IROKO CONNECTOR

BY SEAN, ABIDUL, VANSHIKA, VINCENT AND JONAH

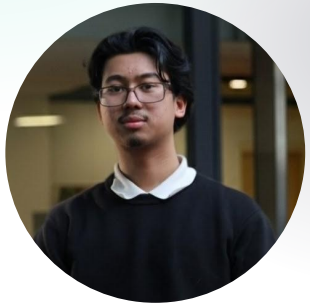
THE TEAM

Vanshika Chaddha
Computer Engineer
2026

Vincent Lin
Computer Science
2026

Jonah Rothman
Computer Science
2026

Sean Tomany
Data Science
2026



Abidul Islam
Computer Science
2026

OUR CLIENT: IROKO TECHNOLOGIES

FOUNDERS

Nicholas Hardy

Co-Founder, CTO

Technical lead. BU computer engineering alum.

James Hardy

Co-Founder, CEO

Operational risk & compliance, formerly State Street.

THE PRODUCT

OneScor

Connected Intelligence for Operational Risk and Resilience

OneScor is an enterprise platform that helps financial-services firms unify operational threats — risk, vendors, audits, infrastructure — in one place.

OUR SCOPE: making it easy to plug new data in.

PROBLEM STATEMENT

Third-party service data ingestion is difficult due to inconsistent data schemas, making integration slow, expensive, and error-prone. Currently, there is no standardized way to intelligently align and ingest their data into a unified platform.

