

NicheExplorer

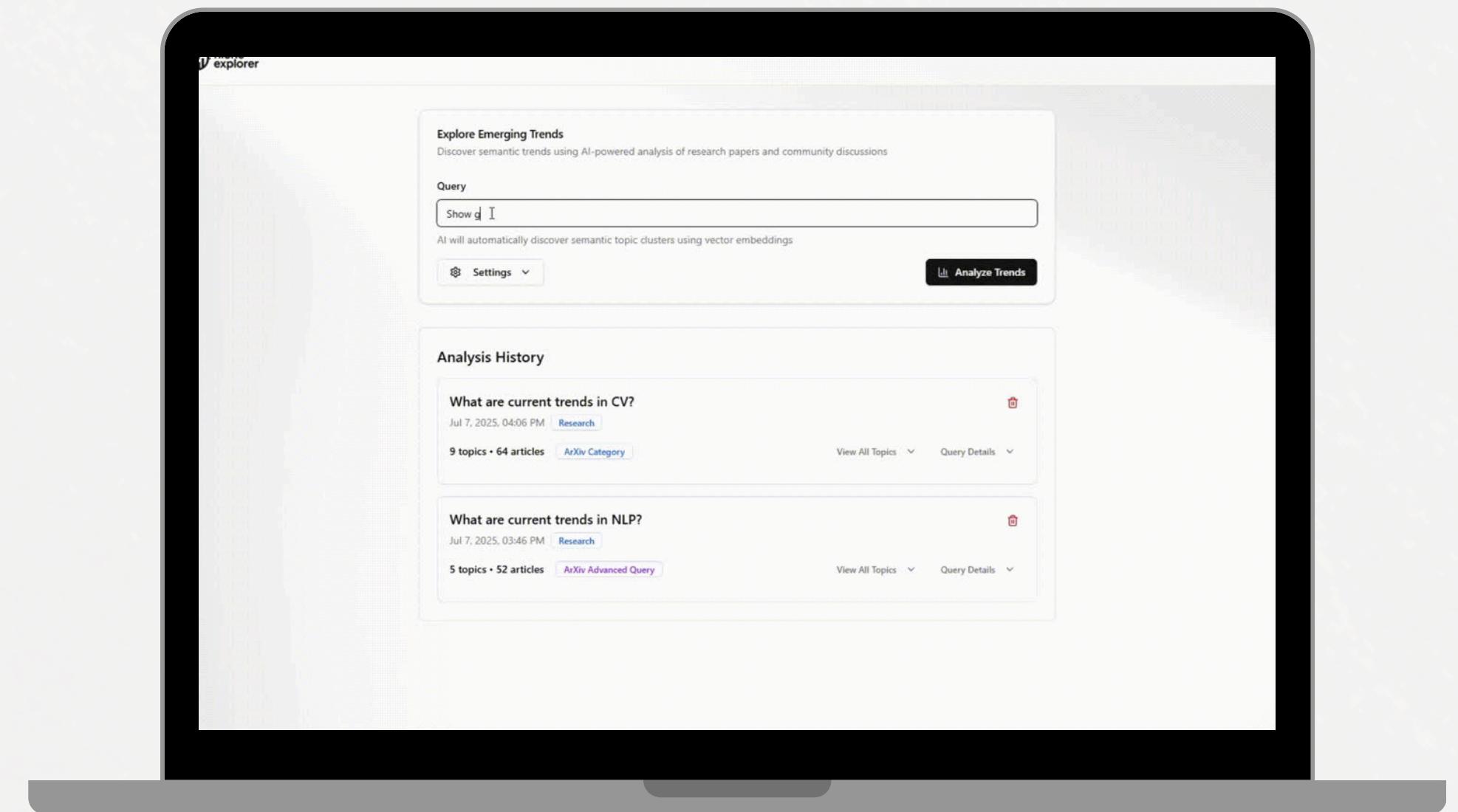
Niche Explorer - Janik Jehkul, Manuel Wilhelm

