Kick off - Feature Store Summit 2025 Real-Time AI, LLMs and Vector Databases

Jim Dowling
CEO & Co-Founder
Hopsworks

FEATURE STORE SUMMIT 2025

Organized by HOPSWORKS

Feature Store Summit 2025

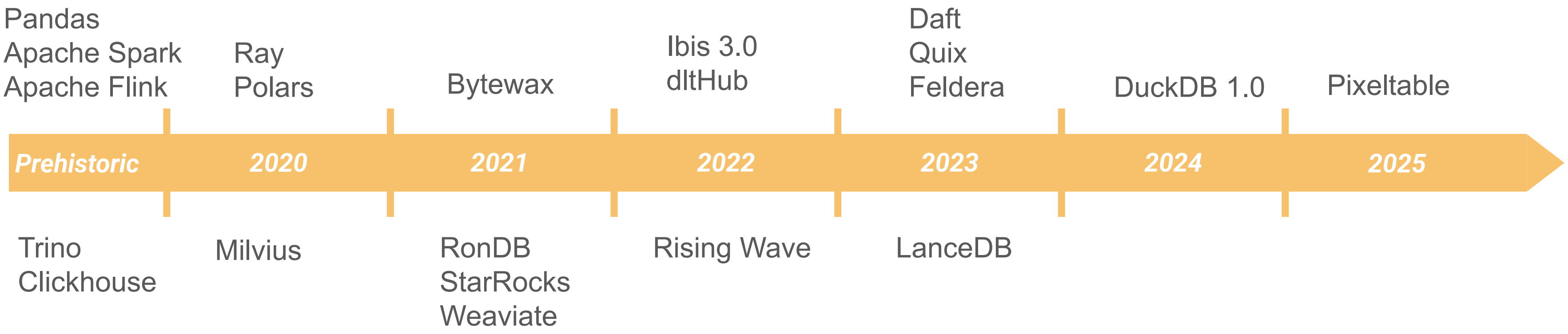
Real-Time AI, LLMs and Vector Databases

13 Presentations from 15 Speakers

Feature Store State of the Union in 2025

2017	2019	2021	2022	2023	2024	2025
Michelangelo from Uber	First Open-Source Feature Stores	First Cloud Vendors Feature Stores	Feature Stores go Real-Time	Feature Stores go Serverless	Offline Store is the Lakehouse	Platform Consolidation, Agents and Context
		Feature Store Summit v1	**Feature Store Summit v2**	**Feature Store Summit v3**	**Feature Store Summit v4**	**Feature Store Summit v5**

Cambrian Explosion in Open-Source Data Engines*

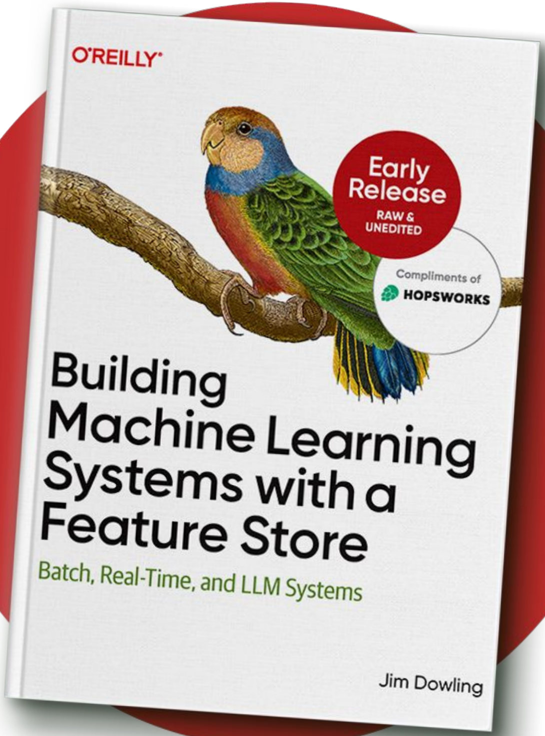


*Write to us on Slack if we are missing any other engines

Building Machine Learning Systems

Batch, Real-Time, and LLM Systems

Scan the QR code and add a Promo Code:
'FS2025' to reserve your spot
to receive the complete digital edition



Sovereign AI Blueprint

What does It Take to Build Truly Trustworthy AI? Download our comprehensive guide with market research at global scale on Sovereign AI.

In our survey of 100+ responses, we uncovered:

- What's actually working today
- The 3 barriers every team is facing
- How top organizations are overcoming them

📘 Get practical insights from the teams already building Sovereign AI systems and see how you can move from vision to execution.

Link: https://www.hopsworks.ai/lp/blueprints/sovereign-ai

Today's Agenda

🎙️ Presentations ⚡Intro/Closing ☕ Breaks

TODAY'S AGENDA:
All times are Pacific Time USA

8:30 AM	⚡	**Kick-off** Jim Dowling, CEO & Co-Founder, Hopsworks
8:40 AM	🎙️	**From Real-Time ML to Agents with Hopsworks** Jim Dowling, CEO & Co-Founder, Hopsworks
9:15 AM	🎙️	**Lyft's Feature Store: Architecture, Optimization, and Evolution** Rohan Varshney, Senior Software Engineer, Lyft
9:40 AM	🎙️	**Powering Real-Time AI at Pinterest: Feature Management and Serving at Scale with Galaxy and Scorpion** Andrey Gubenko, Software Engineer, Pinterest Li Tang, Software Engineer, Pinterest
10:05 AM	🎙️	**Vector Store: Uber's Embedding Platform** Divya Nagar, Staff Software Engineer, Uber Xiyuan Feng, Software Engineer, Uber
10:30 AM	☕	**Break**

🎙️ Presentations ⚡Intro/Closing ☕ Breaks

TODAY'S AGENDA:
All times are Pacific Time USA

10:40 AM	🎙️	**From EC2 to K8s: Zalando's Journey to Large-Scale, Real-Time Feature Serving.** Morteza Ghasempour, Senior Platform Engineer, Zalando
11:05 AM	🎙️	**Predictive Analytics in Financial Industry** Gokulram Krishnan, Manager - AI & Data, EY
11:30 AM	🎙️	**Real time ML at Roku** Krishna Chaitanya Chakka, Senior ML Engineer, Roku
11:55 AM	🎙️	**Bridging Real-Time and Batch: Declarative Feature Engineering with Apache Hamilton + Narwhals** Ryan Whitten, Director, ML Data Engineering, Best Egg
12:20 PM	☕	**Break**
12:30 PM	🎙️	**How Coinbase Builds Sequence Features for Machine Learning** Joseph McAllister, Senior Engineer, Coinbase

🎙️ Presentations ⚡Intro/Closing ☕ Breaks

TODAY'S AGENDA:
All times are Pacific Time USA

12:50 PM	🎙️	**Real-time ML: Accelerating Python for inference (< 10ms) at scale** Chase Haddleton, Software Engineer, Chalk
13:10 PM	🎙️	**Real-Time Feature Aggregation at Scale: iFood's Path to Sub-Second Latency** Willian Moreira, Machine Learning Platform Lead, iFood
13:30 PM	🎙️	**Building a Generative Recommender with Chronon** Varant Zanoyan, Co-Founder, Zipline AI
13:50 PM	🎙️	**On-Demand Feature Life Cycle Management** Aaron Hunsaker, Machine Learning Systems Engineer, Clicklease
14:05 PM	⚡	**Wrap-Up** Jim Dowling, CEO & Co-Founder, Hopsworks

THANKS TO OUR PARTNERS:

MLOps community Stockholm

NUMFOCUS
OPEN CODE = BETTER SCIENCE

AI SWEDEN

SCAN & JOIN OUR SLACK CHANNEL

featurestoreorg.slack.com

Get the chance to ask direct questions to the speakers after each session.

OUR COMMUNITY

Join our Slack channel

Follow us on Linkedin

Subscribe to our Newsletter

Submit Blogs on Medium

FEATURE STORE SUMMIT 2025

UP NEXT:

From Real-Time ML to Agents with Hopsworks

Jim Dowling

CEO & Co-Founder

Hopsworks

Organized by HOPSWORKS

From Real-Time ML to Agents with Hopsworks

Jim Dowling, CEO, Hopsworks

Organized by HOPSWORKS

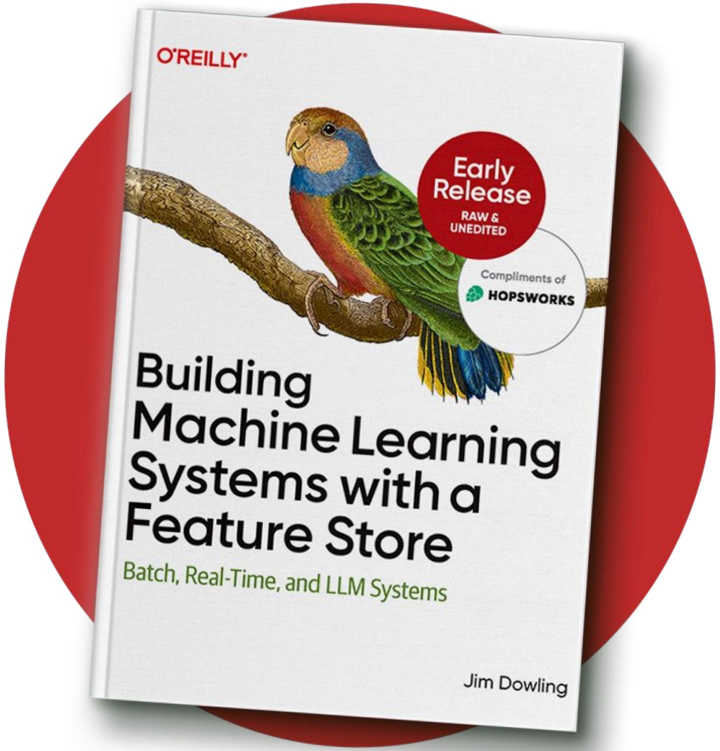
AGENDA

- **Feature Store Architecture** (Lakehouse First vs Real-Time First)
- **Shift Left vs Shift Right** Data Transformations
  - Shift-Right: Pushdown Aggregations in RonDB
  - Shift-Left: Rolling Aggregations with Incremental Computation
- **Data Models** for Feature Stores: Snowflake Schema beats Star Schema
- **Real-Time Context Engineering** with a feature store

