MINIO

Building a Data Infrastructure for AI/ML

Keith Pijanowski
MinIO

Agenda

- Data Lakehouses
- MLOps
- Traditional AI
- Generative AI
- Distributed Training
- Considerations for GPU usage

The Data Lakehouse

Data Warehouse + **Data Lake** = **Data Lakehouse**

Open Table Formats

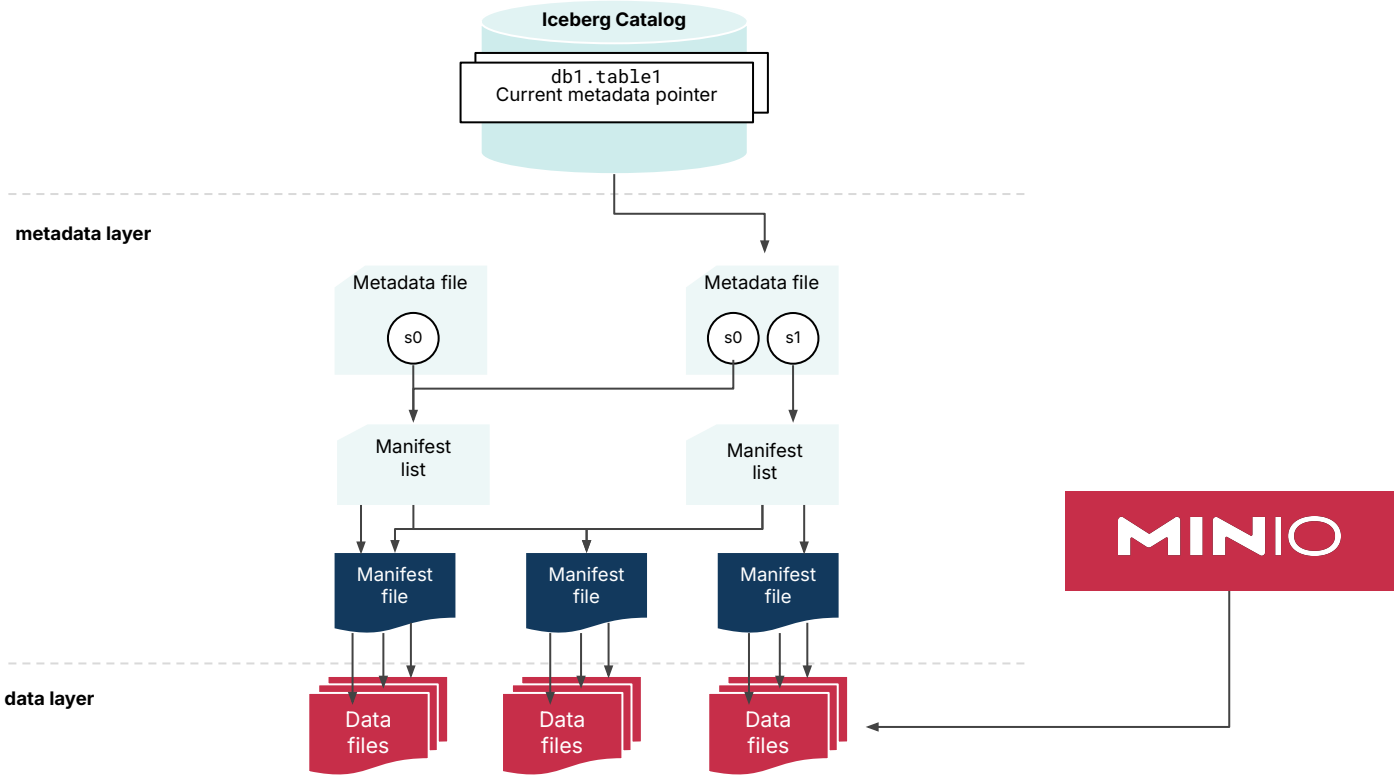
- Apache Iceberg: key contributor Netflix
- Apache Hudi: key contributor Uber
- Delta Lake: key contributor Databricks

An OTF Based Data Warehouse

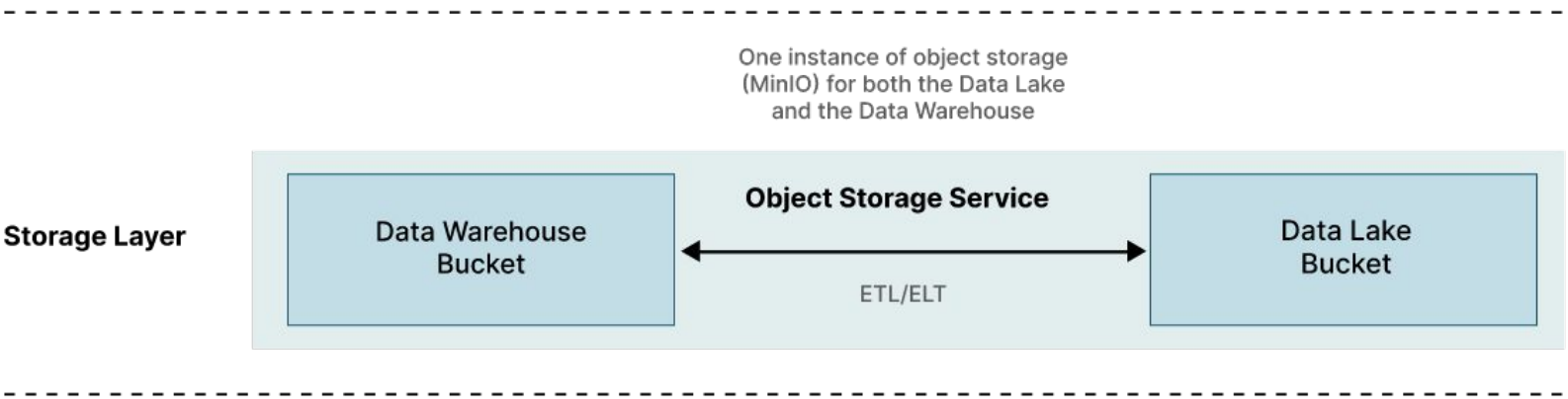
OTF Based Data Warehouse

- Acid Transactions
- Schema Evolution
- Partition Evolution
- Time Travel
- External Tables
- Zero Copy Branching

