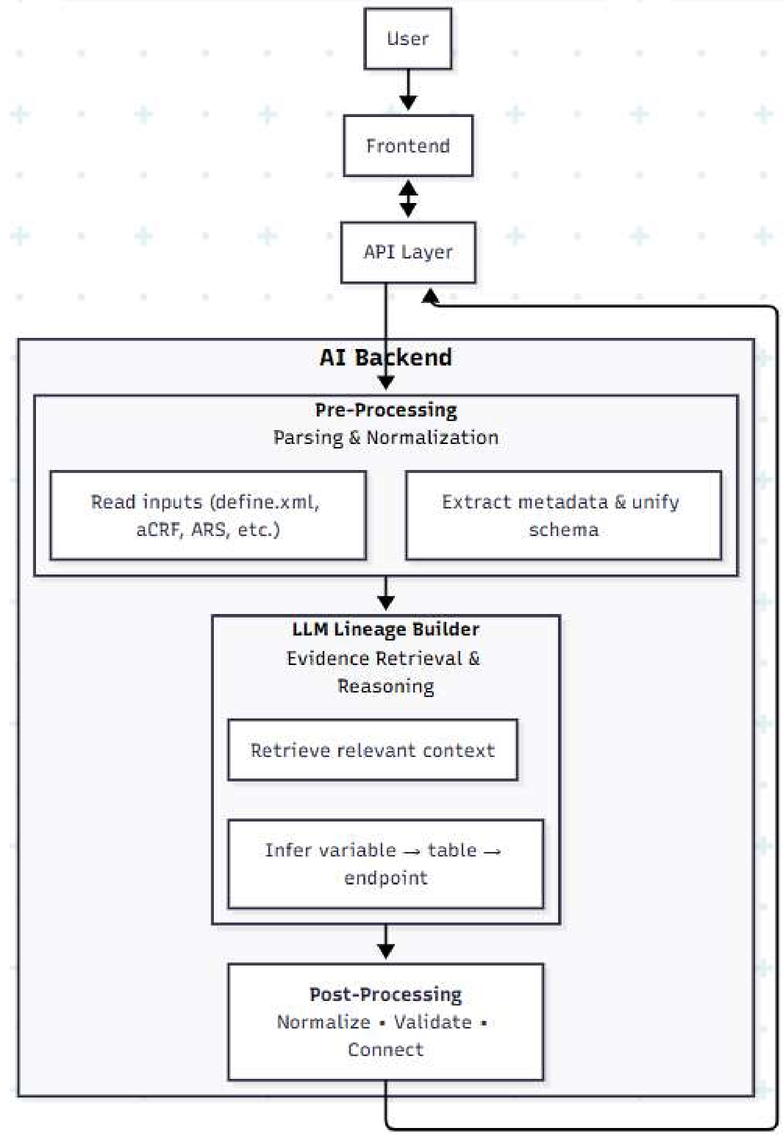
App Structure

App Structure

- Frontend
  - User Interface
  - React/Next.js (JavaScript Library)

- API
  - Bridge between UI and AI
  - FastAPI (Python Package)

- AI core: Python AI Engine
  - GPT-4o (mini)
  - Text-embedding-3-model from OpenAI



