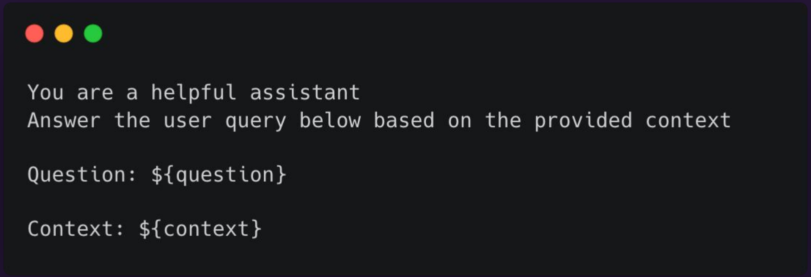
Lost Minute Travel

A state-of-the-art RAG chatbot on Cloudflare

By LiquidMetal AI

Simple RAG

- PDFs embedded into **Vectorize**
- KNN-search to find K matches
- Insert text chunks into prompt



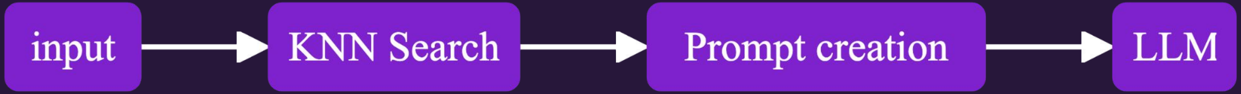
```
You are a helpful assistant
Answer the user query below based on the provided context

Question: ${question}

Context: ${context}
```

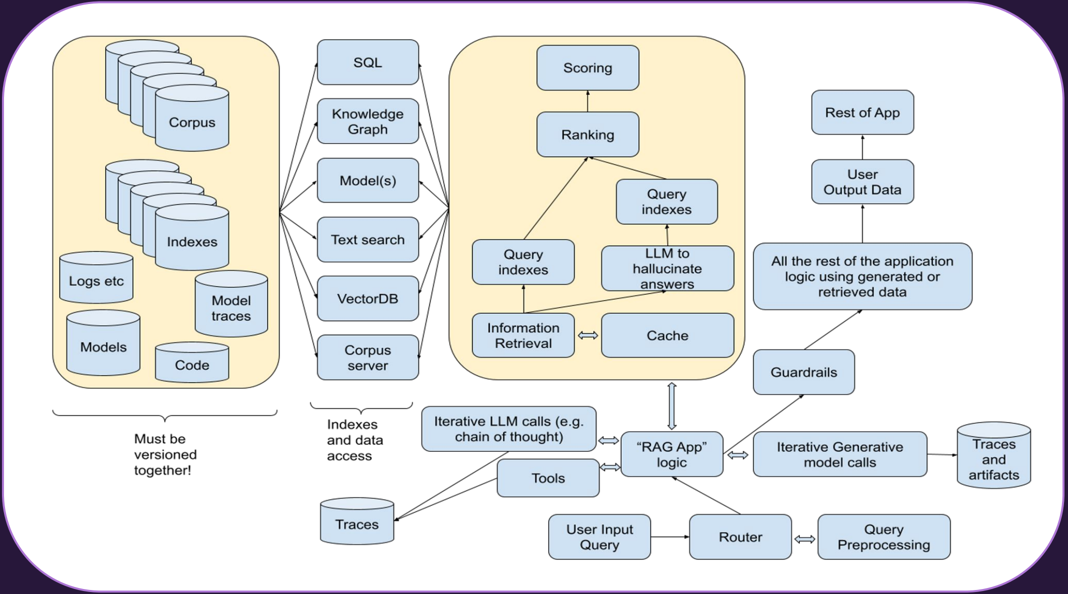
Cool, but we can do better!

Simple RAG



input → KNN Search → Prompt creation → LLM

State-Of-The-Art (SOTA) - RAG



A real SOTA RAG application

- Indexes PDFs, webpages, images, tables, wikipedia and any other data source
- Finds context based on embeddings, entity relationships, topic and topic relationships etc.
- Data management (governance, lineage, metadata)
- Uses a cognitive architecture to model the behaviour

Architecture

- Indexer (Today)
- Retriever and Cognitive Architecture (future talk)

Where to begin?