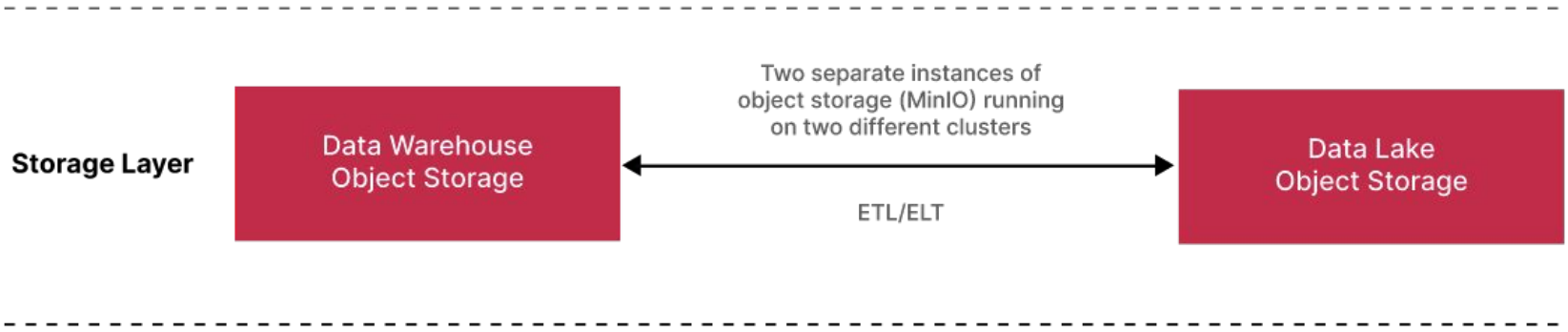
Apache Iceberg Architecture



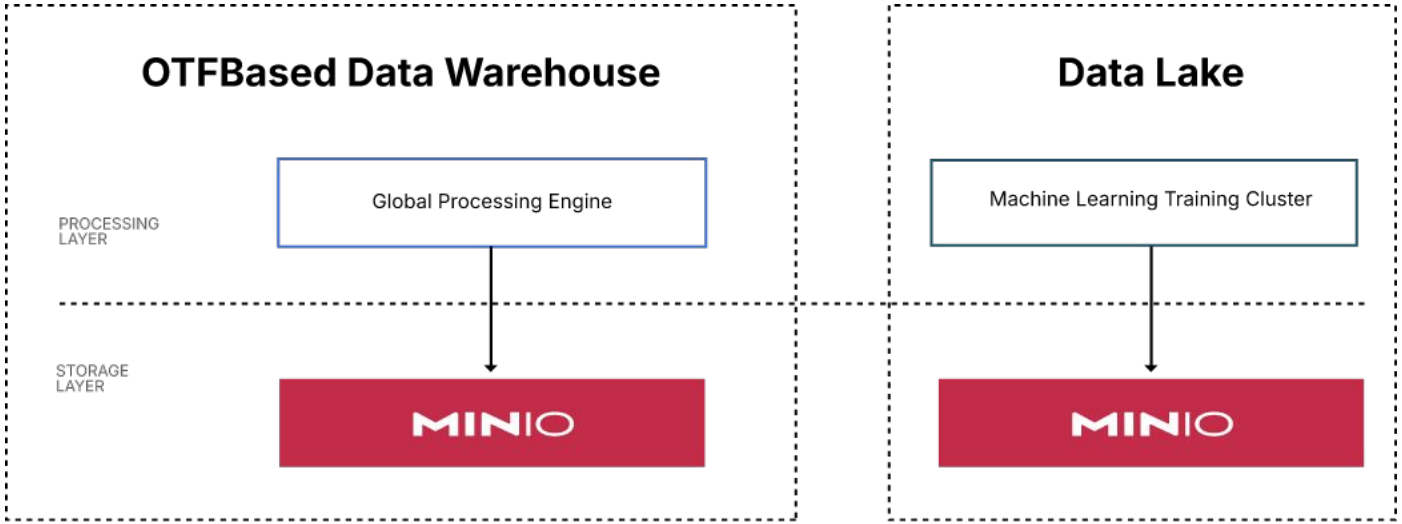
A Simple Option for the Storage Layer



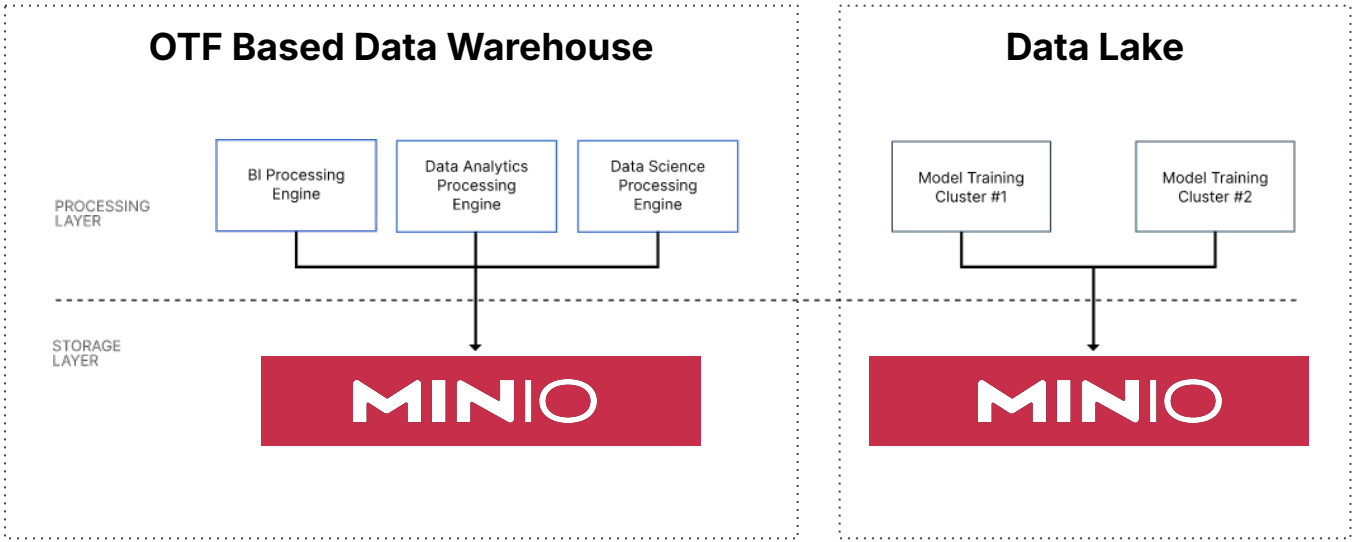
Two Instance of Min for the Storage Layer