NicheExplorer

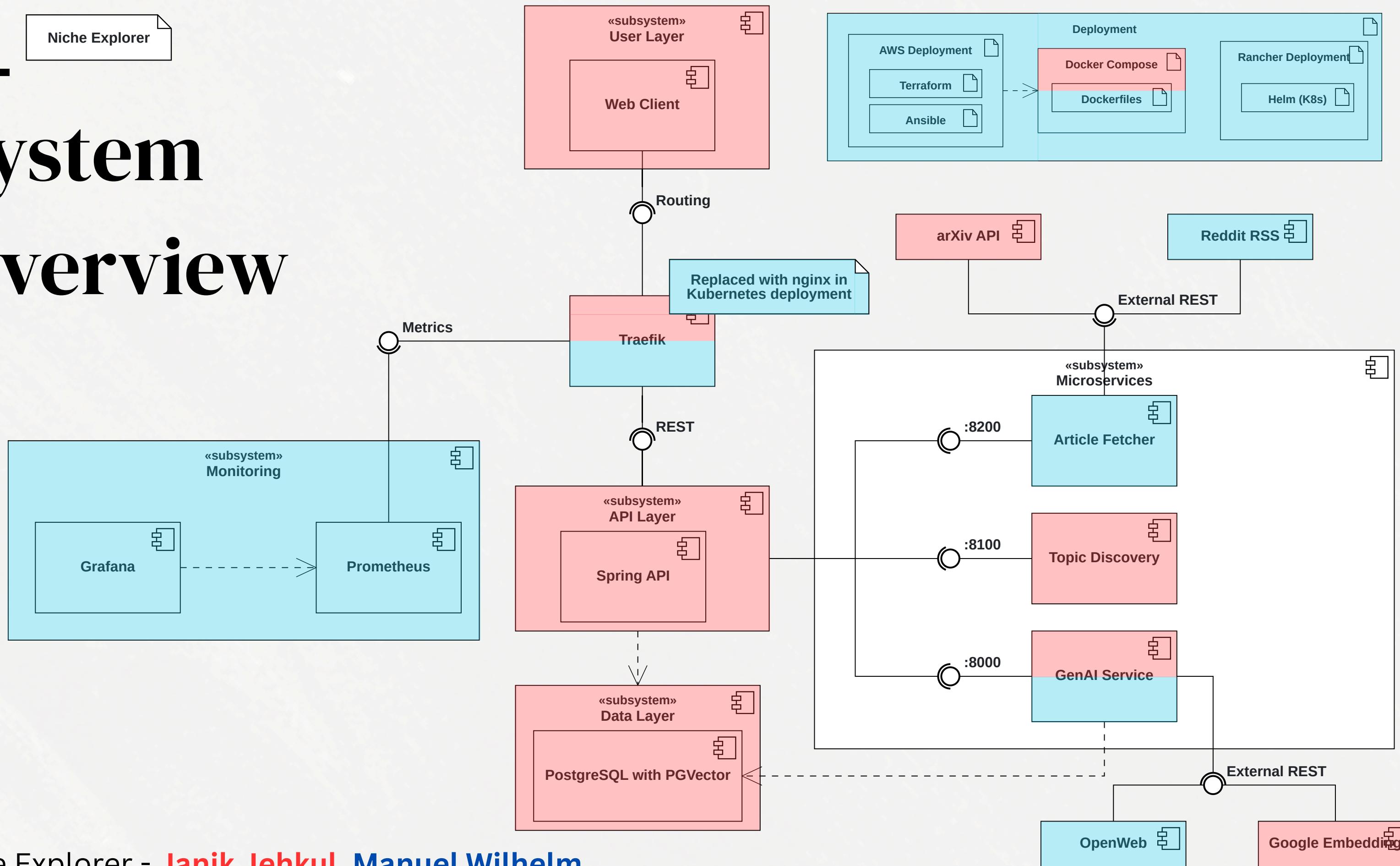
Problem: Great ideas are buried under a mountain of data

NicheExplorer:

- **Aggregates documents** from arXiv and reddit
- **Identifies underlying trends** using topic modeling
- **Presents the discovered topics** in an interactive list, allowing users to explore the topics