Building Machine Learning Systems

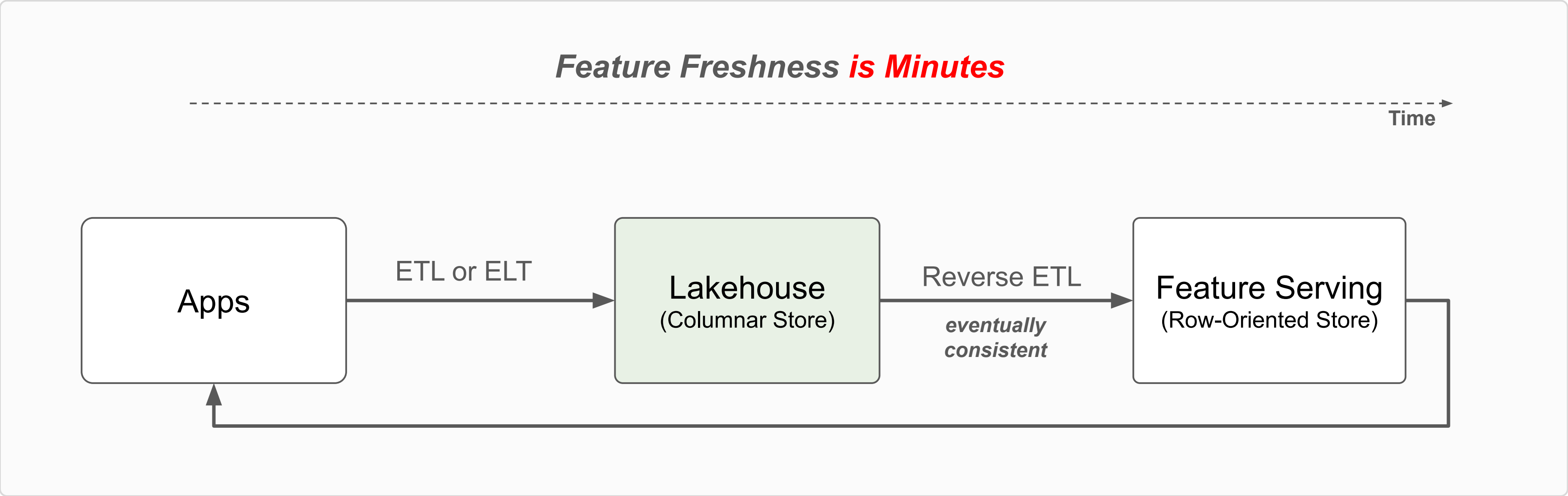
Batch, Real-Time, and LLM Systems

Scan the QR code and add a Promo Code:
‘FS2025’ to reserve your spot
to receive the complete digital edition

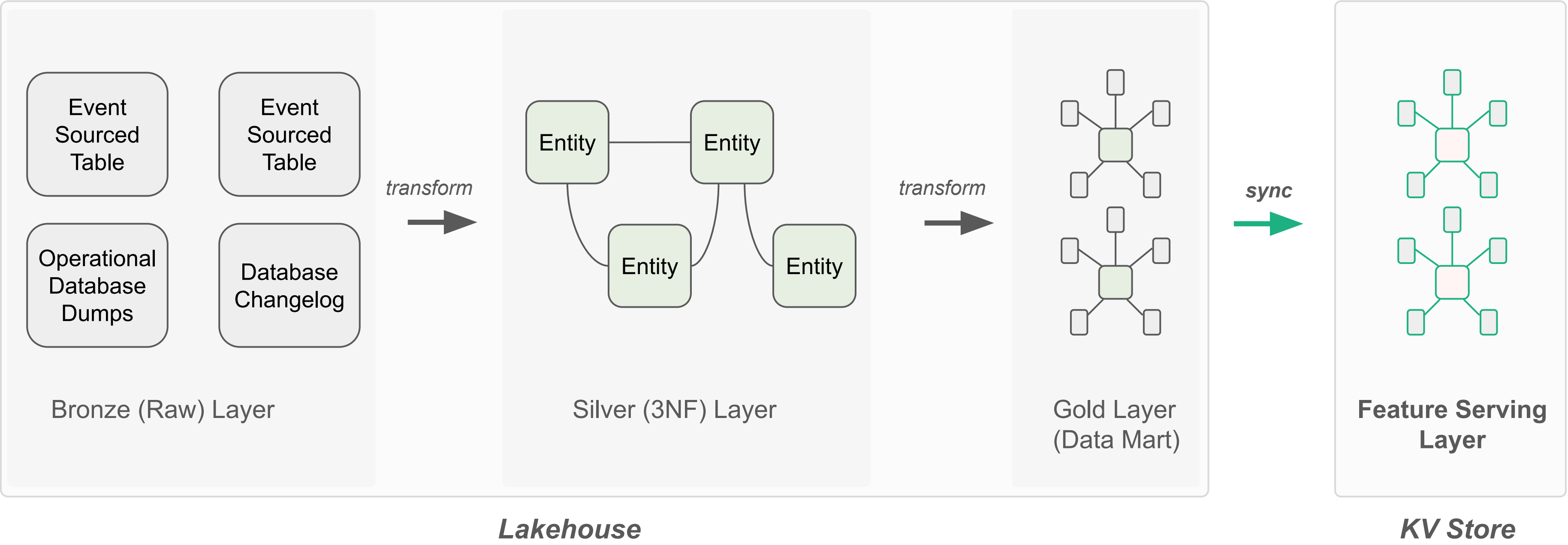


Feature Store Architectures

Lakehouse-First Feature Store Architecture (Databricks/Vertex)



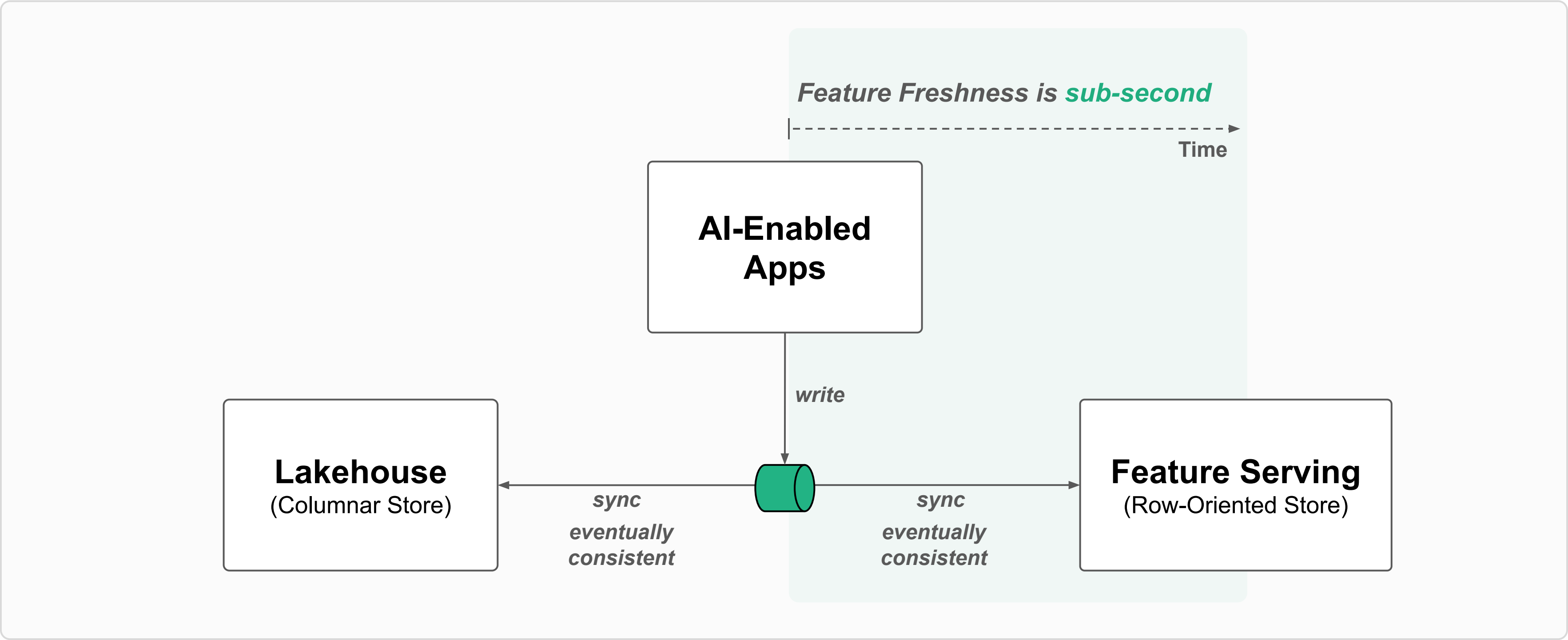
Lakehouse to Feature Serving. The Data model is Star Schema*.