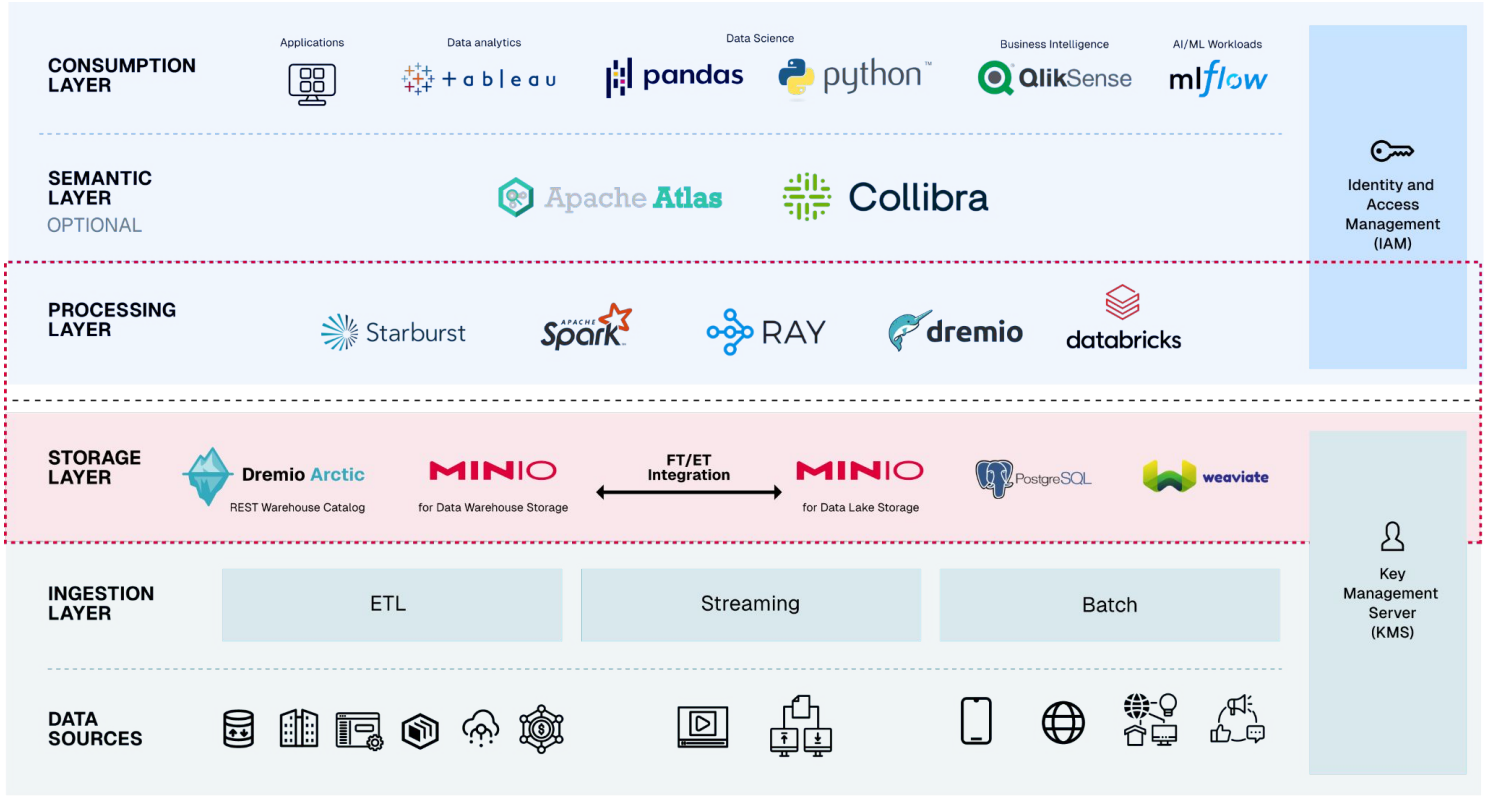
A Simple Option for the Processing Layer



Disaggregated Compute



Reference Architecture



Machine Learning Operations

Machine Learning vs. Application Development

- **Application Development**

  Coding is the main activity

  Data does not change while coding

  Unit test and end to end test

- **Machine Learning**

  Experimentation is the main activity

  Data Changes

  Metrics track quality

MLOps Features

1. Support from a major player
2. Data Lakehouse Integration
3. Experiment Tracking
4. Facilitate Collaboration
5. Training Pipeline
6. Data Pipeline
7. Serverless Functions
8. Model Registry
9. Model Packaging
10. Model Serving

MLflow from Databricks

1. Support from a major player - **Databricks**
2. Data Lakehouse Integration - **Data Lake integration**
3. Experiment Tracking - **Yes**
4. Facilitate Collaboration - **Yes**
5. Training Pipeline - **Some using MLflow Recipes**
6. Data Pipeline - **Some using MLflow Recipes**
7. Serverless Functions - **No**
8. Model Registry - **Yes using Data Artifacts**
9. Model Packaging - **Yes**
10. Model Serving - **Yes**

Kubeflow from Google

1. Support from a major player - **Google**
2. Data Lakehouse Integration - **Data Lake integration**
3. Experiment Tracking - **Yes**
4. Facilitate Collaboration - **Yes**
5. Training Pipeline - **Yes, using Kubeflow Pipelines**
6. Data Pipeline - **Yes, using Kubeflow Pipelines**
7. Serverless Functions - **Yes, Hermetic Functions, Python Containers, Custom**
8. Model Registry - **Yes using Data Artifacts**
9. Model Packaging - **Yes via an extension**
10. Model Serving - **Yes via an extension**

MLRun from Iguazio (McKinsey and Company)

1. Support from a major player - **McKinsey and Co.**
2. Data Lakehouse Integration - **Data Lake integration**
3. Experiment Tracking - **Yes**
4. Facilitate Collaboration - **Yes**
5. Training Pipeline - **Yes**
6. Data Pipeline - **Yes**
7. Serverless Functions - **Yes using Nuclio and KubeFlow Pipelines**
8. Model Registry - **Yes using Data Artifacts**
9. Model Packaging - **Yes**
10. Model Serving - **Yes**

Definitions

- **Traditional AI**

  Directly model the relationship between input data and output labels.

  Examples: Regression, Categorization, and Classification

- **Generative AI**

  Probability Distributions

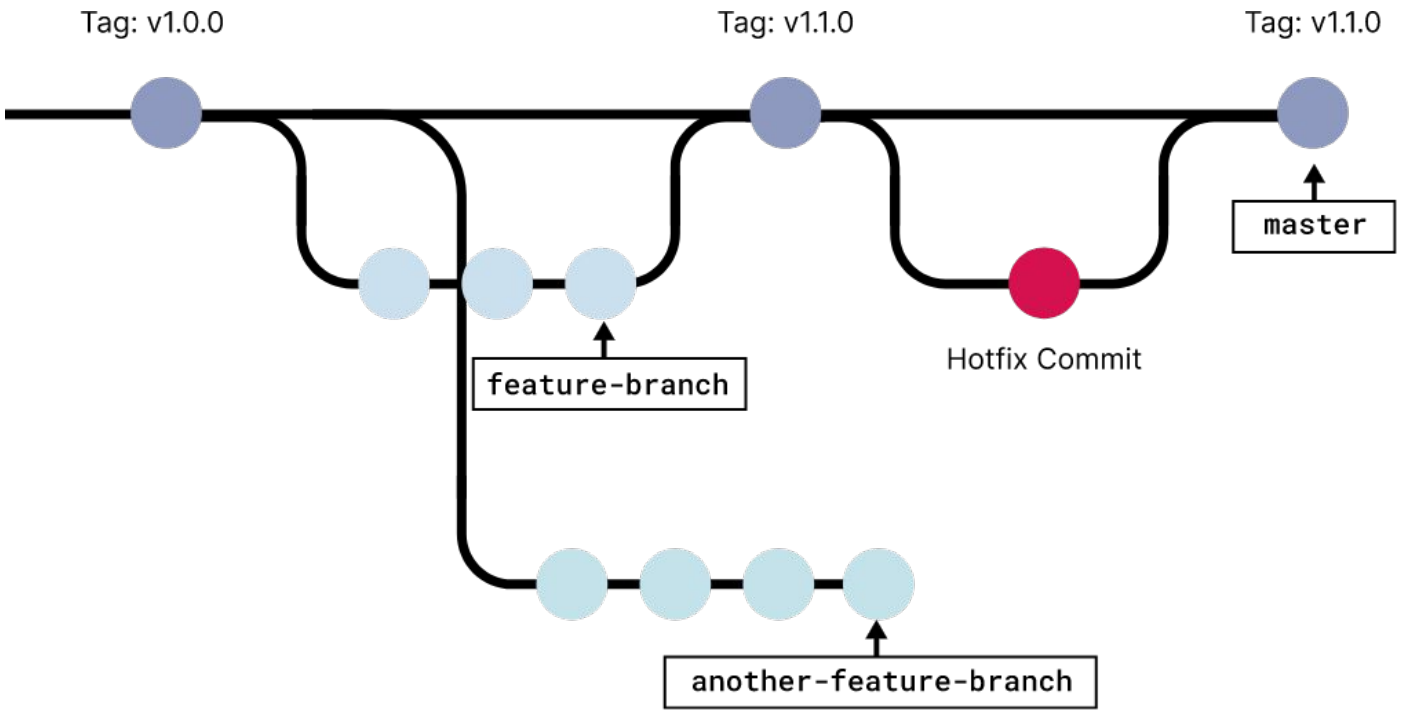
  Generates new data. Trained on unstructured data (text). Harder to test.