This is a lot for a lightning talk, but I’ll be around if you want to ask questions

email : fokke@liquidmetal.ai

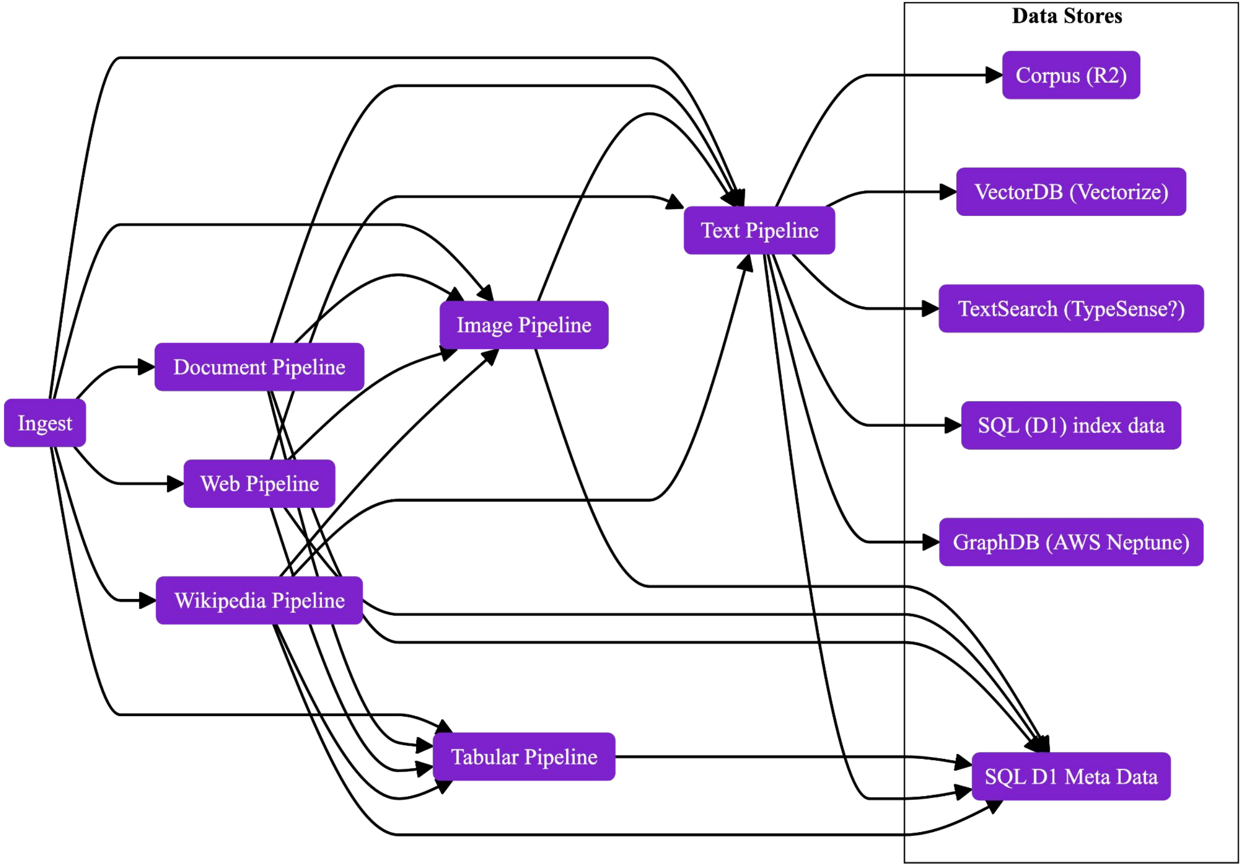
Data and Infra

- **Cloudflare workers** - Compute
- **Cloudflare queues** - Pipes
- **Cloudflare durable objects** - Cognitive architecture
- Data storage:
  - Data lake in **R2**
  - **D1** for metadata and tabular data
  - **Vectorize** for vector embeddings
  - AWS Neptune Graph Database
  - Typesense for text search