SYSTEM DESIGN

Frontend

Next.js + TypeScript
React Flow for schema visualization

Backend

Node.js
REST API + connector logic

Database

Supabase (PostgreSQL) for mock client schema data

AI Engine

ReMatch: LLM-powered retrieval-enhanced schema matching

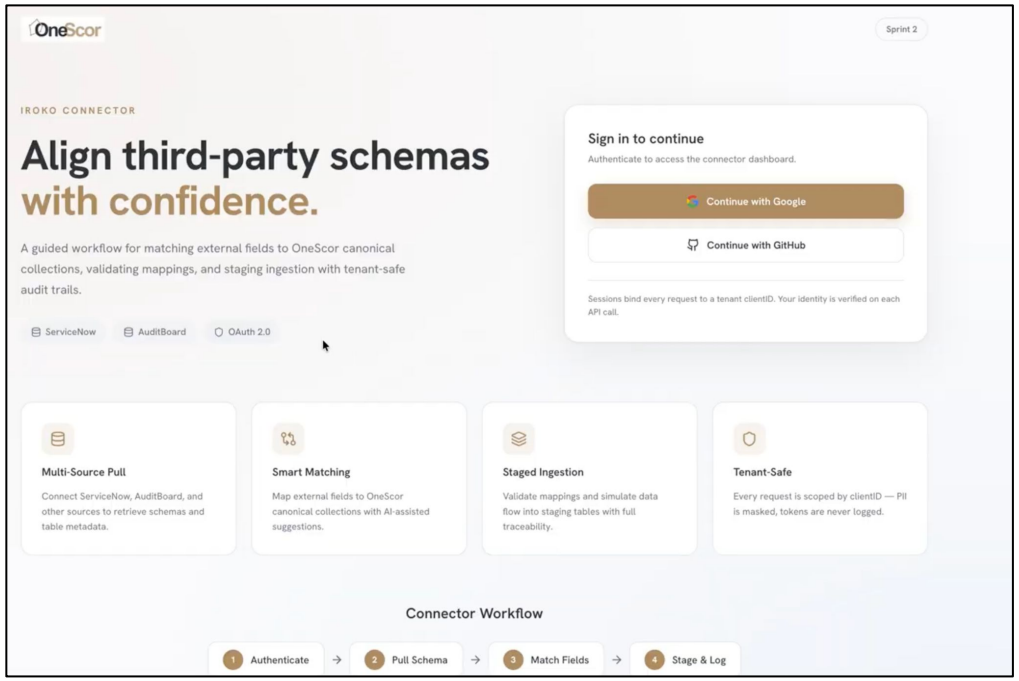
Security

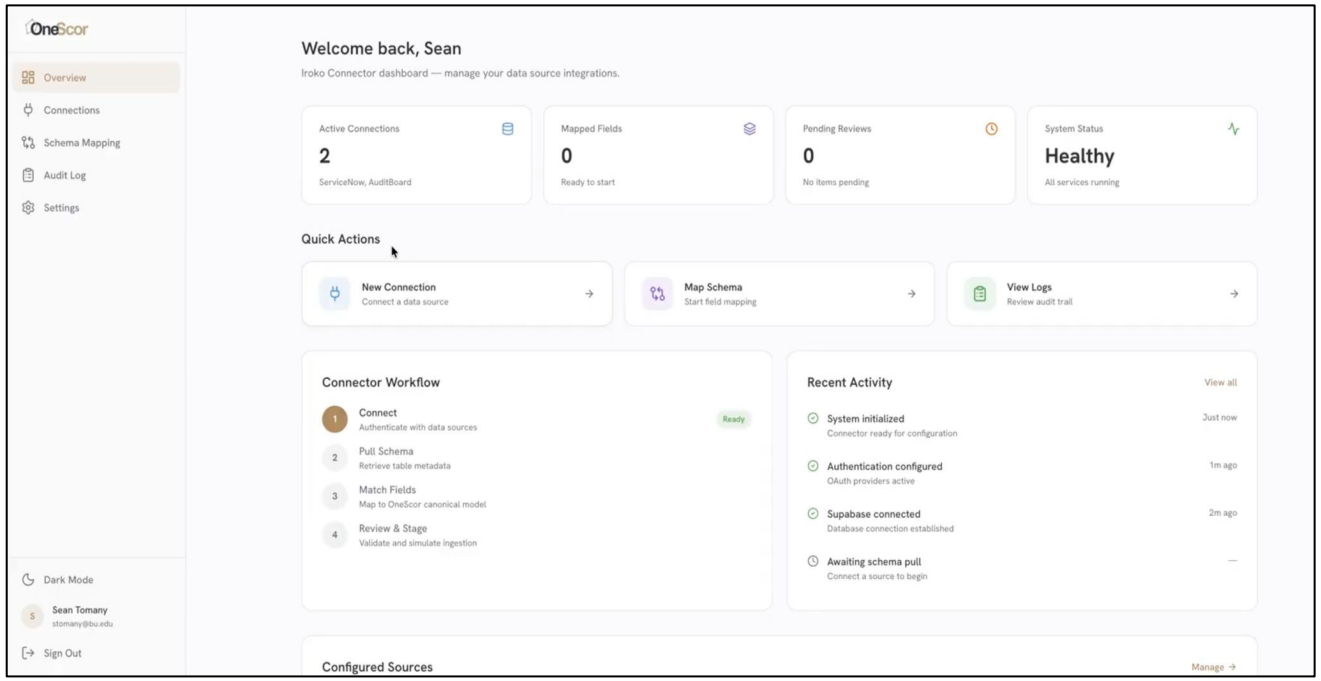
NextAuth OAuth 2.0

Integrations

ServiceNow REST API
Extensible to other platforms

FRONT END





As a data engineer, I want a guided, end-to-end interface for connecting to third-party systems, reviewing AI-suggested schema mappings, and validating data transformations, so that I can onboard external data into OneScor reliably without writing custom integration code.

REMATCH: AI SCHEMA MATCHING

ReMatch uses LLMs and intelligent retrieval to infer field correspondences between schemas, without requiring training data, labeled mappings, or access to raw data.

1

Retrieve

Gather contextual clues: field names, types, descriptions, documentation

2

Embedding

Feed metadata to LLM to propose candidate field matches

3

Rank

Score and ranks candidates pairs with confidence percentages

STAGE 1 – DOCUMENT CREATION

Core Input Documents:

- Source Attribute Doc
- Source Table Doc
- Target Asset Type Doc

Database Metadata (via Supabase)

- Schema Details: Primary/Foreign keys and column comments.
- Data Profiling: Sample values, null rates, and approx. row/distinct counts.