Traditional AI

Traditional AI and the Data Lake

- **Unstructured Data lives in the Data Lake**
  - Examples: Images, Videos, Audio
- **Structured Data may also live in the Data Lake.**
  - Examples: AVRO, Parquet
- **Add structured data to the Data Warehouse of other workloads will benefit from it.**

Zero Copy Branching



Generative AI

MINIO

Document Pipelines

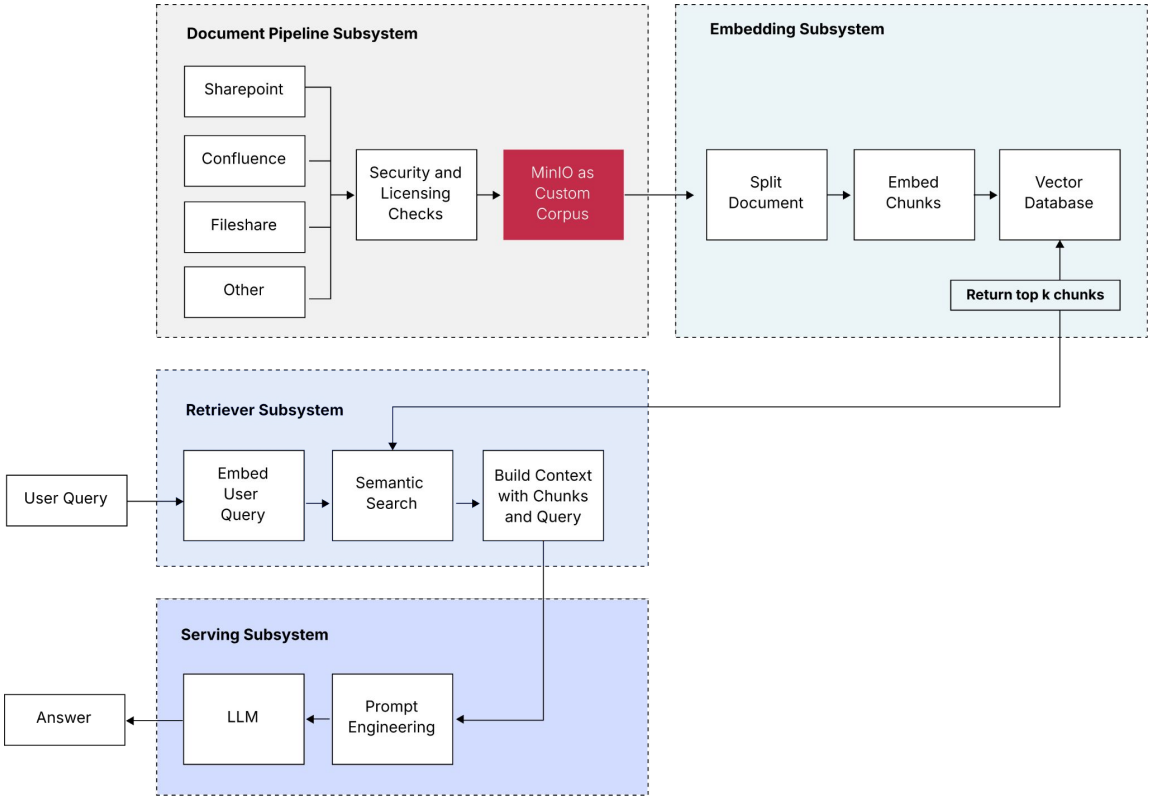
Search without a Vector DB

```
SELECT snippet
FROM MyCorpusTable
WHERE (text like '%artificial
intelligence%' OR
        text like '%ai%' OR
        text like '%machine learning%' OR
        text like '%ml%' OR
         ... and on and on ...
```

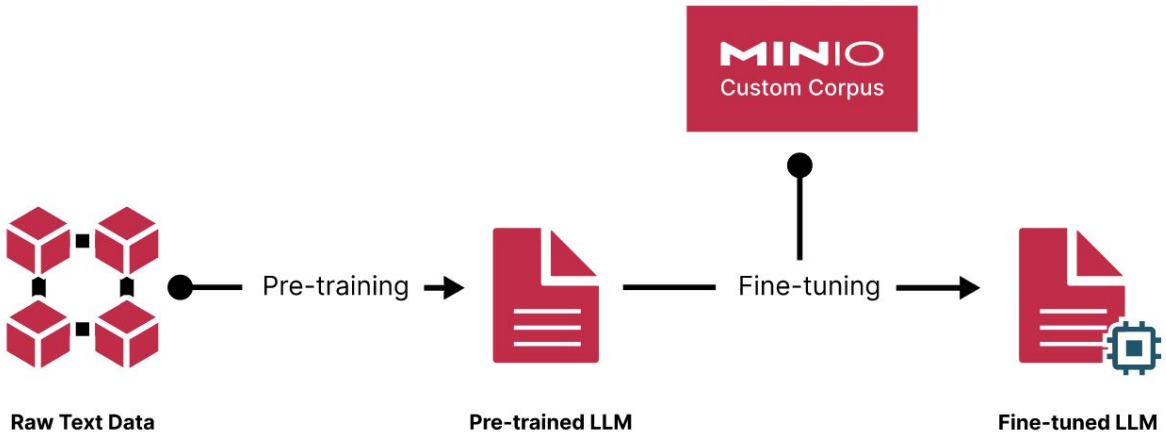
Search with a Vector DB

```
{
Get {
        MyCorpusTable(nearText: {concepts:
["artificial intelligence"]})
      {snippet}
    }
}
```

Retrieval Augmented Generation (RAG)