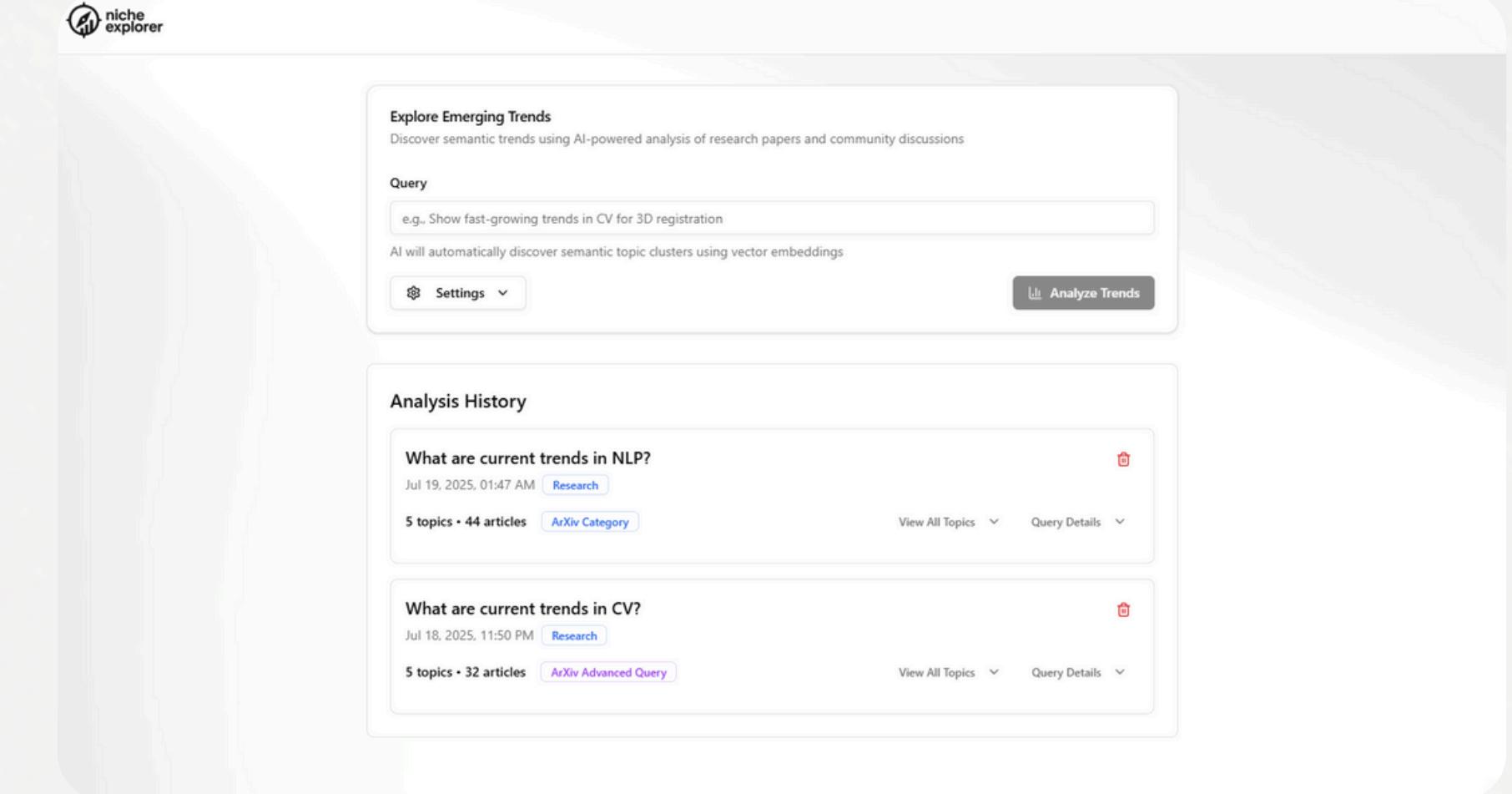


System Overview



Client - Frontend

- **Core stack:** React, Vite & TypeScript
- **UI Framework:** Tailwind CSS paired with shadcn components
- **Type-Safe API Contract:** Backend communication is handled by a strictly-typed API client auto-generated OpenAPI specification