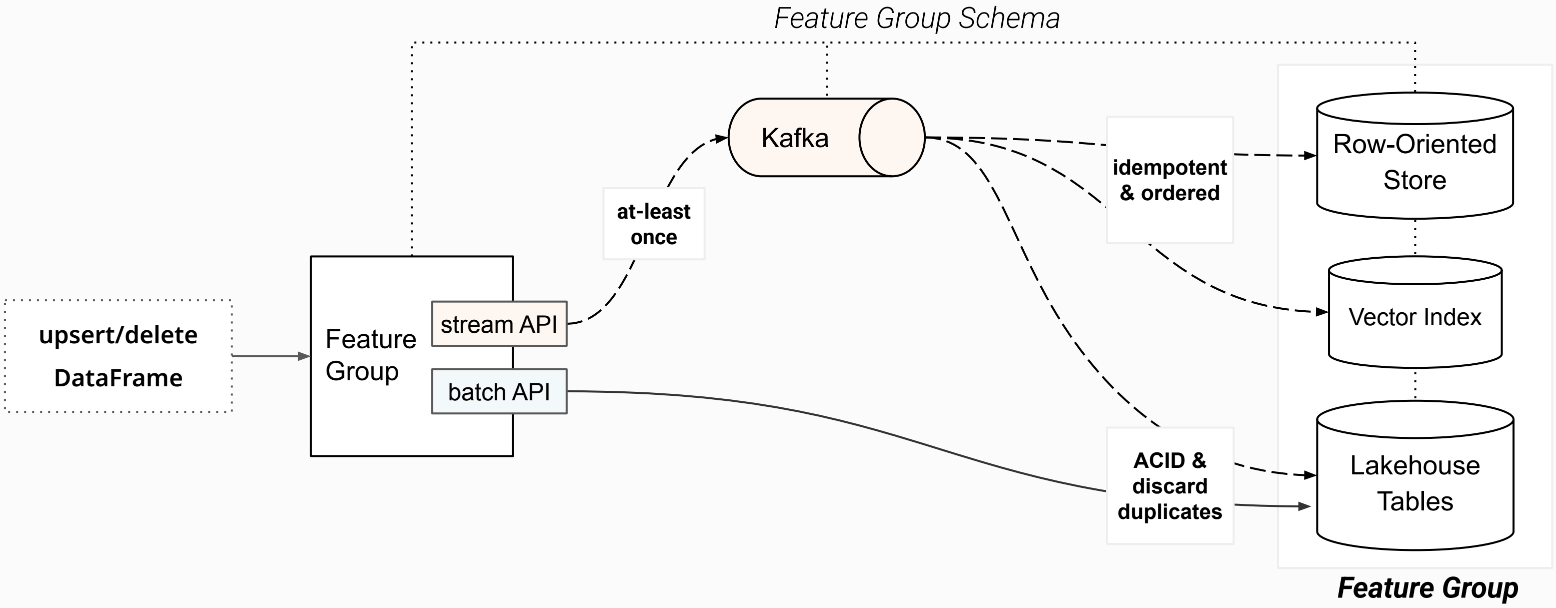
Chapter 4: Feature Stores
Chapter 10: Training Pipelines

*Star Schema can increase complexity of real-time AI systems.

Hopsworks: Real-Time First Feature Store Architecture



Hopsworks Feature Store Architecture



Idempotent and atomic updates ensure **consistent data** between offline and online stores*.

*Martinez et al, “The Hopsworks Feature Store for Machine Learning”, SIGMOD 2024

Data Transformations: Shift Left or Shift Right

FEATURE STORE SUMMIT 2025

Organized by HOPSWORKS

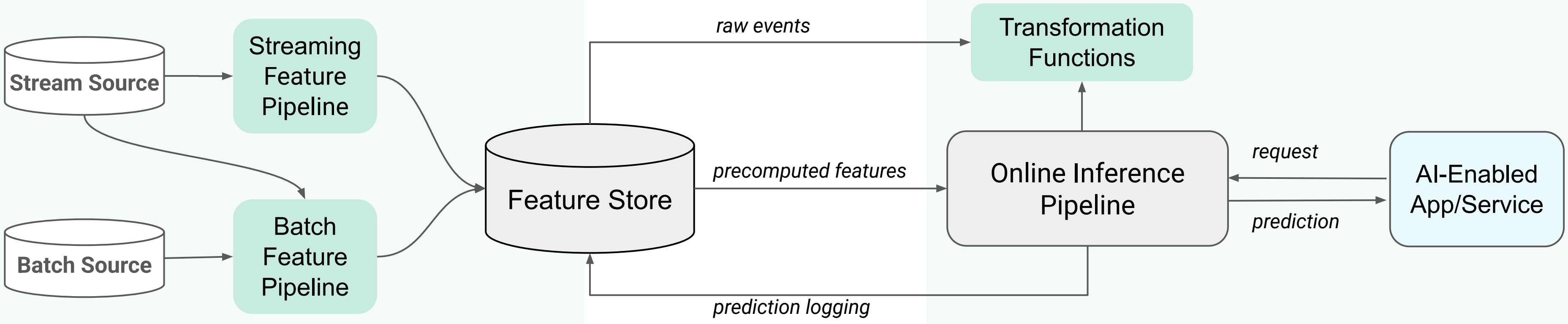
Hopsworks supports the widest set of Shift Left and Shift Right Transformations

Small/Large Scale, More Operational Overhead.
Huge Choice of Open-Source Engines.
No Offline-Online Skew.

Ease-of-Use, Limited Scalability.
Mostly Closed Source Engines.
Offline-Online Skew.

Shift Left
(precomputed features)

Shift Right
(on-demand features)

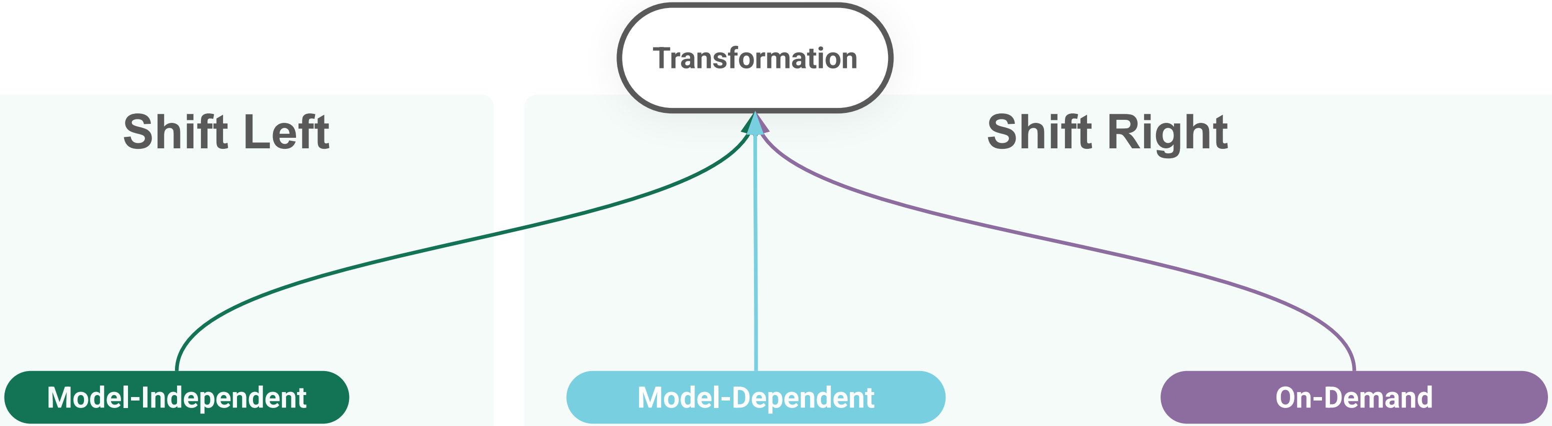


Chapter 8: Batch Feature Pipelines
Chapter 9: Real-Time Features

Shift-Left: Model-Independent Data Transformations
Shift-Right: Model-Dependent & On-Demand Data Transformations

Shift Left

Shift Right



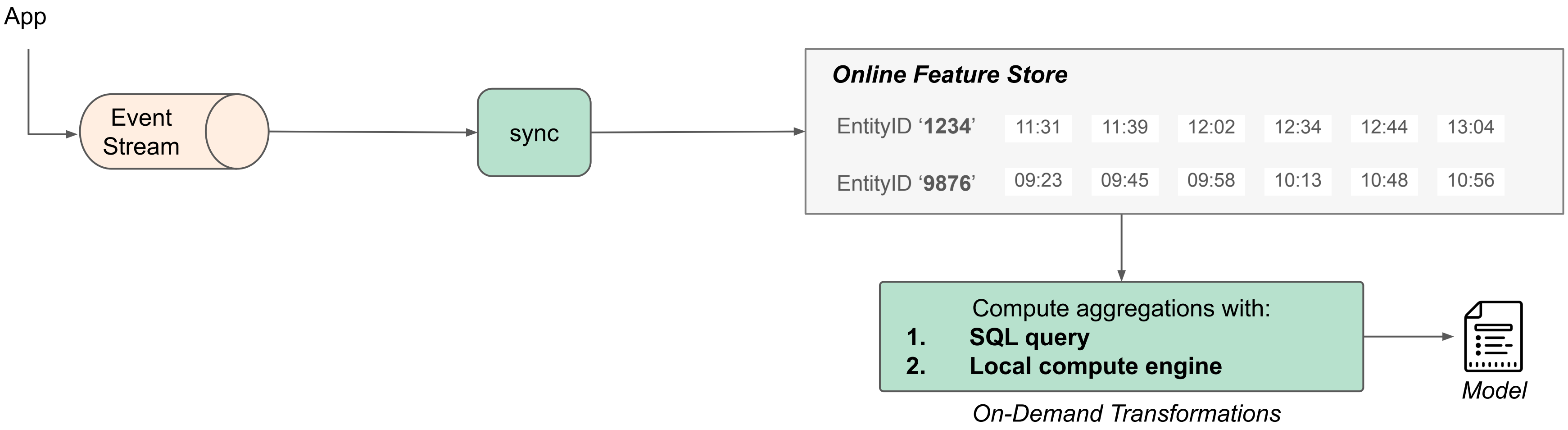
Data engineering transformations - aggregations, windowed count, RFM. Use the best compute framework for your needs.
Produces: **Reusable Features**.

Transformations specific to a model. Feature encoding, scaling, imputation are parameterized by a model's training data.
Produces: **Model-Specific Features**.