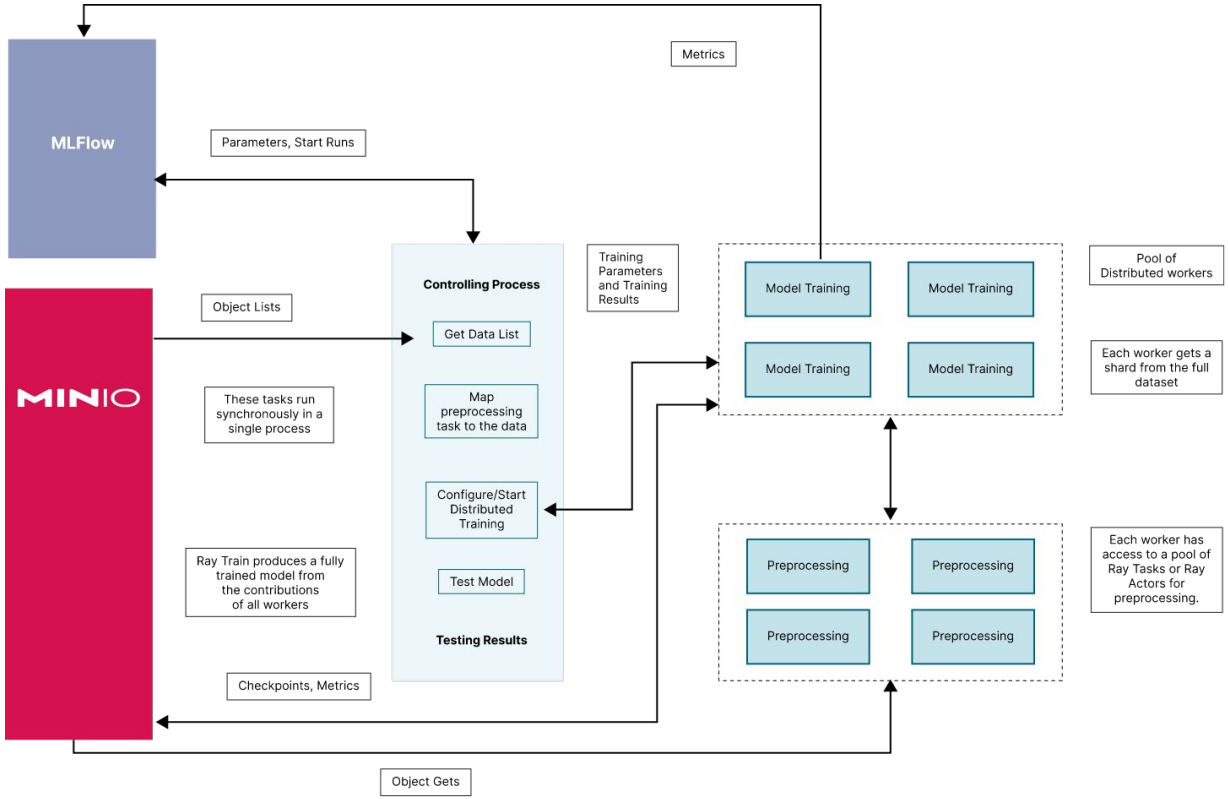
Fine-Tuning LLMs



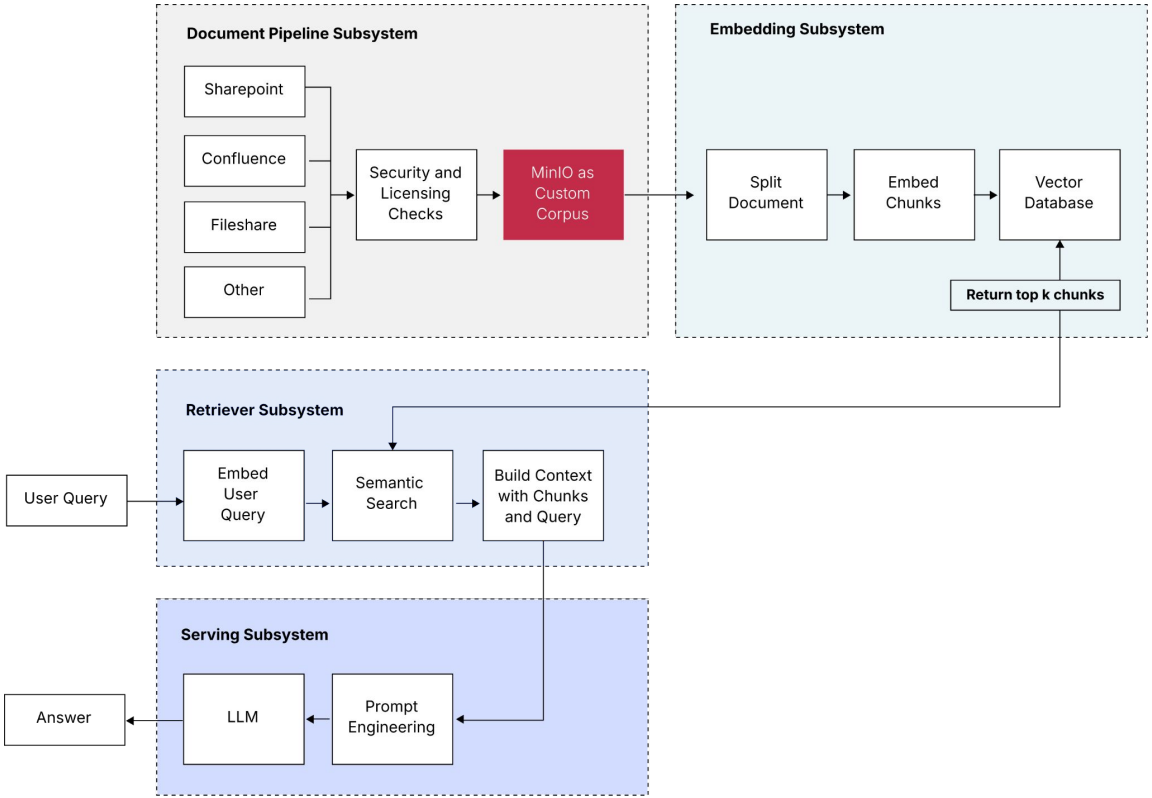
Fine-tuning Process

Distributed Training

Distributed Training for Traditional AI

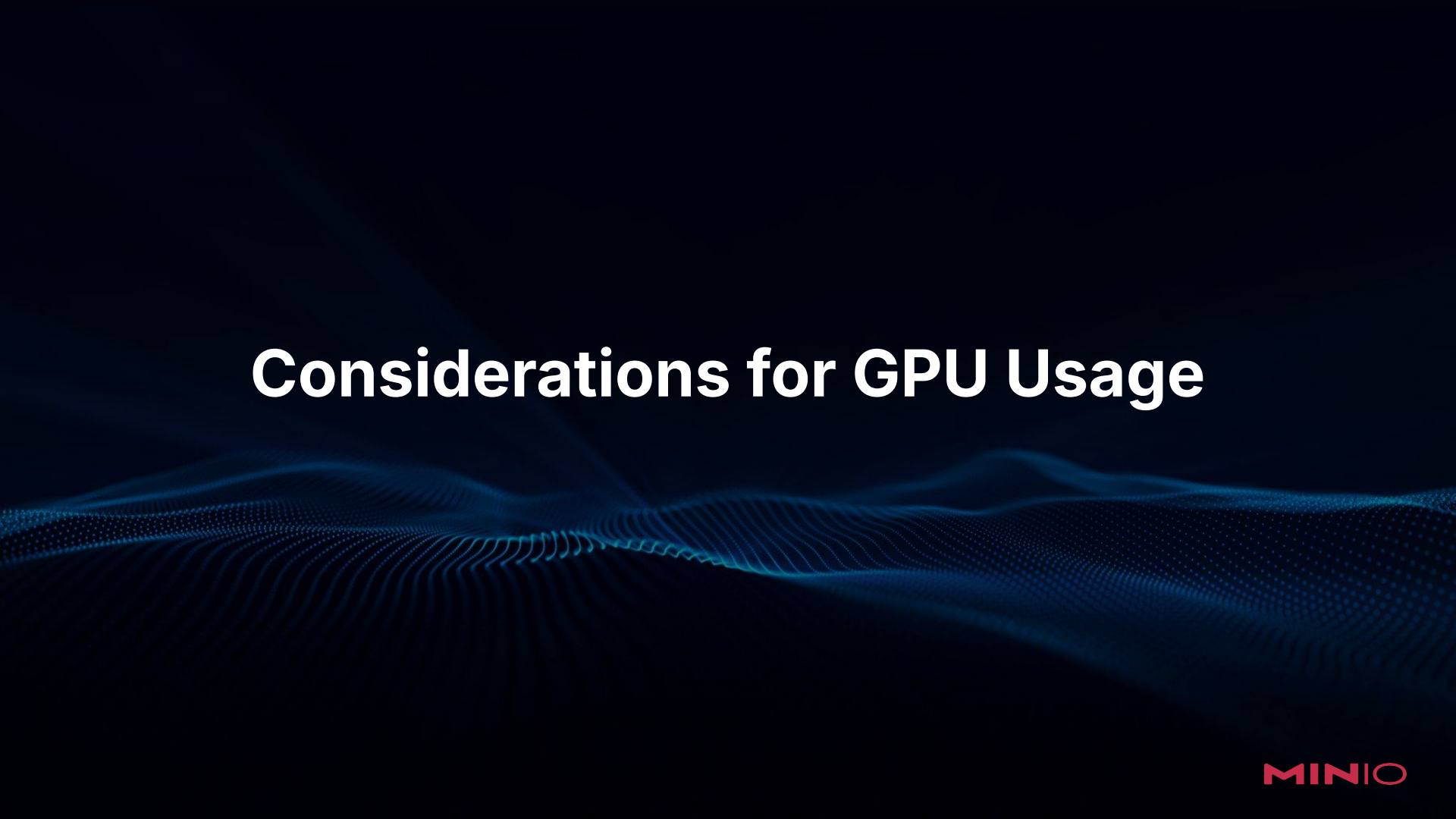


RAG Needs Distributed Training



[Build a Distributed Embedding Subsystem with MinIO, Langchain, and Ray Data]

Considerations for GPU Usage



The Current State of GPUs

Release Date	GPU	Performance	Memory	Memory Bandwidth	Cost
May 2020	**A100**	0.62 petaFLOPS	40 GB	1,555 GB/s	$10,000
March 2022	**H100**	1.98 petaFLOPS	80 GB	3.35 TB/s	$25,000
June 2024	**H200**	1.98 petaFLOPS	141 GB	4.8 TB/s	$30,000
Early 2025	**B100**	3.5 petaFLOPS	192 GB	8 TB/s	$35,000
Early 2025	**B200**	4.5 petaFLOPS	192 GB	8 TB/s	$40,000