The screenshot shows the Niche Explorer web application. At the top, there's a navigation bar with the 'niche explorer' logo. Below it, a large search bar is labeled 'Explore Emerging Trends' with the sub-instruction 'Discover semantic trends using AI-powered analysis of research papers and community discussions'. A placeholder text 'e.g., Show fast-growing trends in CV for 3D registration' is present, along with a 'Settings' dropdown and a 'Analyze Trends' button. Below the search bar, the 'Analysis History' section displays two previous queries:

- What are current trends in NLP?** (Jul 19, 2025, 01:47 AM) - Research, 5 topics • 44 articles, ArXiv Category. Includes 'View All Topics' and 'Query Details' buttons.
- What are current trends in CV?** (Jul 18, 2025, 11:50 PM) - Research, 5 topics • 32 articles, ArXiv Advanced Query. Includes 'View All Topics' and 'Query Details' buttons.

Java

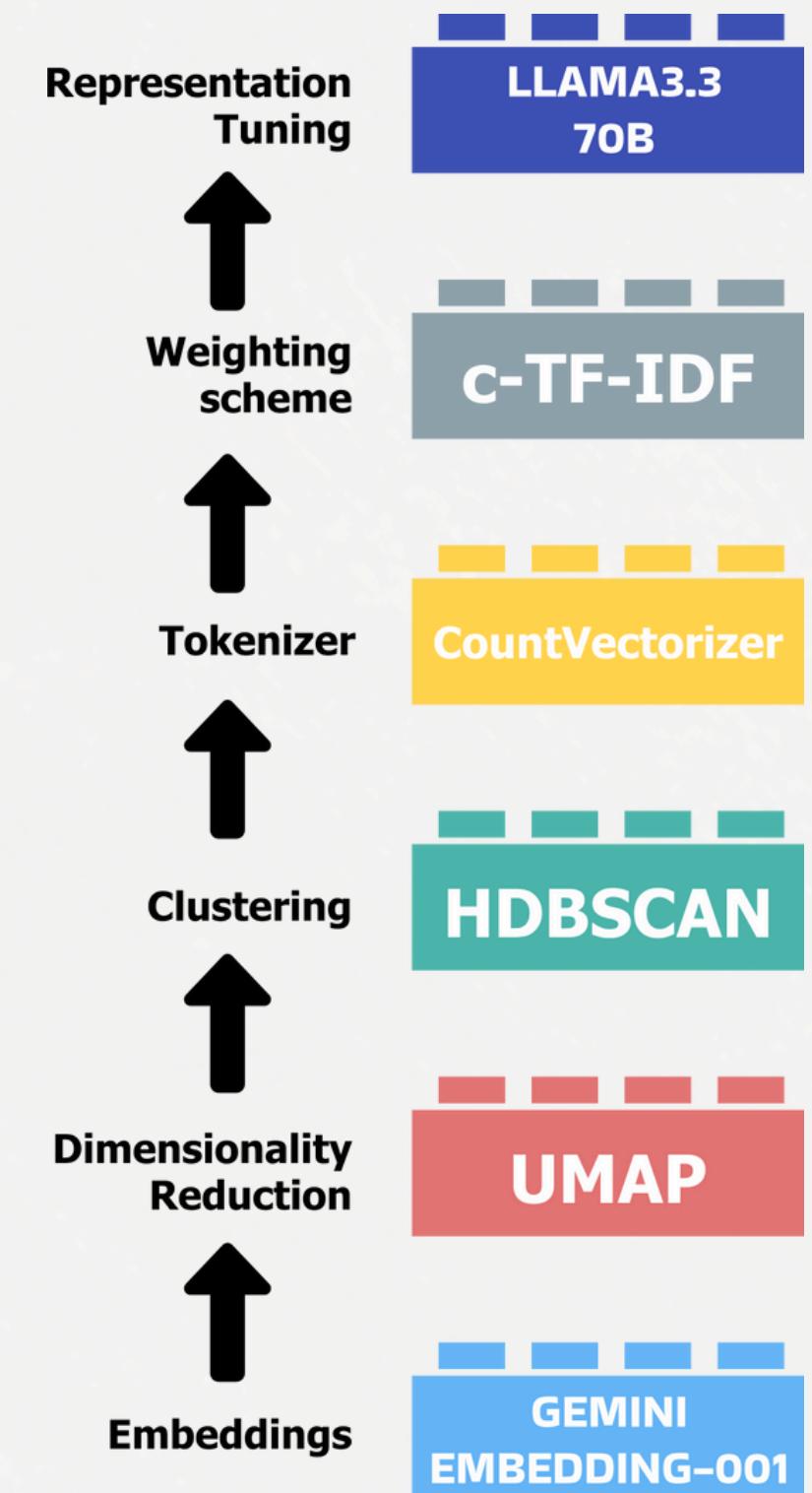
- **Core stack:** Spring Boot & Java, built with Gradle
- **Layered Architecture:**
 - **Controller:** validate & route HTTP requests/responses
 - **Services:**
 - **AnalysisService:** DB persistence (PostgreSQL + PGVector)
 - **AnalysisOrchestrationService:** microservice orchestration
 - **Config:** centralize API clients
- **Type-Safe API Contract:** Clients auto-generated from OpenAPI spec