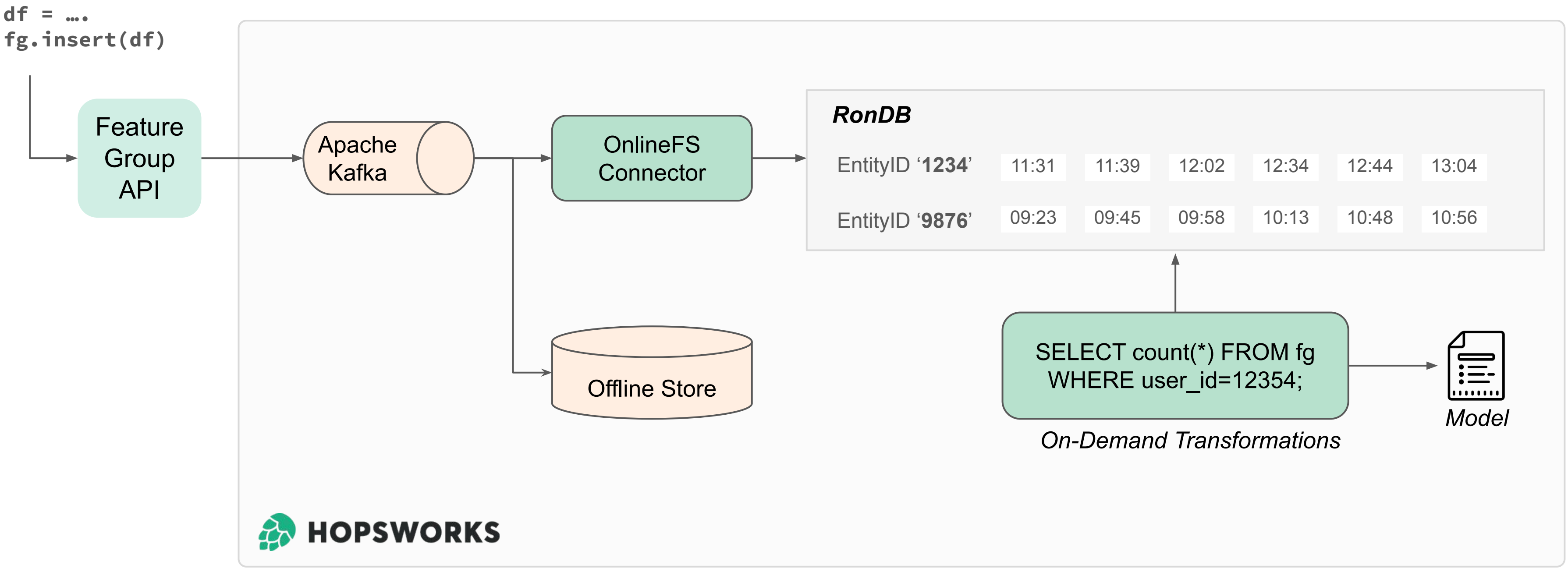
Need request-time data to be computed online. Can also be applied to historical data to precompute features (backfill) in feature pipelines.
Produces: **On-Demand Features**.

Chapter 2: The FTI Architecture
Chapter 6: Model-Independent Transformations
Chapter 7: Model-Dependent / On-Demand

Shift Right: Compute Real-Time Features On-Demand



Shift Right: Compute Real-Time Features with Pushdown Aggregations in Hopsworks

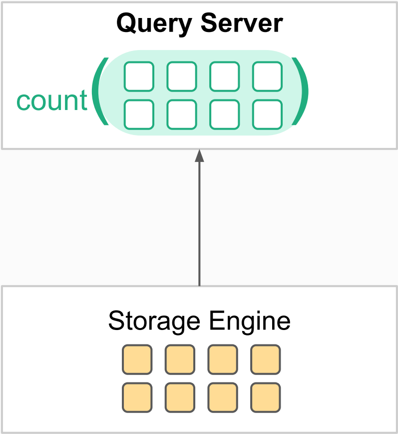


Aggregations in (1) RDBMS, (2) Feature Engine (3) RonDB Pushdown Aggregations

Throughput ❌
Latency ❌

(1) RDBMS (e.g., MySQL)

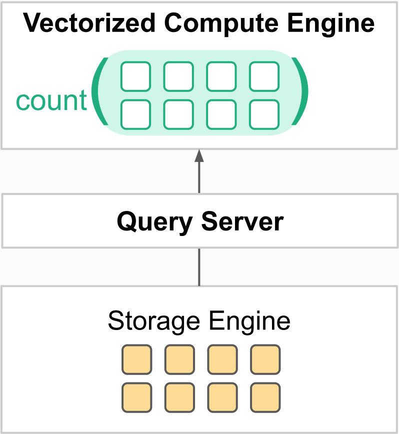
```
SELECT page, count(*) FROM
pageviews GROUP BY page
```



Throughput ✅
Latency ❌

(2) Real-Time Feature Engine

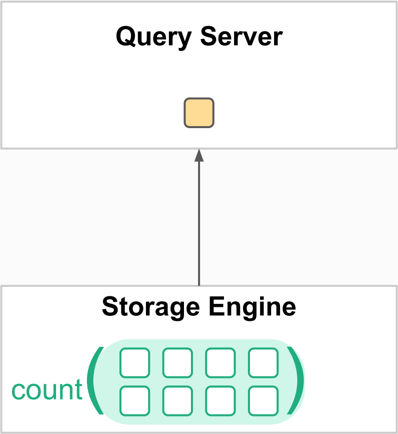
```
Aggregation(groupBy=page,
agg=count, schedule='daily', ..)
```



Throughput ✅
Latency ✅

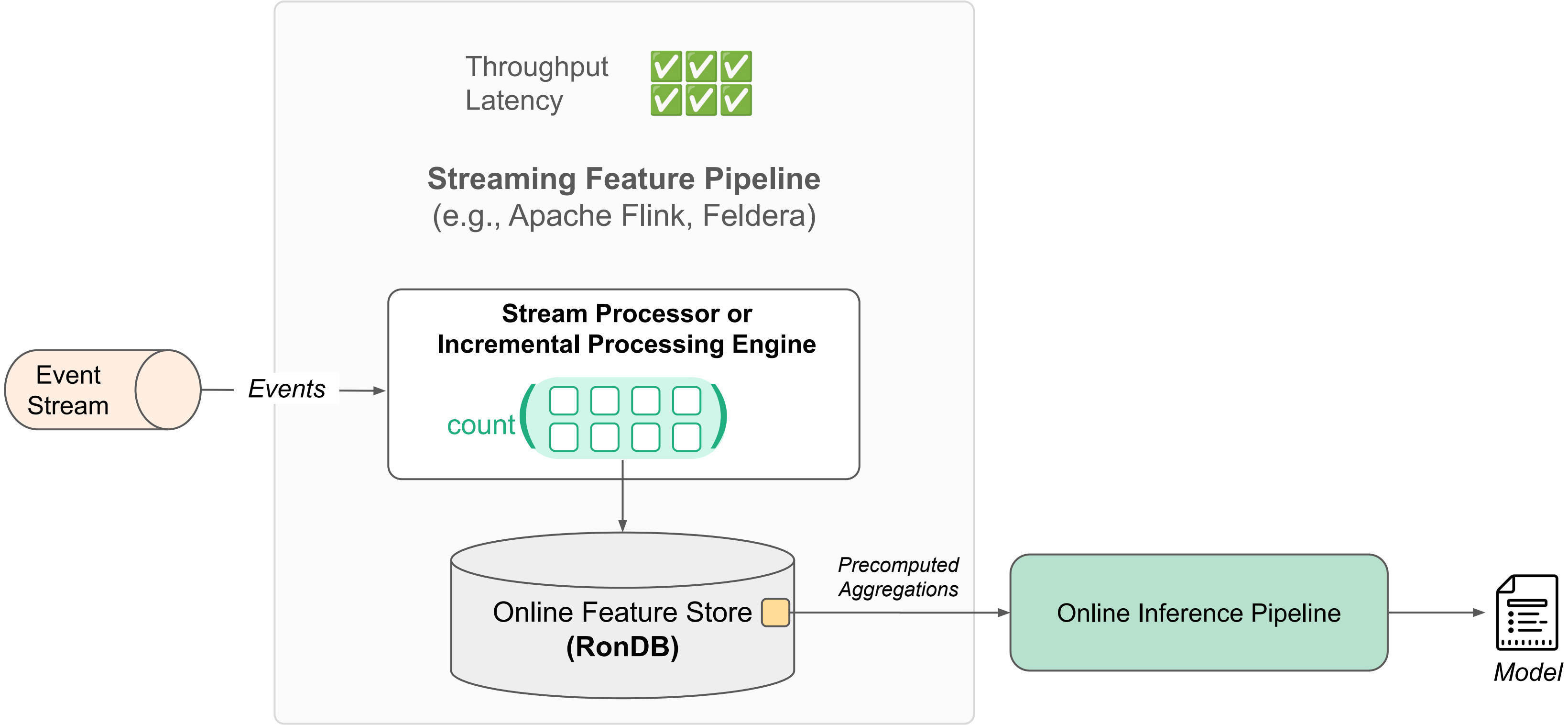
(3) RonDB Pushdown Aggregations*

```
SELECT page, count(*) FROM
pageviews GROUP BY page
```