One petaFLOP

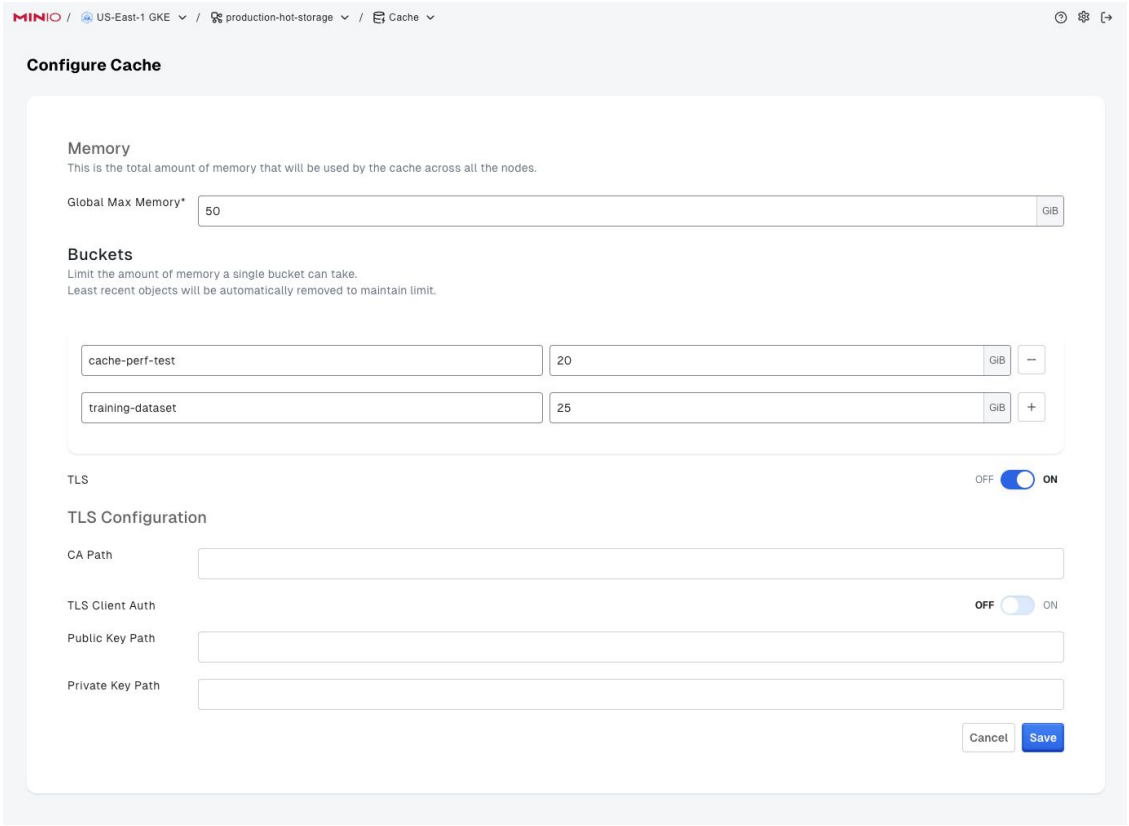
petaFLOP:
1,000,000,000,000,000

B200:
4,500,000,000,000,000

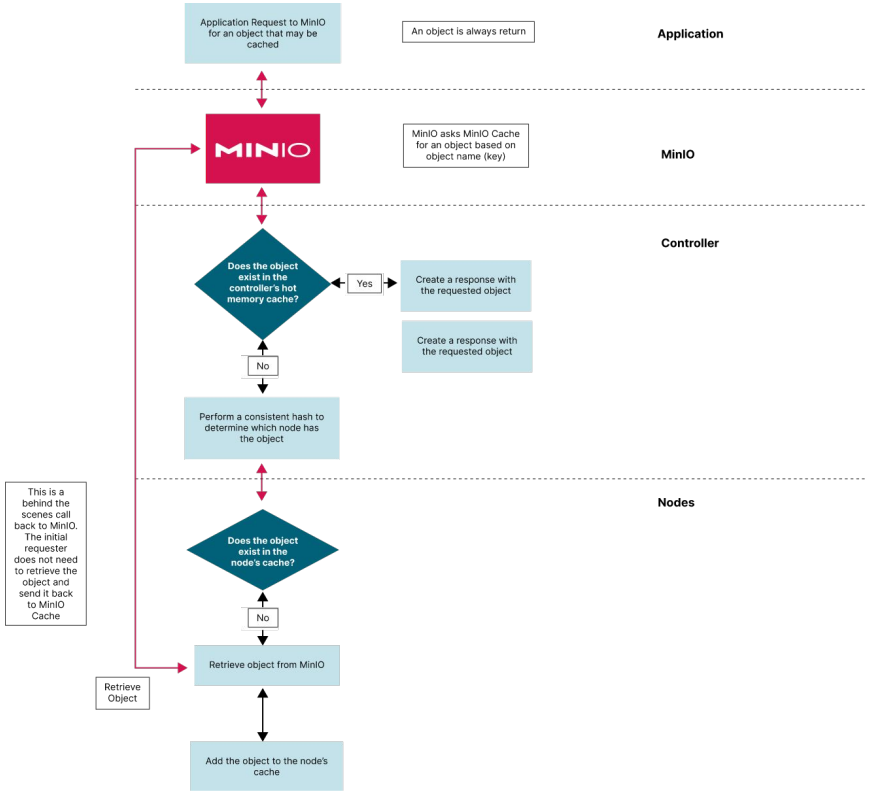
The Starving GPU Problem

The Starving GPU Problem - Simplified

MinIO Cache


MINIO / US-East-1 GKE / production-hot-storage / Cache
Configure Cache
Memory
This is the total amount of memory that will be used by the cache across all the nodes.
Global Max Memory*
50
GiB
Buckets
Limit the amount of memory a single bucket can take.
Least recent objects will be automatically removed to maintain limit.
cache-perf-test
20
GiB
training-dataset
25
GiB
TLS
OFF
ON
TLS Configuration
CA Path
TLS Client Auth
OFF
ON
Public Key Path
Private Key Path
Cancel
Save

MinIO Cache


Application Request to MinIO for an object that may be cached
An object is always return
Application
MINIO
MinIO asks MinIO Cache for an object based on object name (key)
MinIO
Controller
Does the object exist in the controller's hot memory cache?
Yes
Create a response with the requested object
Create a response with the requested object
No
Perform a consistent hash to determine which node has the object
Nodes
This is a behind the scenes call back to MinIO. The initial requester does not need to retrieve the object and send it back to MinIO Cache
Does the object exist in the node's cache?
No
Retrieve object from MinIO
Retrieve Object
Add the object to the node's cache

FLOPS and More FLOPS

Name	Unit	Value
kiloFLOPS	kFLOPS	10^3
megaFLOPS	MFLOPS	10^6
gigaFLOPS	GFLOPS	10^9
teraFLOPS	TFLOPS	10^{12}
petaFLOPS	PFLOPS	10^{15}
exaFLOPS	EFLOPS	10^{18}
zettaFLOPS	ZFLOPS	10^{21}
yottaFLOPS	YFLOPS	10^{24}