Python - Article Fetcher & Gen AI

- **Article Fetcher:** Python, FastAPI
 - Fetching ArXiv articles from the ArXiv API
 - Fetching Reddit posts from the Reddit RSS feed
- **Generative AI:**
 - **Text Generation:** Turns a prompt into structure text using OpenWebUI or Gemini
 - **Embedding:** Generates and fetches Gemini embeddings
 - **Classification:** Predicts arXiv or reddit category & generates query

Python - Topic Discovery

- **Core stack:** Python, FastAPI
- **ML-Pipeline:**
 - Get embeddings from PGVector
 - Cluster articles with BERTopic
 - Sub-clustering when topic is large (>10 docs)
 - Label/summarize each topic
- **Output:** Ranked list of topics with title, description relevance



Monitoring

Prometheus (Metrics, Alerts)

- **Core metrics:**
 - Service Availability
 - Number of Calls to External Providers
- **Core Alerts:**
 - Service Unavailable
 - Calls Approaching Provider Rate Limit
 - Calls Exceeding Provider Rate Limit

Grafana (Visualization)



Deployment

Local - Docker Compose

Rancher - K8s Deployment

Cloud - AWS Deployment

Docker Compose
Local Docker Compose Overwrite

Helm: Monitoring Deployment
Application Deployment

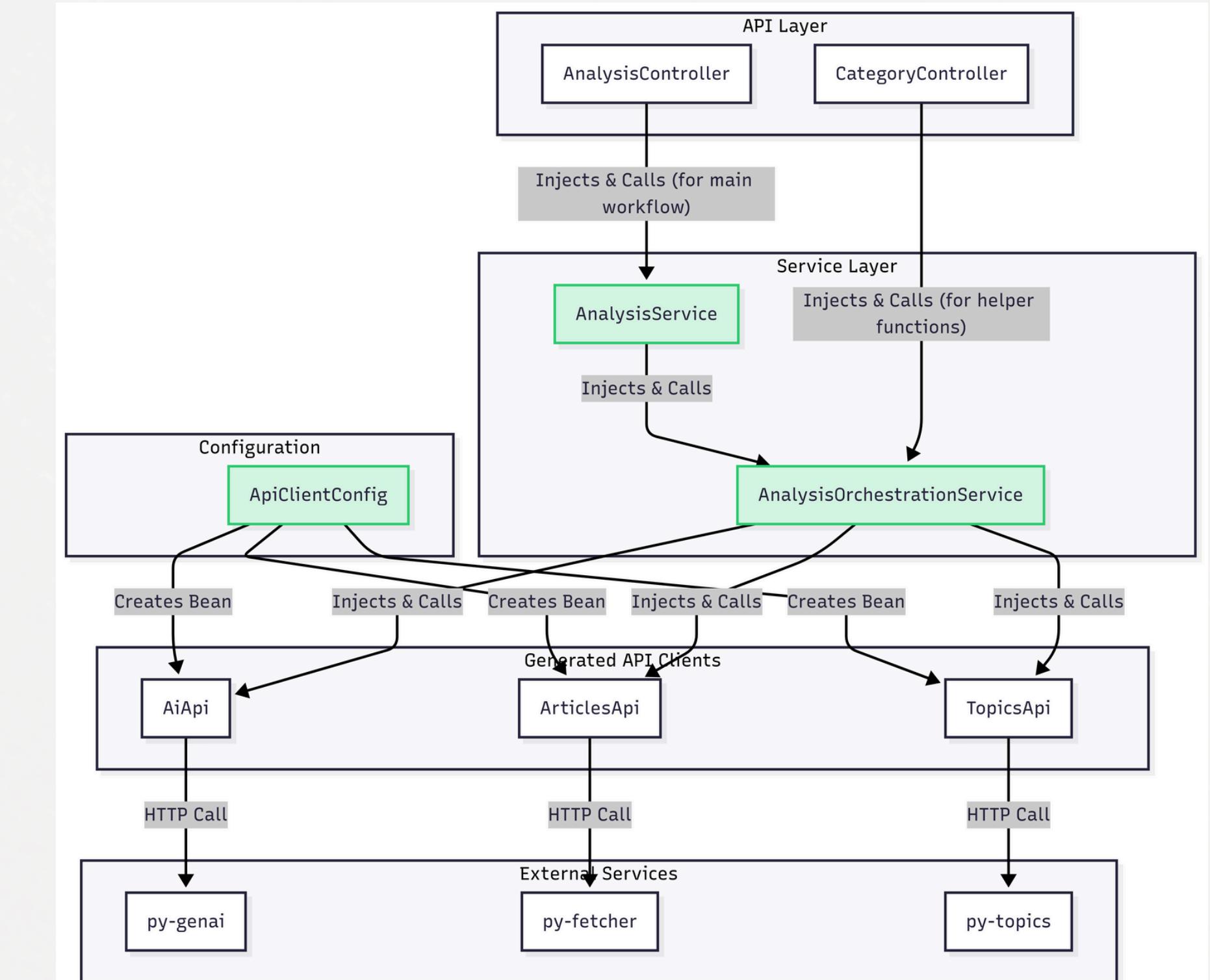
(Terraform)
Ansible
Docker Compose
AWS Docker Compose Overwrite