*Up to **50X latency reduction** for RonDB compared to MySQL

Shift Left to Streaming for Higher Throughput and Lower Latency



Rolling Aggregations - the Queen of Real-Time Aggregated Features

Rolling aggregation
Window size: 1hour
Aggregation function: SUM

Time

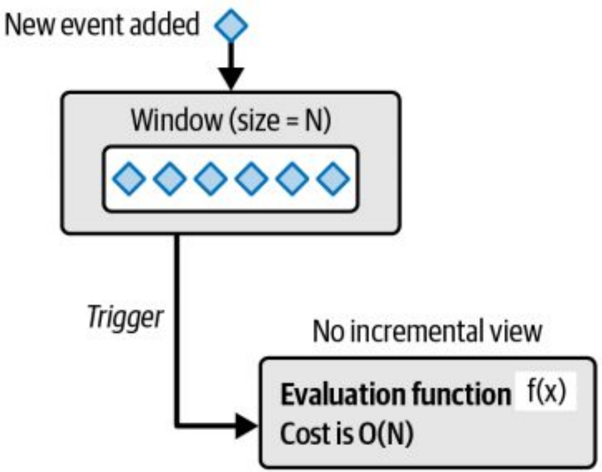
cc_num	event_time	amount	1hour_sum
1234 5678 9012 3456	0h 1m	$30.95	$30.95
1234 5678 9012 3456	0h 3m	$1.99	$32.94
1234 5678 9012 3456	0h 7m	$11.99	$44.93
...			
1234 5678 9012 3456	0h 43m	$21.00	$607.98
1234 5678 9012 3456	0h 52m	$98.95	$628.98
1234 5678 9012 3456	0h 57m	$113.99	$727.93
1234 5678 9012 3456	1h 2m	$10.00	$841.92
1234 5678 9012 3456	1h 7m	$44.95	$845.87

SUM=$845.87
over last 60 mins

Suboptimal Alternatives

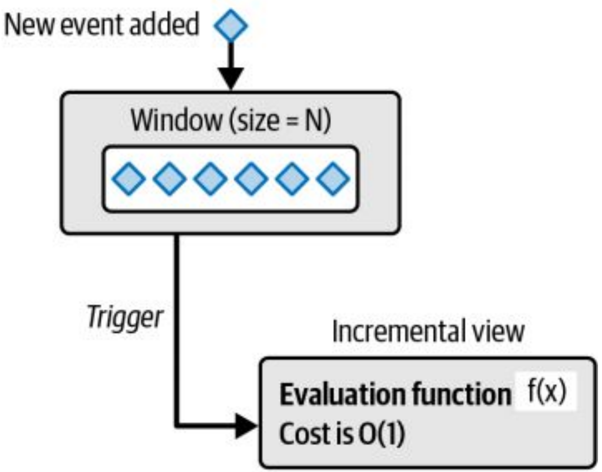
- Sliding Window Aggregations
  - Feature freshness == slide length
- Tiled Time-Window Aggregations
  - Higher latency at online inference
  - Still requires a streaming pipeline

Incremental View Maintenance reduces Computational Complexity for Rolling Aggregations



Full recompute of f(x) over all N events for every new event (Apache Spark Streaming, Apache Flink, etc.)

Apache Flink O(N)



Incremental view recomputes only new events with the evaluation function (Feldera)

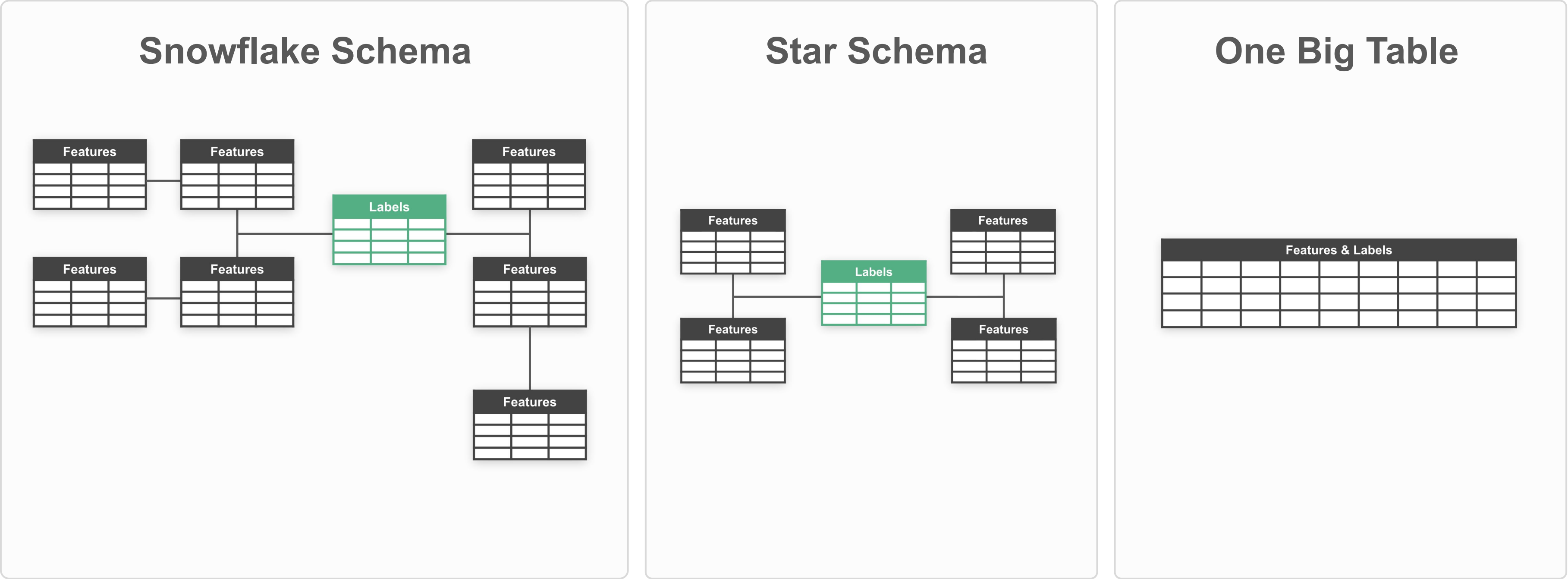
Feldera O(1)