ronnaFLOPS	RFLOPS	10^{27}
quettaFLOPS	QFLOPS	10^{30}

https://en.wikipedia.org/wiki/FLOPS

Blog and White Paper Links Information

MLOps
The Architects Guide to Machine Learning Operations (MLOps)

MLflow
Setting up a Development Machine with MLFlow and MinIO
MLflow Tracking and MinIO
MLflow Model Registry and MinIO

MLRun
Setting Up A Development Machine with MLRun and MinIO
Model Training and MLOps using MLRun and MinIO

KubeFlow
Setting up a Development Machine with Kubeflow Pipelines 2.0 and MinIO
Building an ML Data Pipeline with MinIO and Kubeflow v2.0
Building an ML Training Pipeline with MinIO and Kubeflow v2.0

Blog and White Paper Links Information

Distributed Training

Distributed Training with Ray Train and MinIO
Distributed Data Processing with Ray Data and MinIO
Distributed Training and Experiment Tracking with Ray Train, MLflow, and MinIO

Generative AI

Build a Distributed Embedding Subsystem with MinIO, Langchain, and Ray Data
Improve RAG Performance with Open-Parse Intelligent Chunking

OTF-based Data Warehouse

Data Lake Mysteries Unveiled: Nessie, Dremio, and MinIO Make Waves
Building Modern Data Architectures with Iceberg, Tabular and MinIO
The Disruptive Nature of Data Lakehouses
A Developer's Introduction to Apache Iceberg using MinIO
Building a Data Lakehouse using Apache Iceberg and MinIO

Reference Architectures

Modern Datalake Reference Architecture
AI/ML Within A Modern Datalake

Summary

- The Data Lakehouse
- MLOps Feature List
- Traditional Workloads
- Generative Workloads
- Distributed Training
- GPU Usage

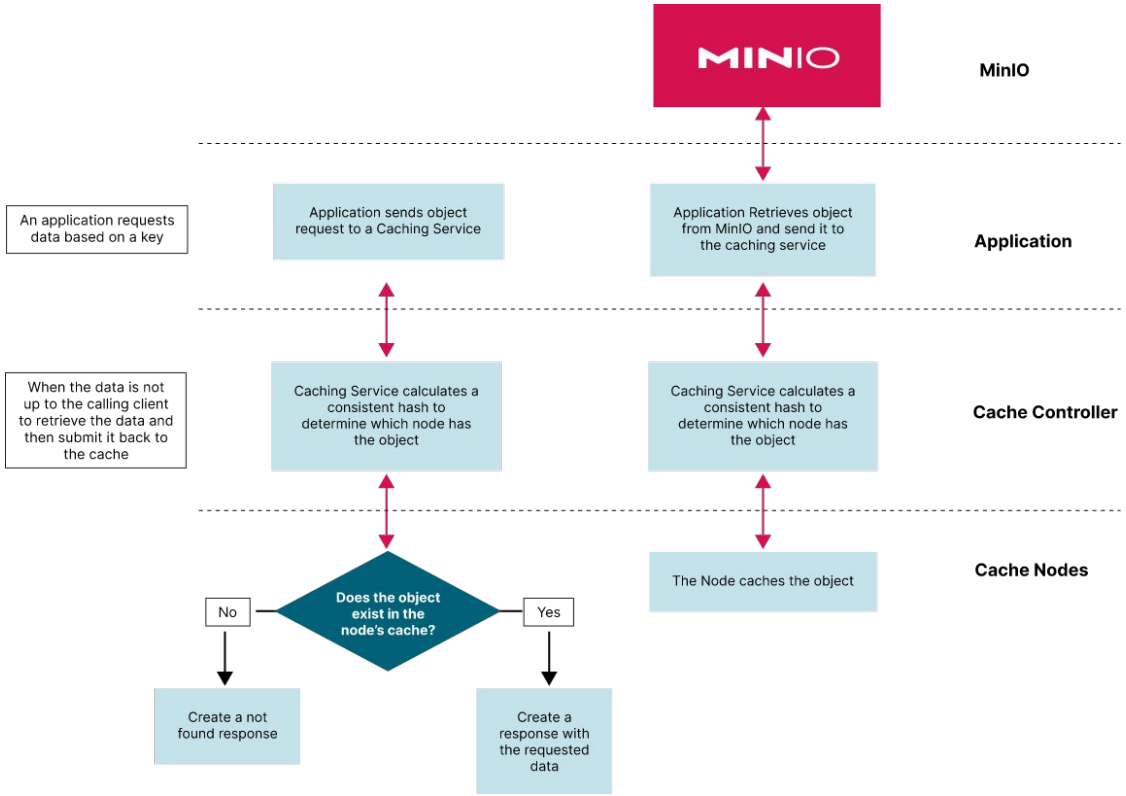
MINIO

Thank you!