Backend AI Workflow

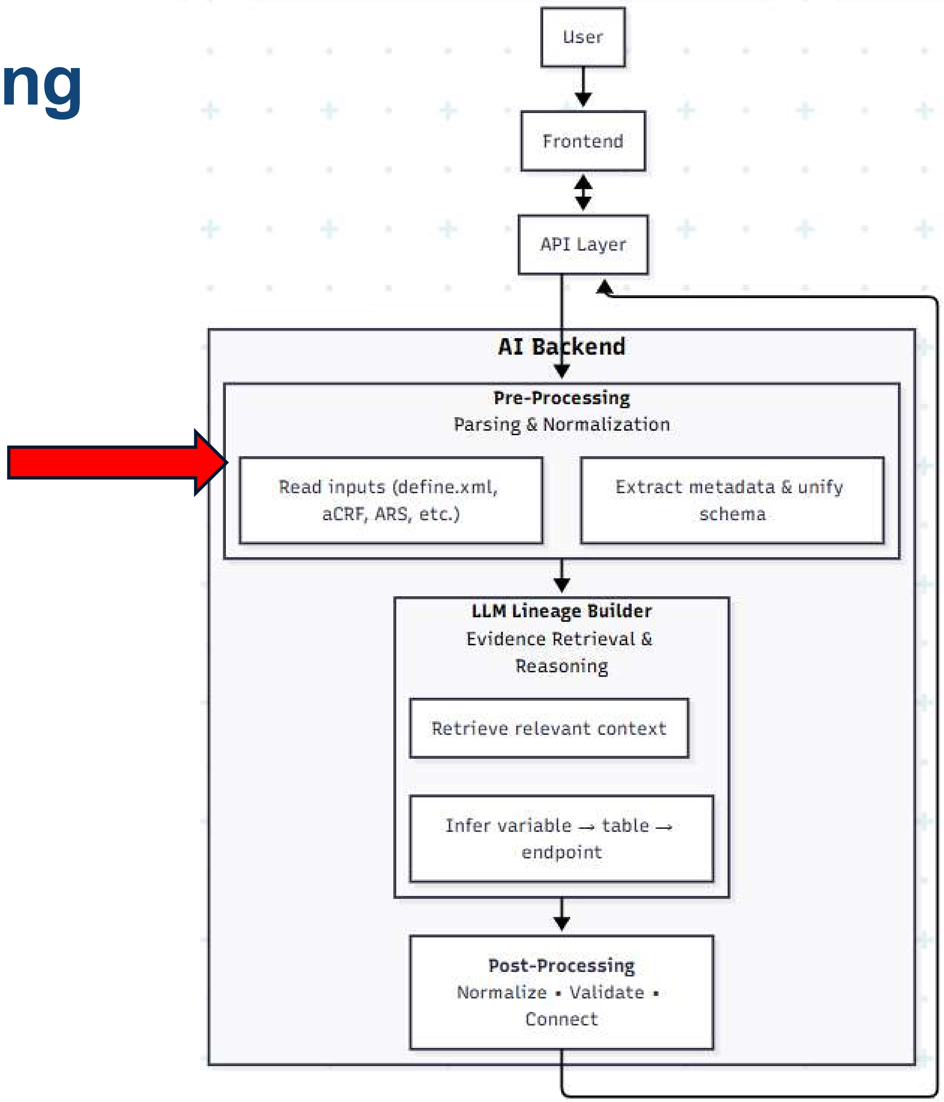
How Tracil's Backend Works

- **Pre-processing**: Python code to convert structured (XML, JSON) and semi-structured (PDF/CRF) inputs into a unified, machine-readable JSON file

- **LLM Reasoning:** passes the inputs to an LLM reasoning layer to infer variable derivations, dependencies, and gaps, returns a JSON graph

- **Post-processing:** standardize the JSON lineage graph so that it can be visualized interactively in the frontend UI

Pre-Processing



Pre-Processing: Parsing & Normalization

Input formats supported: Specification (.xlsx), define.xml, aCRF, ARS JSON, Protocol PDF, USDM, and TLFs (RTF)

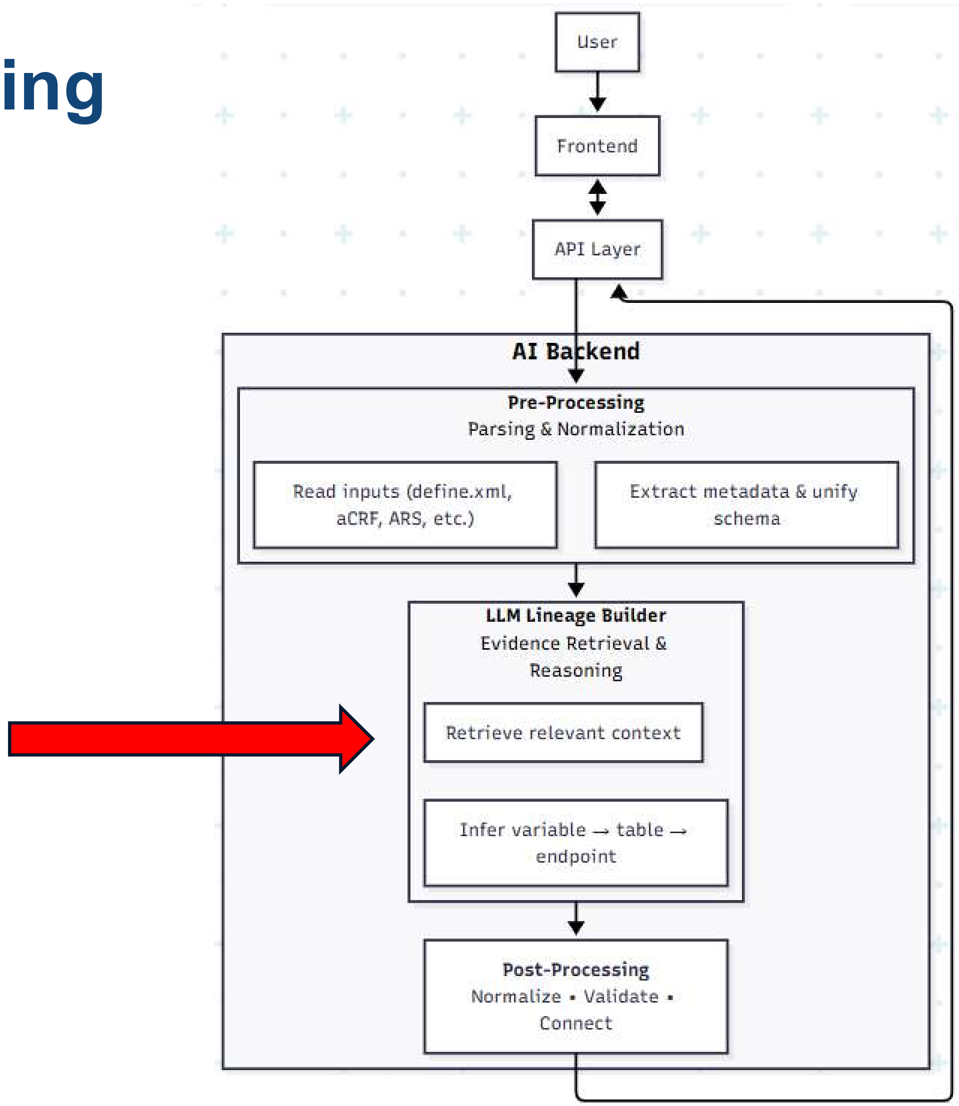
FastAPI endpoint `/process-files`:

- Extracts variable metadata, derivation notes, and dataset context
- Normalizes the information extracted into unified schema of names, domain tags, and relationships information

Why important:

- Provides the frontend with structured data for visualization and user interaction.
- gives LLM a clean, consistent input so it can reason semantically

LLM Reasoning



LLM Reasoning (Part 1): Lineage Builders & Evidence Retrieval

Three Routes by Target

- Detects what you're tracing: 1. protocol related 2. ADaM/SDTM variable 3. table/cell; sends it to the right builder

Gather the Evidence

- Collects all supporting pre-processed metadata from the session (aCRF index, protocol text, unified JSON returned by /process-files API, etc.)
- These documents form the "evidence base" for reasoning

Find What Matters (Chunk + Retrieve)

- Splits large documents into small readable sections
- Converts each into numerical "embeddings" so the AI can compare meanings
- Selects only the Top K (≈12) most relevant pieces to focus the analysis

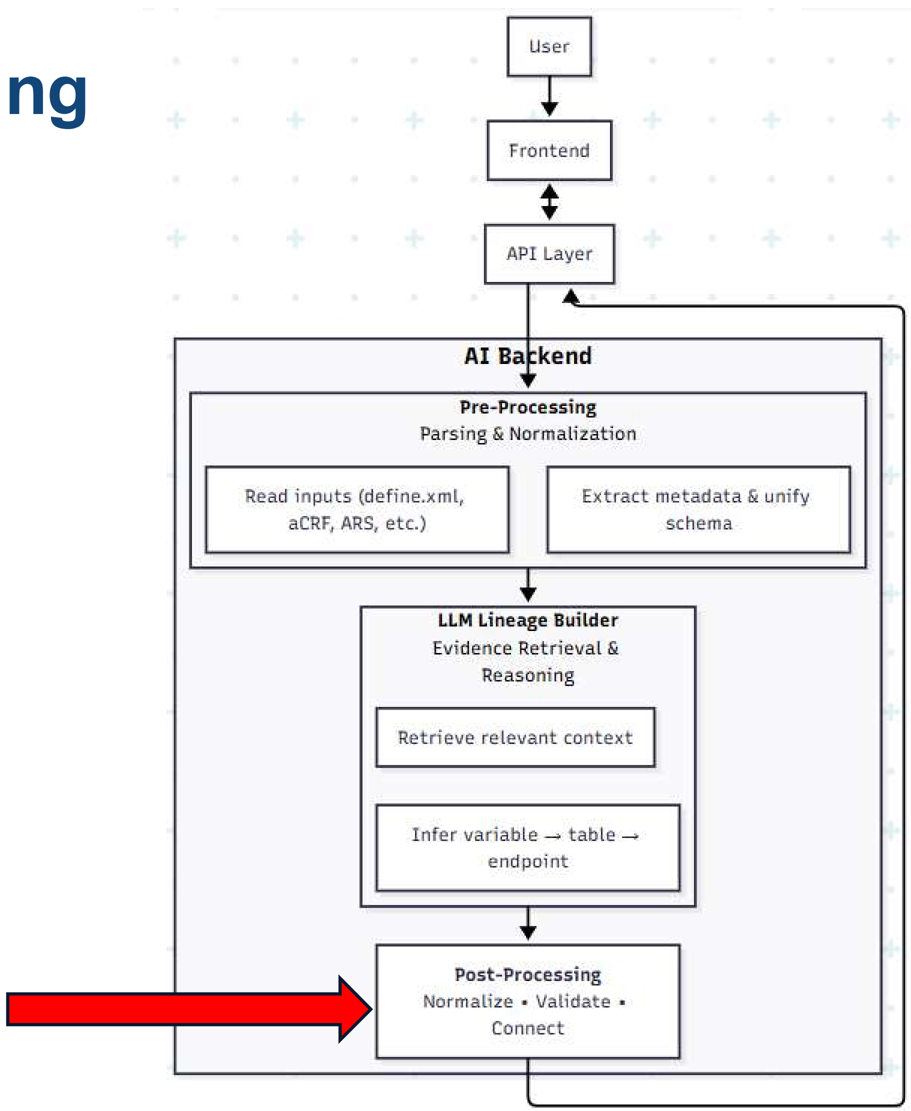
Ask the AI Model

- Sends the selected context to GPT-4o for reasoning (mini model as fallback).
- The model returns a structured JSON lineage, showing each variable, link, and explanation.

LLM Reasoning (Part 2): Prompt Design

- **Clear Role Definition:**
  - The AI is given the role of a *"Senior CDISC standards expert"* — it follows clear forward and backward tracing rules and uses standardized variable names (like **ADSL.AGE** or **DM.BRTHDTC**) to keep everything consistent.
- **Structured Thinking:**
  - Each task follows a fixed schema (variable, endpoint, or table) so the AI always knows what format to produce
- **Evidence packing:**
  - The AI reviews only the top relevant document sections tagged as evidence before reasoning
- **Explainable Results:**
  - Every link in the lineage includes a short explanation starting with [direct], [reasoned], or [general], and cites where the information came from
- **Direction rules:**
  - hard requirement to emit edges upstream → downstream only; forbid illegal shortcuts (e.g., ADaM → CRF)
- **Resilience:**
  - Chat gpt-4o API called with temperature = 0 and response format as JSON object. Did not use gpt-5 because JSON not supported. Built-in checks handle small formatting issues automatically

Post-Processing



Post-Processing: Normalize, Validate, Connect

Normalize the Structure

- Add tags and standardize names (e.g., `ADVS.AVAL → ADaM variable`, `VS.VSORRES → SDTM variable`)
- Fix missing labels and edge directions for clear flow (`source → result`)