Data Modelling for Feature Stores

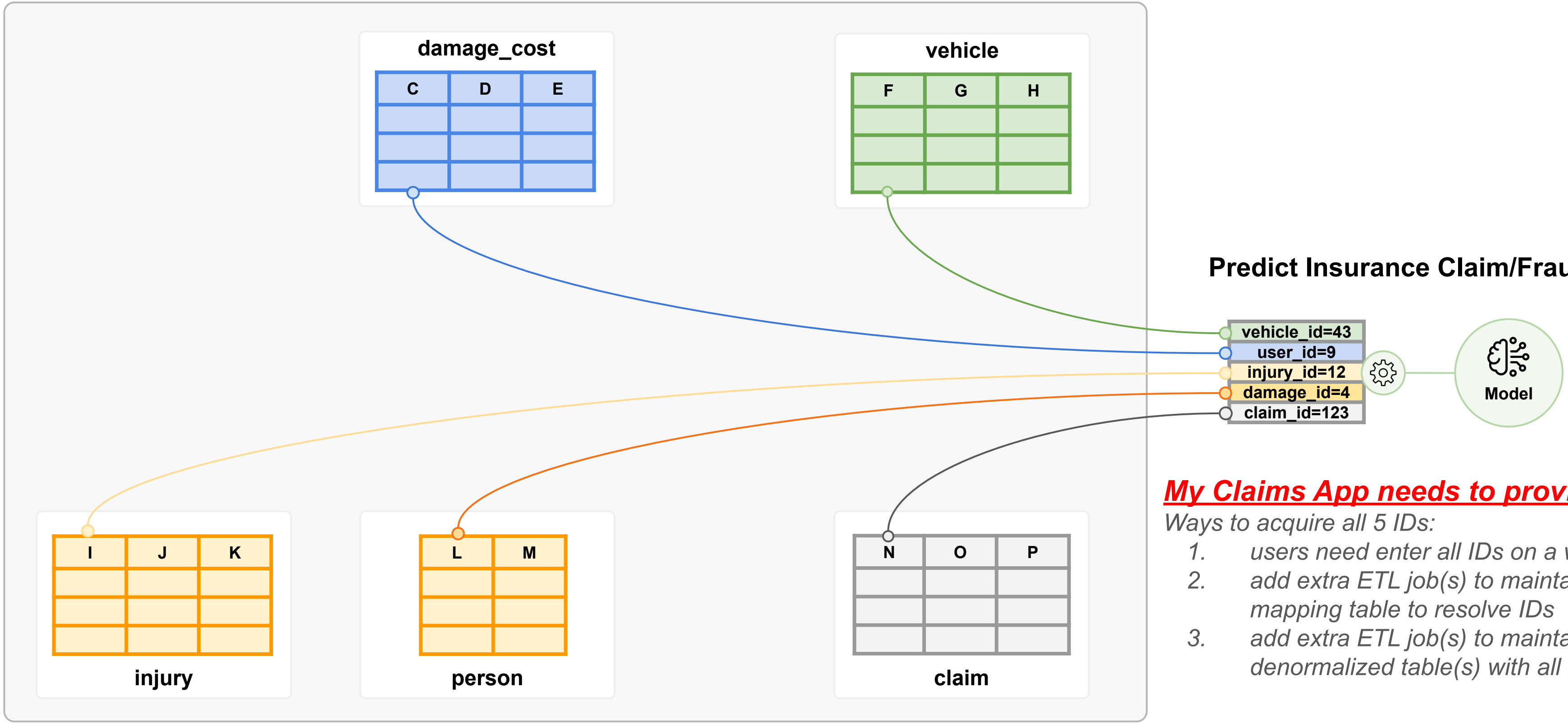
FEATURE STORE SUMMIT 2025

Organized by HOPSWORKS

Data Models for Feature Stores



Limitations of Star Schema Data Model for Insurance Claims Use Case

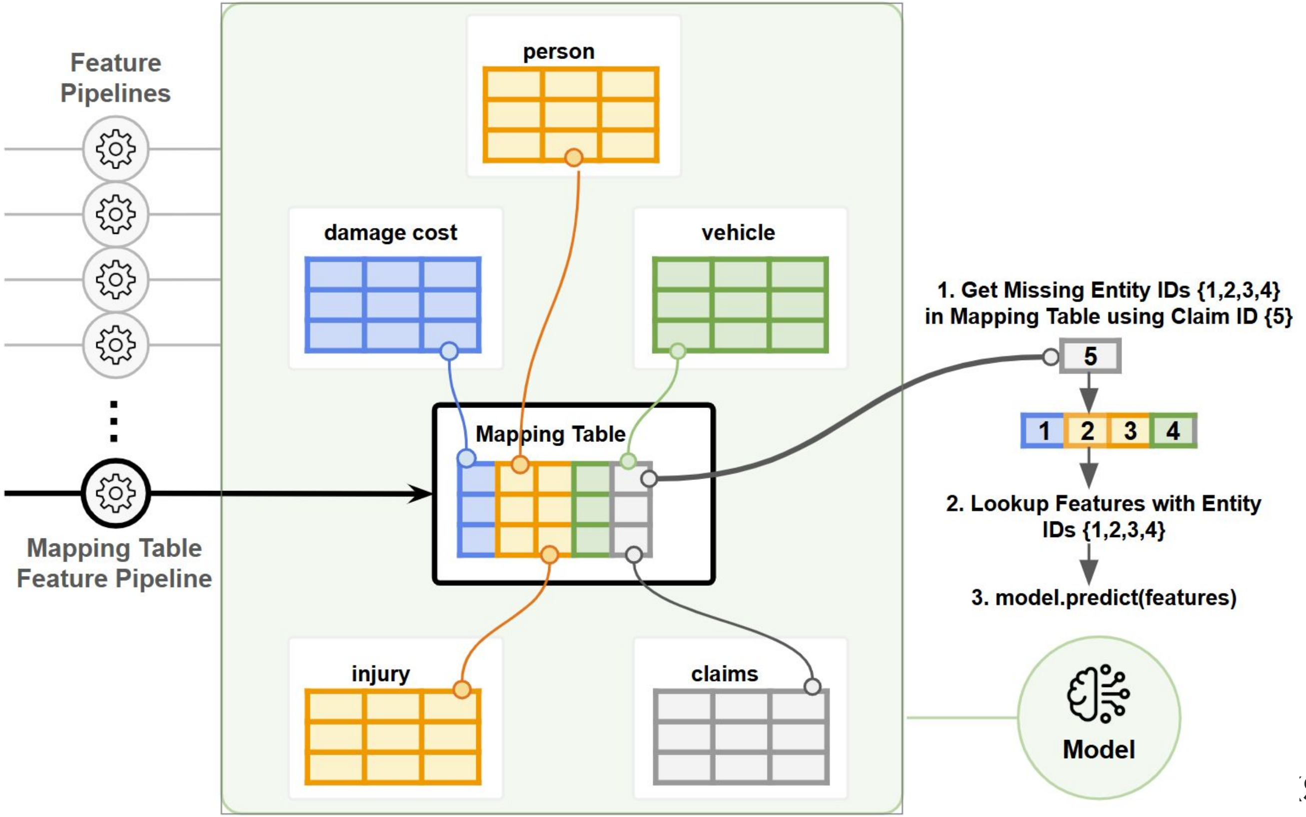


My Claims App needs to provide 5 IDs

Ways to acquire all 5 IDs:

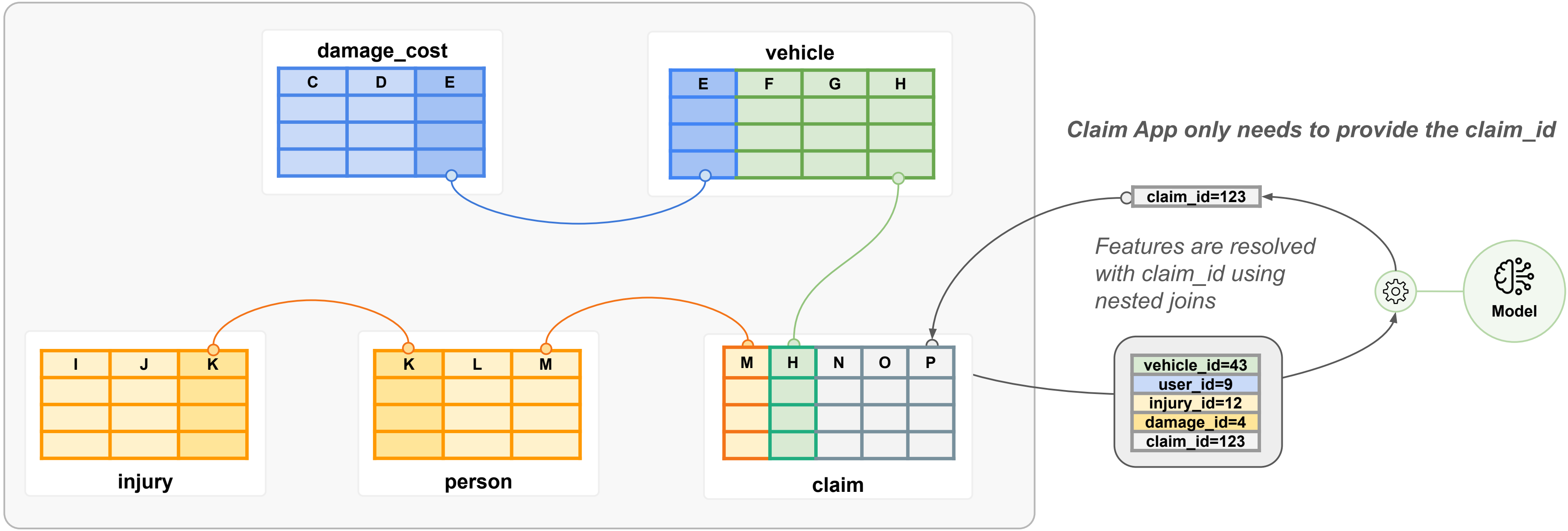
1. *users need enter all IDs on a webpage*
2. *add extra ETL job(s) to maintain a mapping table to resolve IDs*
3. *add extra ETL job(s) to maintain denormalized table(s) with all features*

“Simplify” Claims App so users only need to provide the claim_id



How can we avoid adding a Mapping Table and its pipeline?

Snowflake Schema Data Model solves the Mapping Table Problem*



*Online feature store needs to support nested JOINs

Snowflake Schema in Hopsworks

FEATURE STORE SUMMIT 2025

```python
vehicle_fg, damage_cost_fg, claim_fg, person_fg, injury_fg = fs.get_feature_group(...)
```

Python

```python
person_subtree = person_fg.select_features() \
        .join(injury_fg.select_features())
vehicle_subtree = vehicle_fg.select_features() \
        .join(damage_cost_fg.select_features())
```

Python

Feature Selection

```python
all_features = claim_fg.select_features() \
        .join(person_subtree) \
        .join(vehicle_subtree)
```

Python

```python
fv = fs.create_feature_view(name="claims_fv", version=1,
                query=all_features,
                labels=['is_fraud']
)
```

Python

Snowflake Schema in Hopsworks

```python
# Model Training
X_train, X_test, y_train, y_test = fv.train_test_split(test_size=0.2)
model.train(X_train, y_train)
mr = fs.get_project().get_model_registry()
fraud_model = mr.python.create_model(name="claims_fraud",
    metrics=evaluation_dict,
    feature_view=feature_view,
)
mr.save_dir("dir with serialized model")
```

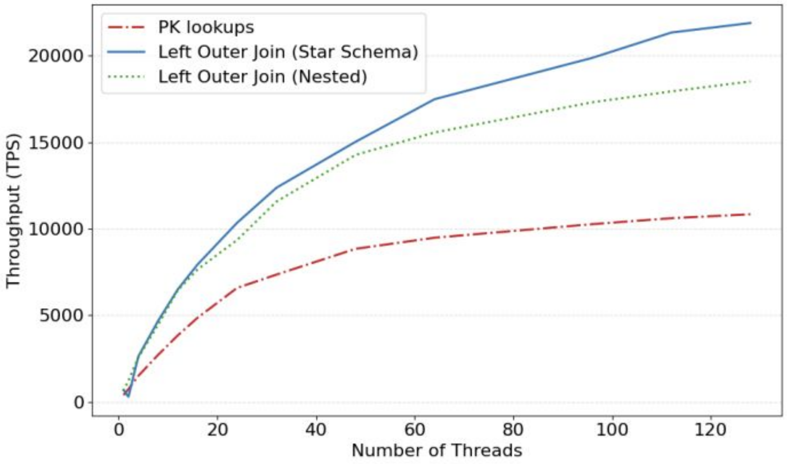
Python

Pushdown Left JOIN

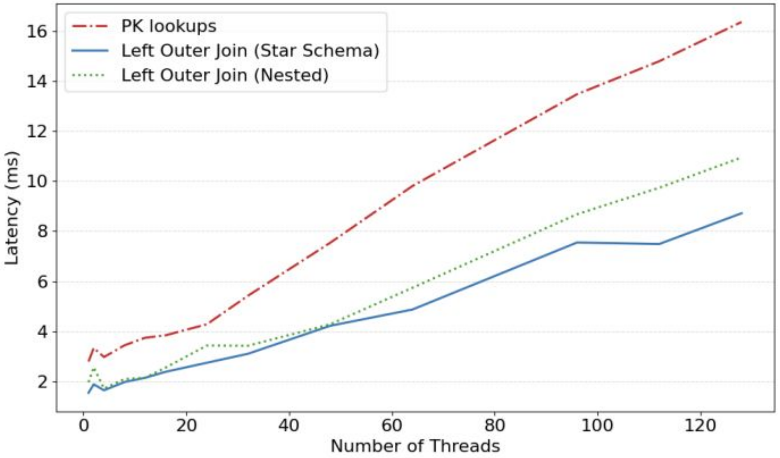
```python
# Model Deployment and Online Inference
feature_vector = fv.get_feature_vector(entry={"claim_id": 1234})
prediction = model.predict(feature_vector)
fv.log(feature_vector, prediction)
```

Python

Pushdown Left JOINs on RonDB have better performance*



(a) Throughput for Primary Key Read, Left Joins (Star Schema and Nested)