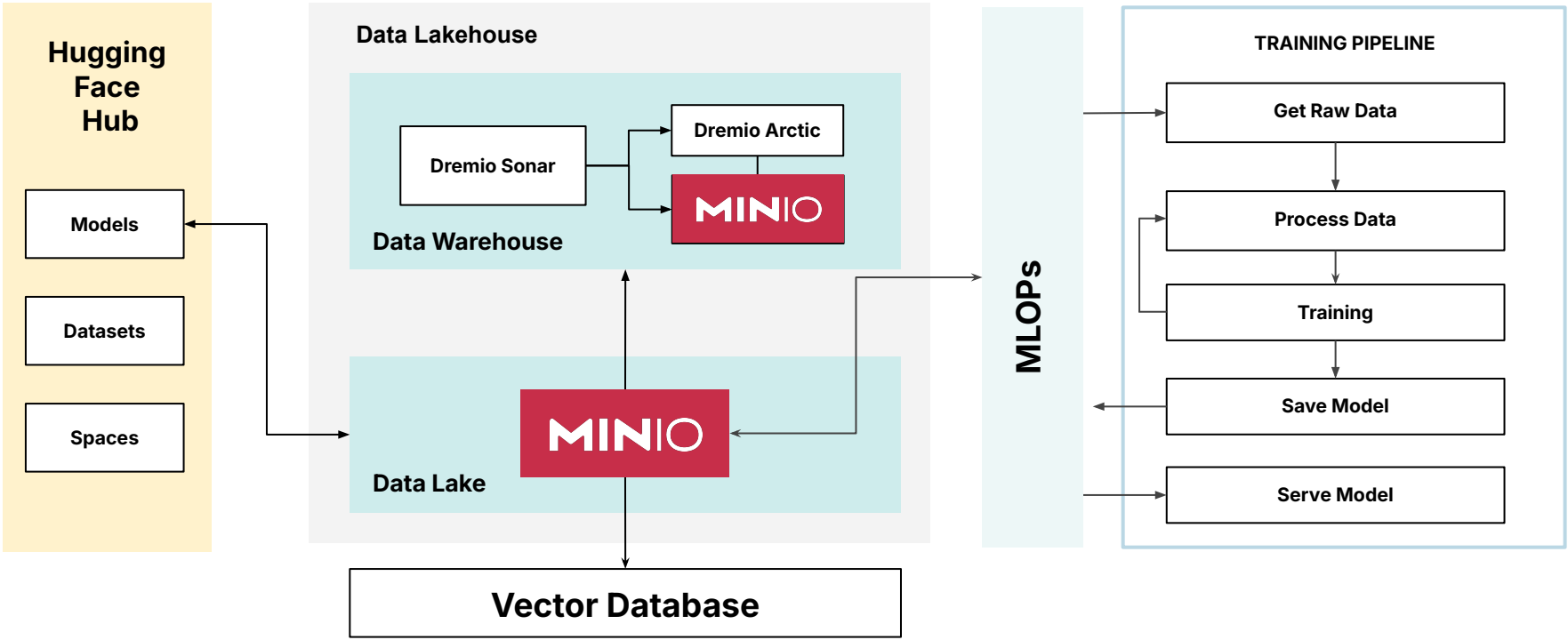
For more information:
keith@min.io

- @minio
- https://github.com/minio/minio
- https://slack.min.io
- https://min.io

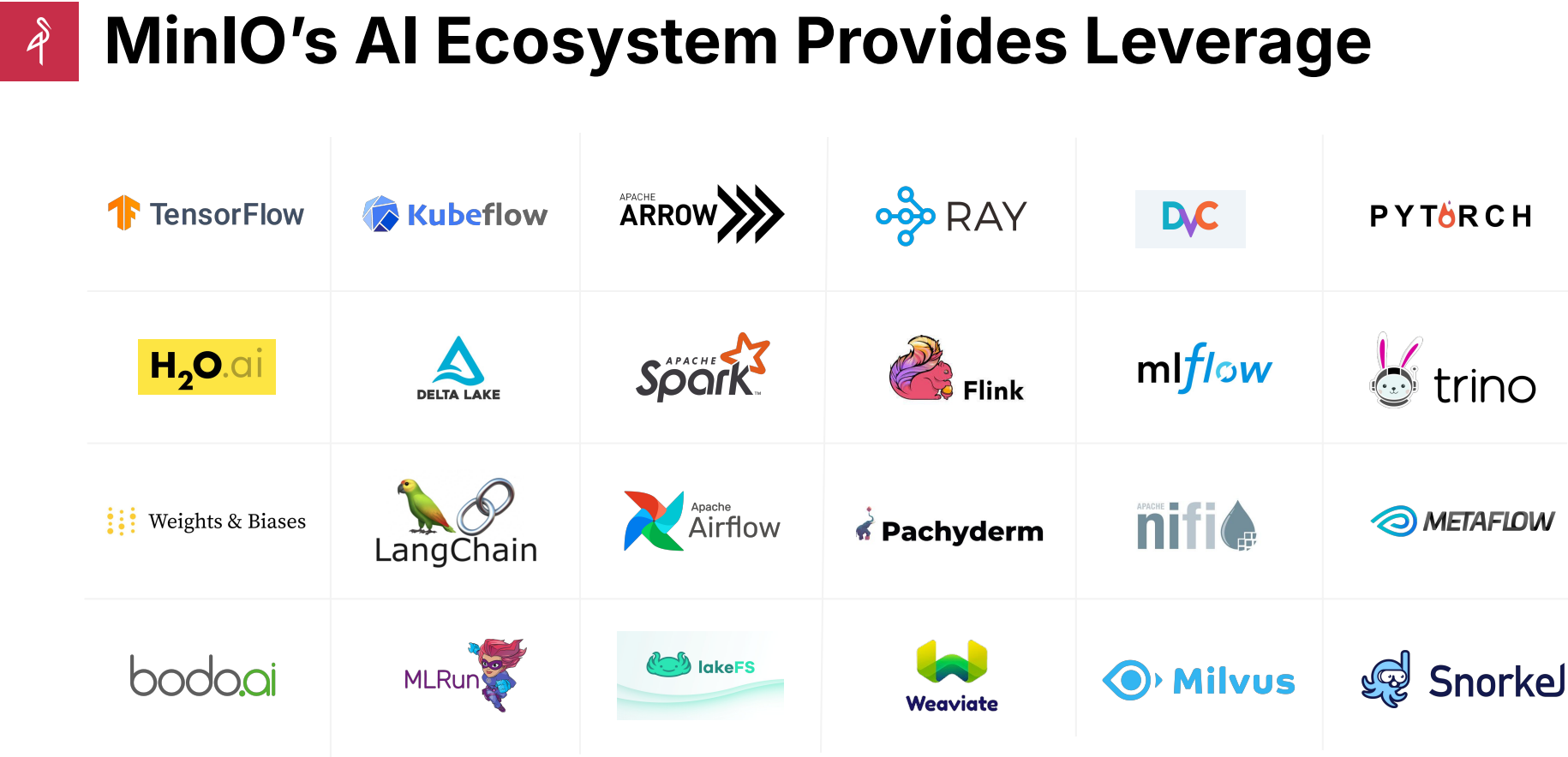
Generic Cache



Modern System Architecture for AI Workloads



MinIO's AI Ecosystem Provides Leverage



MINIO

Uses of Generative AI

- Research (Questions and Answers)
- ChatBot for Support (Conversational AI)
- Document Creation
- Summarization
- Language Translation and Localization
- Entity Recognition