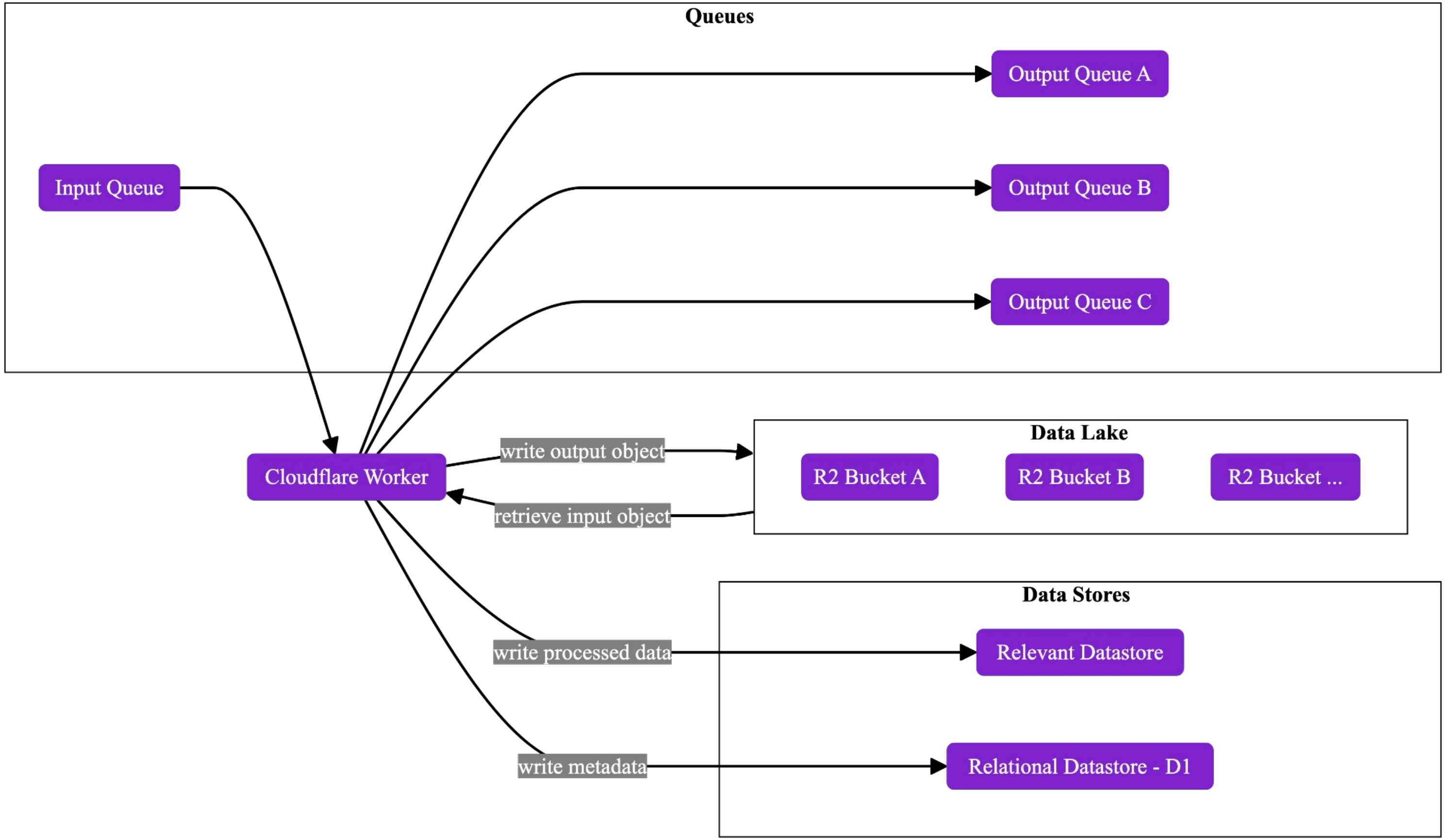
Models

- **Cloudflare AI**:
  - Various LLMs (mostly Llama-based)
  - resnet-50 (image classification)
  - bge-base-en-v1.5 (embeddings)
  - Llava-1.5-7b-hf (image to text)
- Other models:
  - LDA (topic modeling)
  - Wikineural-multilingual-ner (entity recognition)
  - GLiNER (entity recognition)
  - 51-languages-classifier

Indexer Architecture



One Indexer


Queues
Input Queue
Output Queue A
Output Queue B
Output Queue C
Cloudflare Worker
write output object
retrieve input object
Data Lake
R2 Bucket A
R2 Bucket B
R2 Bucket ...
Data Stores
write processed data
Relevant Datastore
write metadata
Relational Datastore - D1

~250 resources to manage

Day 2 challenges

- Managing (large) data generated at the edge for and by your app
  - Data governance of structured and unstructured data
  - Data discovery for new applications
  - Data lineage and metadata, annotations, etc.
- Managing the application lifecycle and resource management
  - Create complex applications on the platform
  - Systematically capture, version, and manage data and code
  - Working with versions of code and data as a unit as you innovate: velocity in continuously improving your app is a core differentiating factor for your success!

Get in touch

Thank you!