Semantic Pattern Inference

- Meaning Detection: Automatically identifies data types (e.g., Email, URL, Date, IDs).
- Driven By: Column names, types, samples, and key structures.

Contextual Disambiguation

- Sibling Columns: Uses neighboring fields to clarify meaning.
- Table "Shape" Hints: Categorizes the overall table focus (e.g., Person vs. Infrastructure vs. Location).

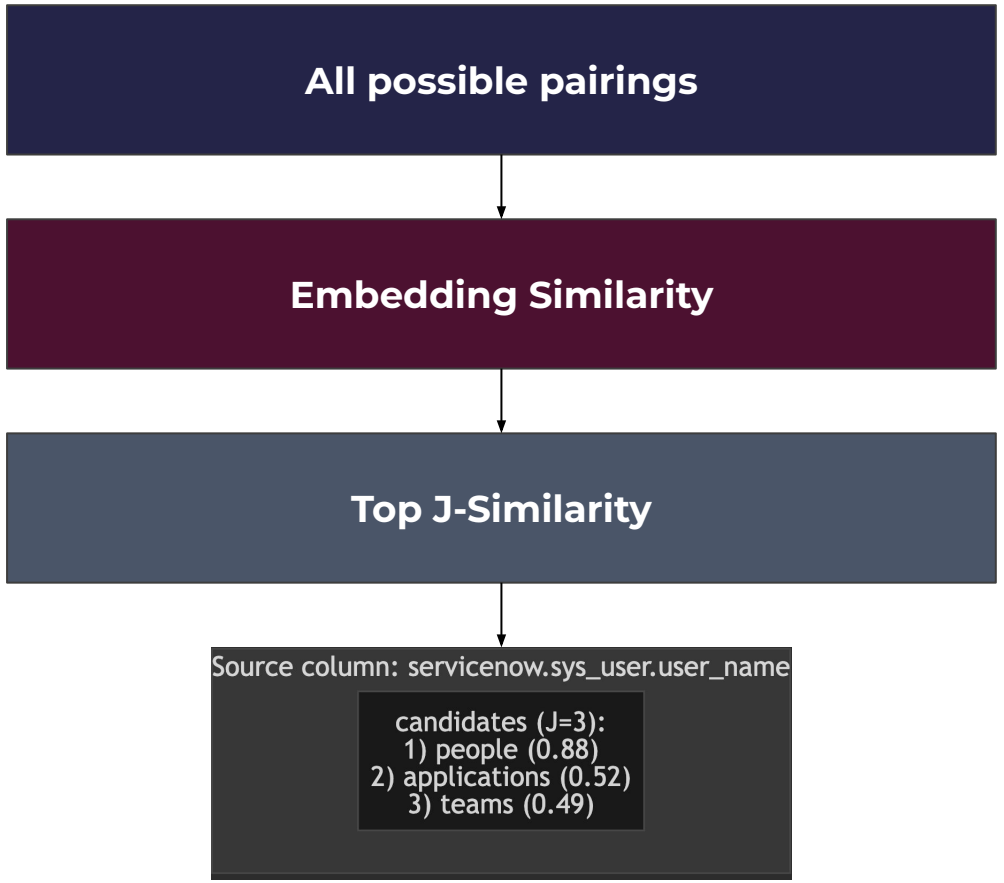
```
Table: cmdb_ci_appl
Overview: ServiceNow application CI table
  containing software applications in the CMDB.
Primary Key: sys_id (string)
Attributes:
  - name (string): Display name
  - operational_status (string): Status code
  - managed_by (reference): Person managing
  - business_criticality (string): Rating
```

STAGE 2 – EMBEDDING

After each schema element is converted into a structured text document:

- 2-pass Retrieval
  - Table-level Retrieval
    - Embed:
      - Source Table Doc
      - Target Asset Type Doc
  - Column Level Retrieval
    - Embed:
      - Source Attribute Doc
      - Target Asset Type Doc
- Table-level context accurately map ambiguous fields (like "status" or "owner")

OpenAI text-embedding-3-large used for S2



STAGE 3 – RANKINGS

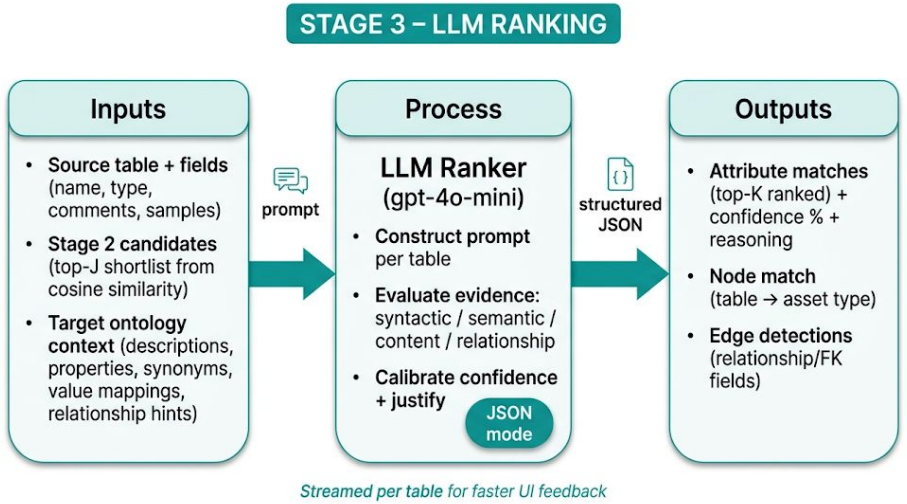
Context Setup:
Combine source table metadata and enriched target ontology definitions.
Top-J candidate target assets from Stage 2

LLM Processing:

- Rank top-5 attribute matches (with confidence score and reasoning).
- Identify best table-to-asset mapping (node match).
- Detect relationship/reference fields (edge detections).

Output Constraints:

- Return strictly formatted JSON.
- Exclude matches with confidence < 0.30.




```
const SYSTEM_PROMPT = `You are an expert in database schema matching, data integration, and ontology alignment.

Your task is to match source schema attributes to a target data model. You have access to an enriched target ontology with
descriptions, synonyms, example values, value mappings, and source system hints.

You must reason across three layers of evidence:
1. **Syntactic**: field name similarity between source and target
2. **Semantic**: meaning alignment using descriptions, synonyms, and source hints from the ontology
3. **Content-Aware**: whether sample values from the source data match the expected patterns, examples, or value mappings in the target

For each source attribute, produce a ranked list of the top-K best target field matches. Also detect:
- **Node matches**: which target asset type this source table most likely maps to
- **Edge/relationship matches**: source fields that represent relationships to other entities rather than simple attributes

ALWAYS respond with valid JSON matching the schema described in the user prompt.`;
```