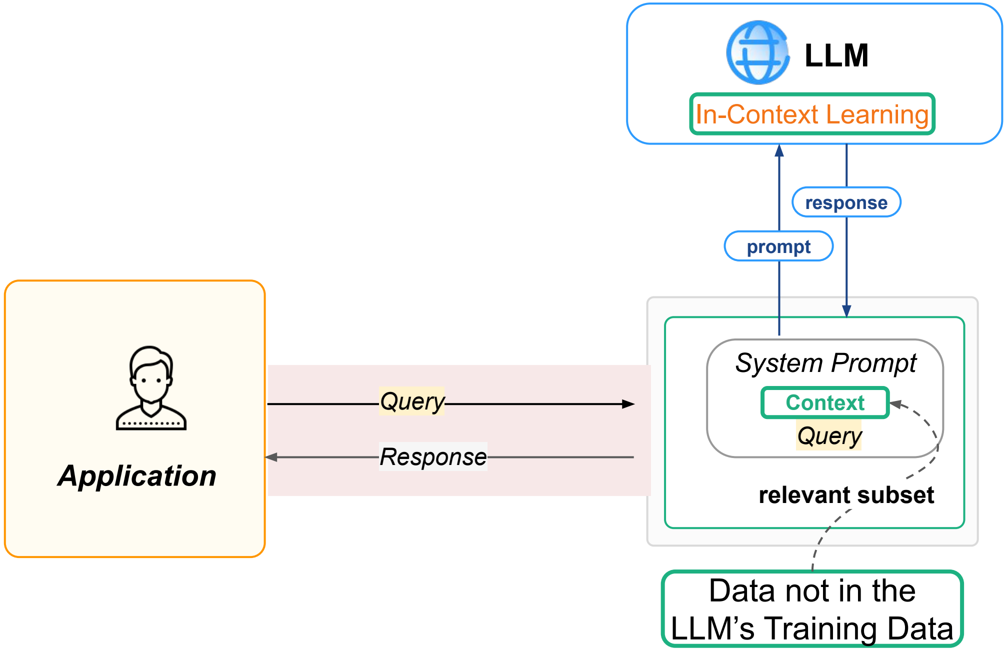


(b) Latency for Primary Key Read, Left Joins (Star Schema and Nested)

*Martinez et al, “The Hopsworks Feature Store for Machine Learning”, SIGMOD 2024

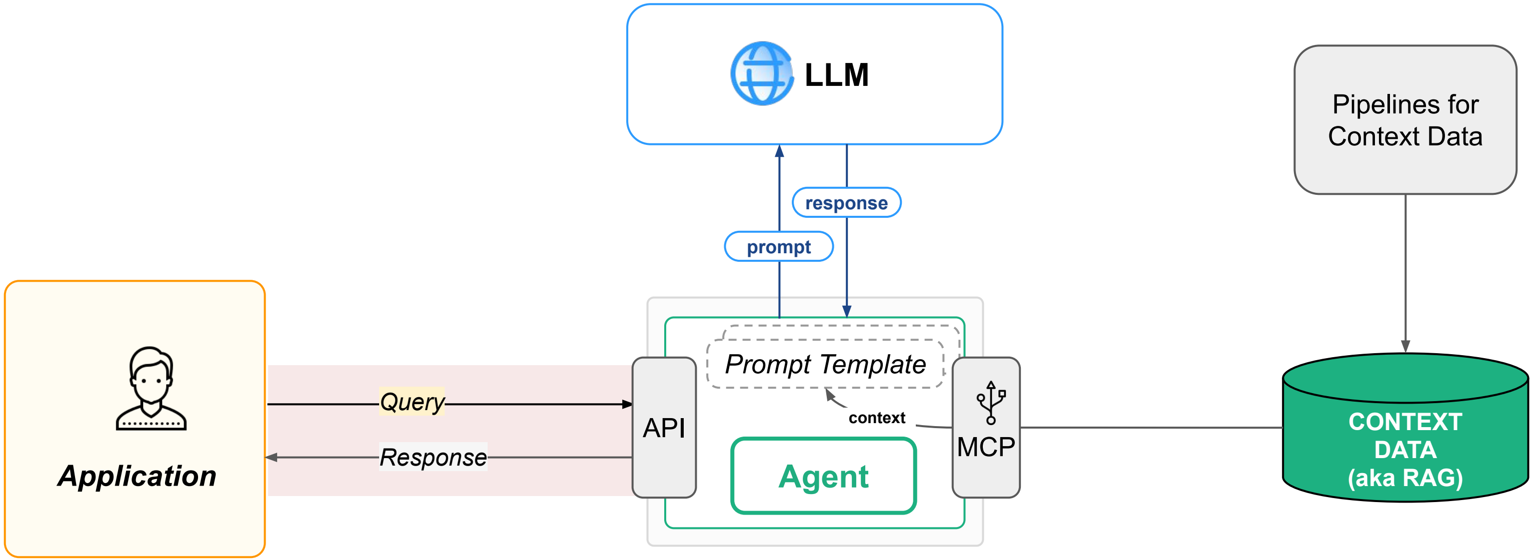
From Real-Time Features to Context

In-Context Learning



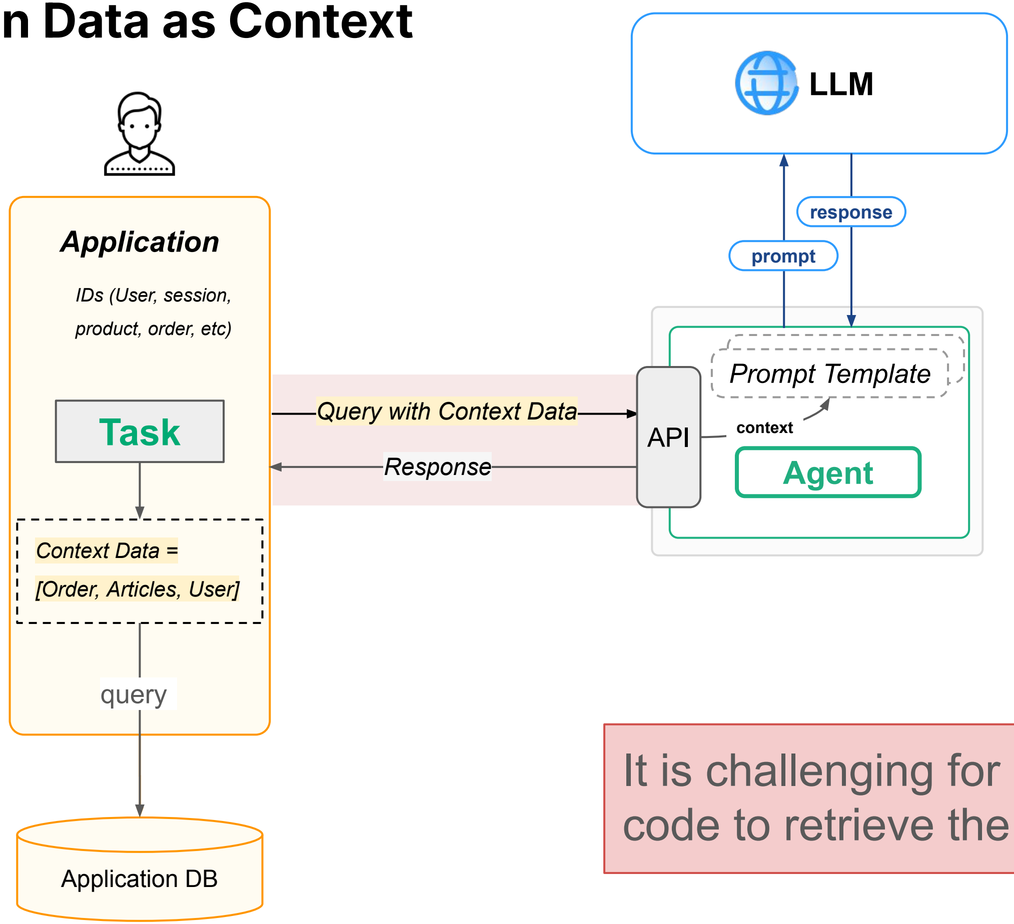
In-Context Learning is the dominant paradigm for incorporating new data in LLMs, lording over the alternative approach of fine-tuning open-source foundation LLMs with your new data.

Context Data for Agents



Context Data is primarily (1) private data and (2) recent data (post LLM training cutoff date)