Dropped out Teammember

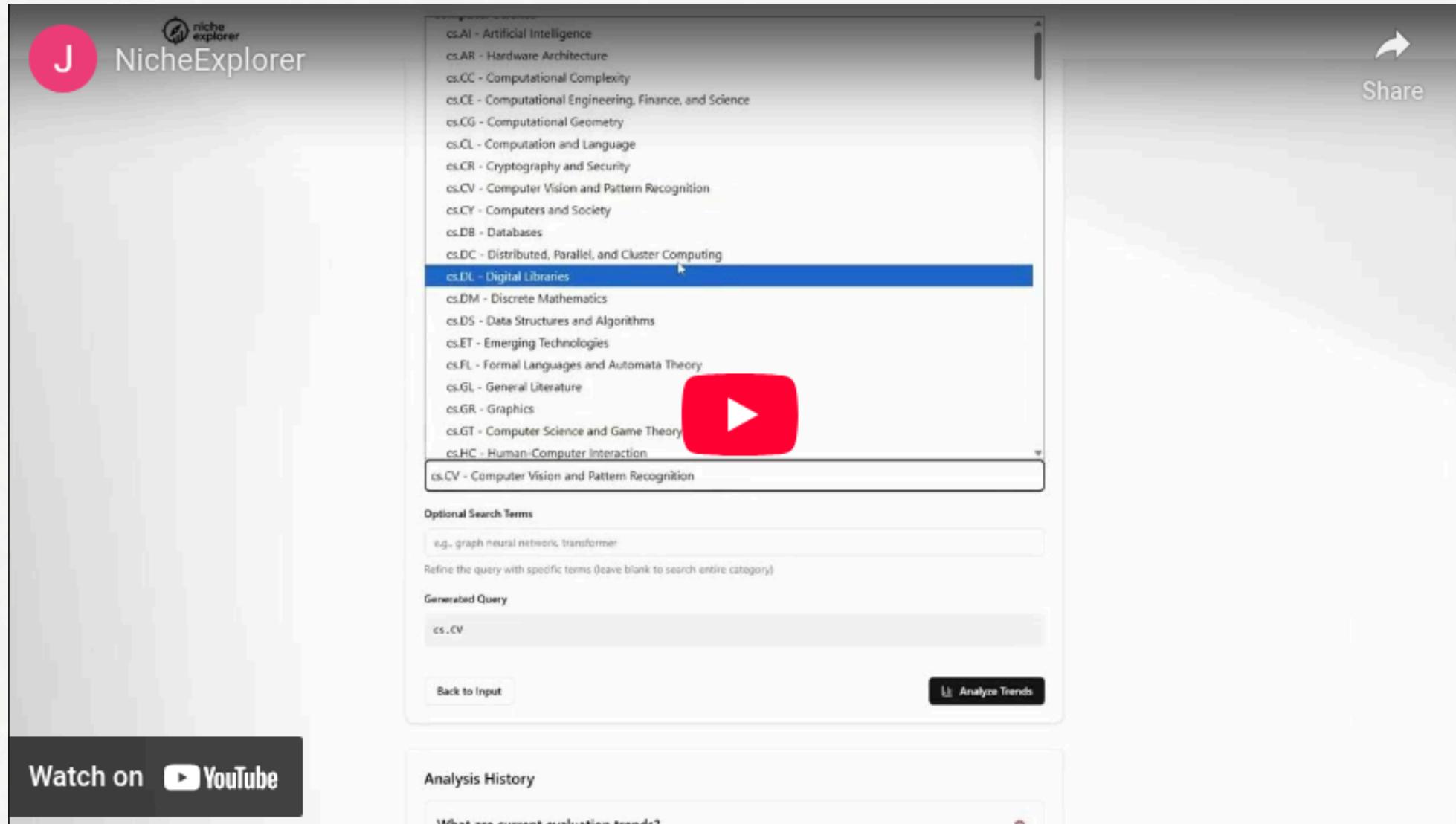
- **When:** Midway through the project
- **Contribution:** No real contribution to the project
 - Lead to more work (major refactorings)
- **Missing parts of the project due to that reason:**
 - Query Responses in frontend will just be empty on errors / when finding nothing
 - Smoother project startup (2-5 minutes wait time before everything running)
 - More test (such as client tests)
 - Credentials management (Keycloak)

THANK YOU FOR LISTENING
QUESTIONS?

Appendix - Java



Appendix - Demo



The screenshot shows the Niche Explorer web application. At the top left is a red circular icon with a white letter 'J'. To its right is the 'NicheExplorer' logo with a small icon above it. On the far right is a 'Share' button with a share icon. The main area contains a sidebar with a list of computer science categories. A blue bar highlights 'cs.DL - Digital Libraries'. Below the sidebar is a search bar with placeholder text 'e.g., graph neural network, transformer'. Underneath is a 'Generated Query' field containing 'cs.CV'. At the bottom are 'Back to Input' and 'Analyze Trends' buttons. A 'Watch on YouTube' button with a YouTube icon is at the very bottom left. An 'Analysis History' section is partially visible at the bottom.

- cs.AI - Artificial Intelligence
- cs.AR - Hardware Architecture
- cs.CC - Computational Complexity
- cs.CE - Computational Engineering, Finance, and Science
- cs.CG - Computational Geometry
- cs.CL - Computation and Language
- cs.CR - Cryptography and Security
- cs.CV - Computer Vision and Pattern Recognition
- cs.CY - Computers and Society
- cs.DB - Databases
- cs.DC - Distributed, Parallel, and Cluster Computing
- cs.DL - Digital Libraries**
- cs.DM - Discrete Mathematics
- cs.DS - Data Structures and Algorithms
- cs.ET - Emerging Technologies
- cs.FL - Formal Languages and Automata Theory
- cs.GL - General Literature
- cs.GR - Graphics
- cs.GT - Computer Science and Game Theory
- cs.HC - Human-Computer Interaction
- cs.CV - Computer Vision and Pattern Recognition

<https://www.youtube.com/watch?v=JZuyDbpnB-A>