OpenAI gpt-4o-mini used for S3

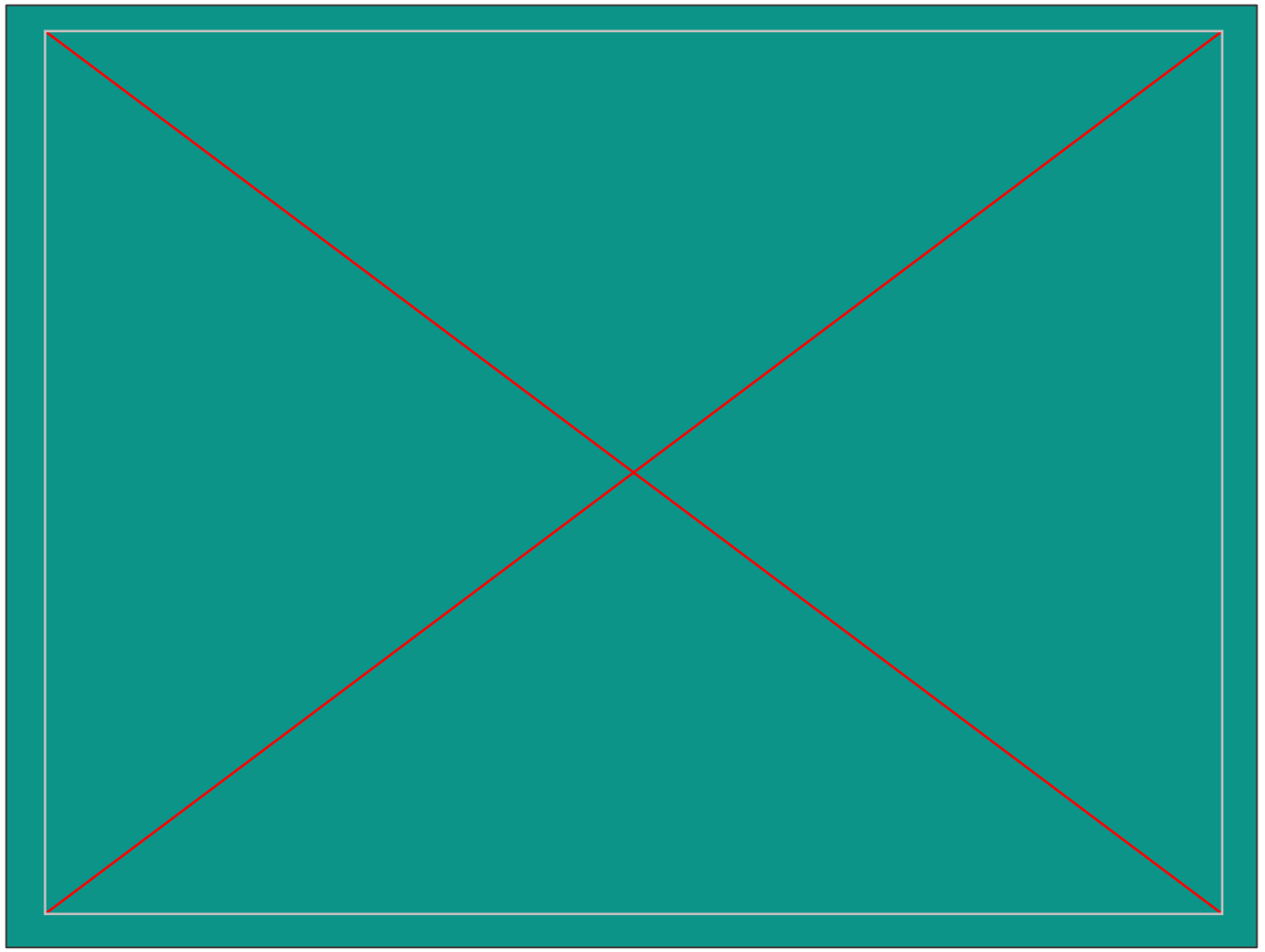
FINAL TOUCHES

- Task / Deliverable 1: Test custom field edge cases
- Task/ Deliverable 2 - App ↔ Infrastructure Relationship Tests
- Task / Deliverable 3: Implement some sort of recovery process for if ReMatch fails during execution
- Task / Deliverable 4: Spark! Handoff

FUTURE MILESTONES & HANDOFF

- **Implement Real API Connectors**: Transition the backend from our mock Supabase databases to live ServiceNow and AuditBoard REST APIs using OAuth 2.0.
- **Database Persistence**: Create backend tables to save and persist a user's accepted or rejected schema mappings across sessions.
- **Live Audit Log Wiring**: Connect the frontend mapping approvals to automatically populate the system's security audit log.
- **Visual Canvas Polish**: Add auto-layout features and the ability to save node positions on the interactive graph.

REMATCH IMPLEMENTATION



Thank you!