Validate the Connections

- Check each link has real evidence; flag gaps if missing
- Remove duplicates/orphans, and add short explanations

Connect Missing Pieces

- Re-query files when links are incomplete (e.g., ADaM variable with no SDTM parent)
- Ensure full trace: Protocol → CRF → SDTM → ADaM → TLF

Conclusion & Future Steps

Key Takeaways

- Tracil automates lineage across Protocol → CRF → SDTM → ADaM → TLF using AI reasoning
- Converts protocol endpoints/objectives/SOA, aCRF, define.xml, specifications, ARS into a **unified JSON schema**
- Provides **explainable AI outputs** with clear variable relationships
- Makes traceability **easy, fast, interactive, and follows CDISC standards.**

Limitations

- **Model Accuracy & Stability:** Same input can yield slightly different results due to the nature of LLM. Addressed this challenge by:
  - pre and post processing
  - prompt engineering
  - setting model parameter to the most deterministic mode
- **Limited Data:** Few open, CDISC-compliant datasets restrict realistic fine-tuning
- **Validation Gap:** Need standardized validation methods to ensure reliable outputs

Possible Improvement

- **Model Expansion:** Test across different LLMs (GPT latest model, Gemini, Claude, etc.); Test on Merck GPTeal and Internal studies
- **Confidence Scoring + User Feedback:** Quantify AI certainty and learn from human corrections

Lessons Learned

Using LLMs Effectively

- Gained experience in how to apply large language models for certain tasks and integrate AI into existing workflows

Setting Up Clear Rules

- *Prompt engineering:* define structured prompts and validation steps to guide the model
- *Iterative prompting:* test, observe, and adjust the prompt to improve the result

Balancing AI and Determinism

- Combine deterministic logic with LLM reasoning for efficiency and accuracy
- Add pre- and post-processing to reduce token use and improve consistency
- Some tasks remain better handled by deterministic scripts

Human Involvement is Always Important!

Thank You!

Acknowledgement

We would like to express our sincere gratitude to everyone who provided guidance, support, and valuable advice throughout this project:

Teammate: Anthony Chow

Managers: Xiaohui Wang, Lily Zhang, Huei-Ling Chen, Jing Su, and Amy Gillespie

Colleagues: David Izard, Jeff Cheng, and Rama Kudaravalli

Contact Information

- Junze Zhang
  - Junze.zhang@merck.com
- Kexin Guan
  - Kexin.guan@merck.com
- Anthony Chow
  - Achow@cdisc.org

- Tracil GitHub Repository:
  - https://github.com/1mgroot/Tracil