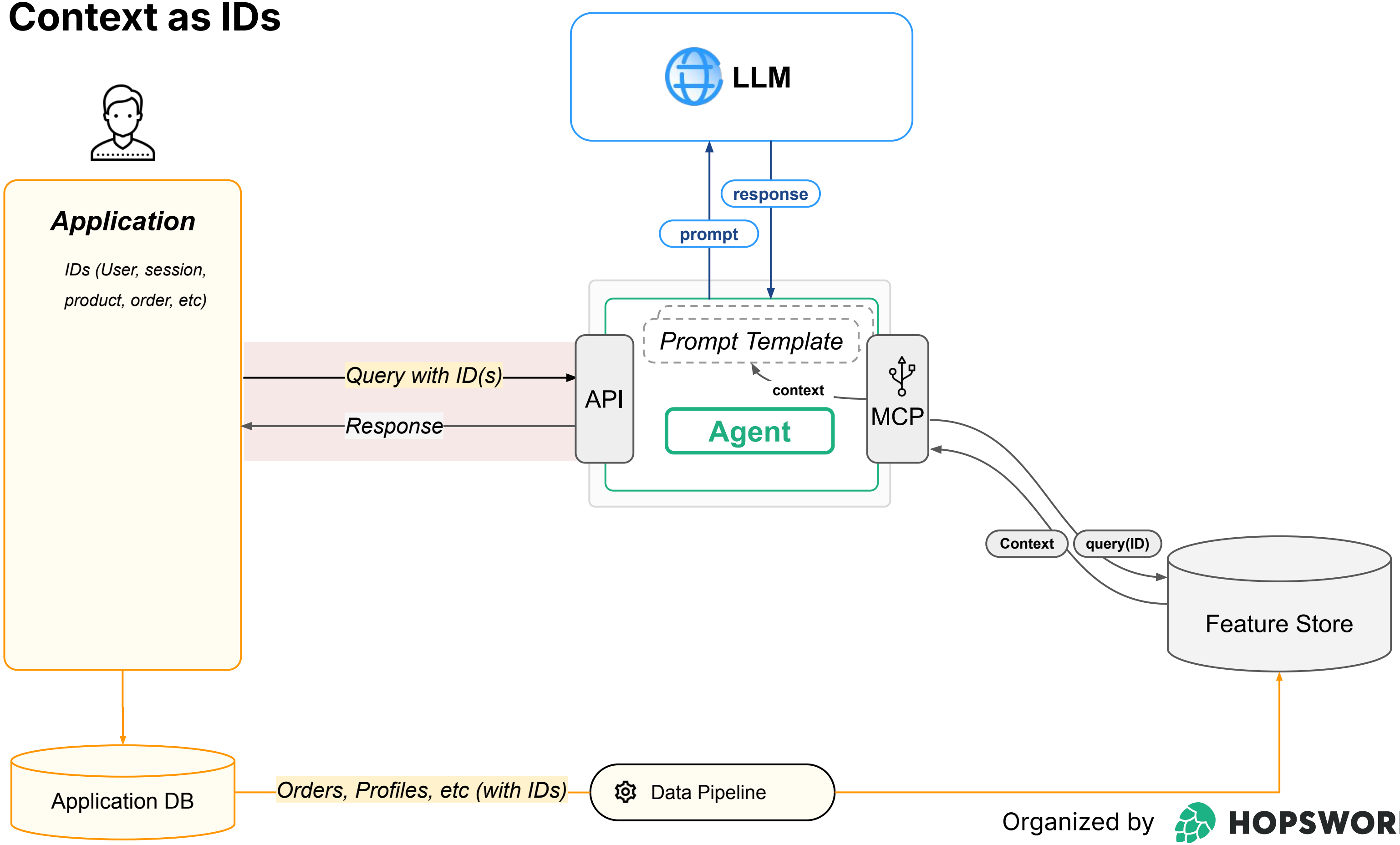
Application Data as Context



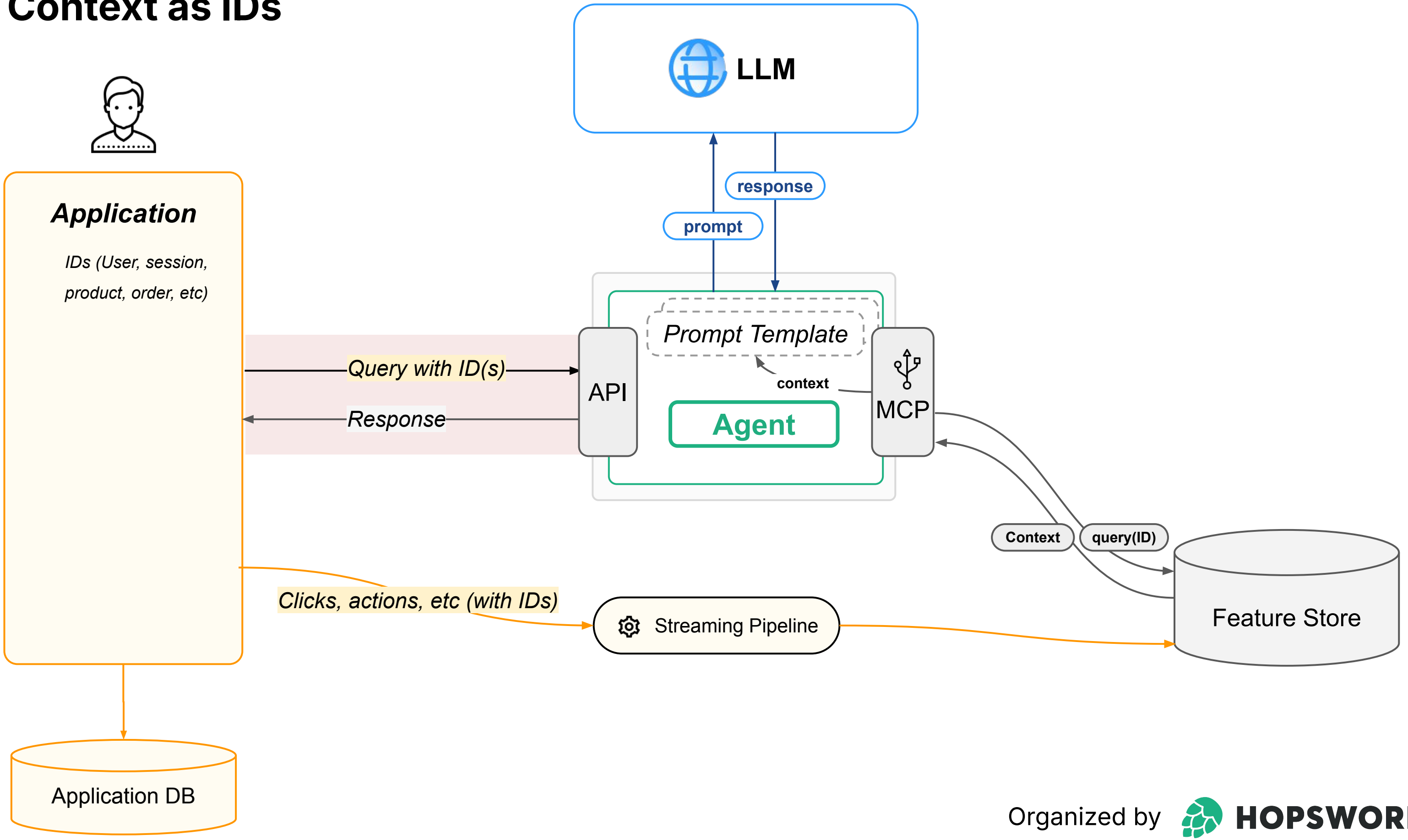
RAG: Information Retrieval Method
It is the application developer's job to retrieve and send context data to the agent.

It is challenging for app developers to write code to retrieve the correct context data!

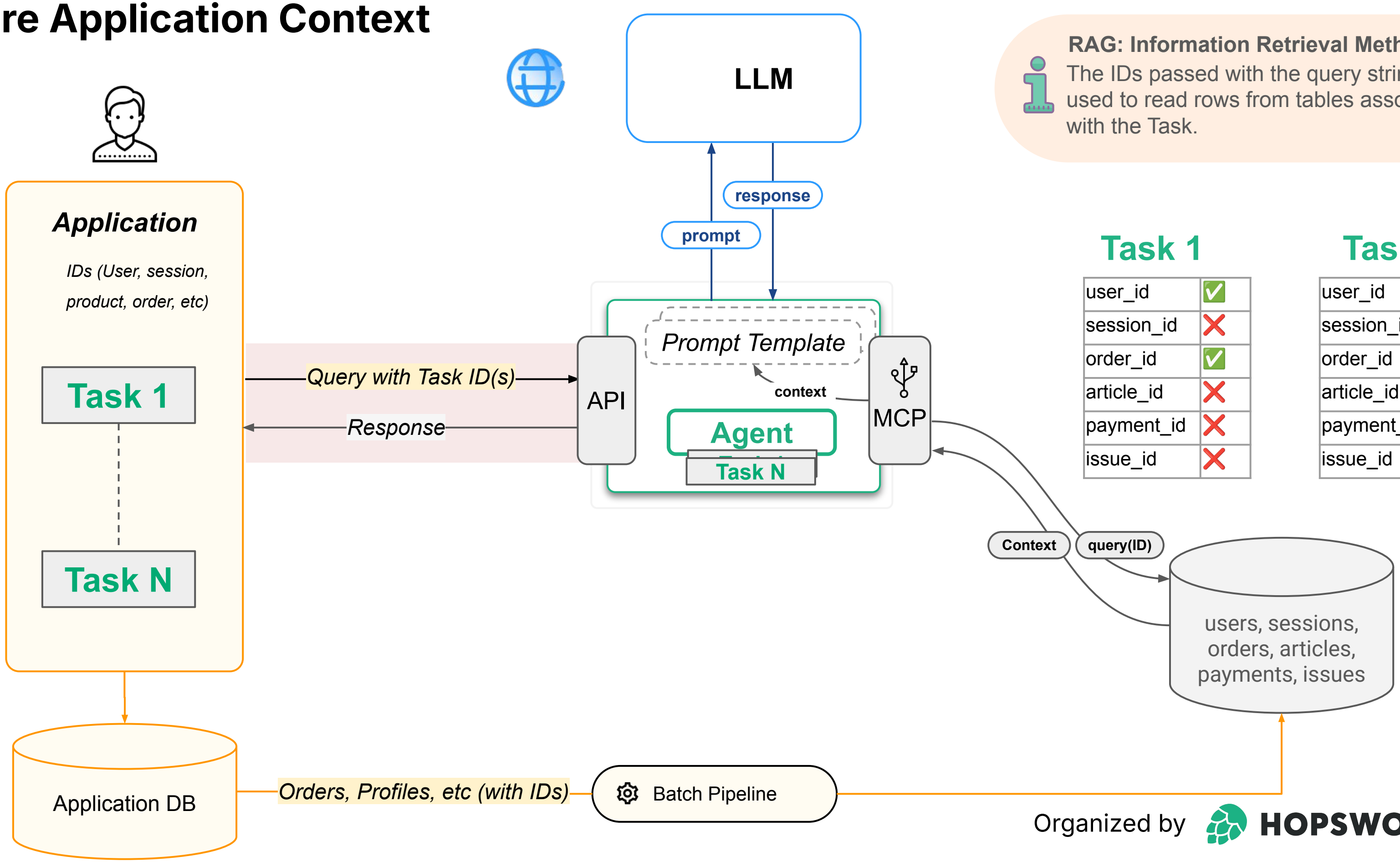
Application Context as IDs



Application Context as IDs



Task-Aware Application Context



RAG: Information Retrieval Method
The IDs passed with the query string are used to read rows from tables associated with the Task.

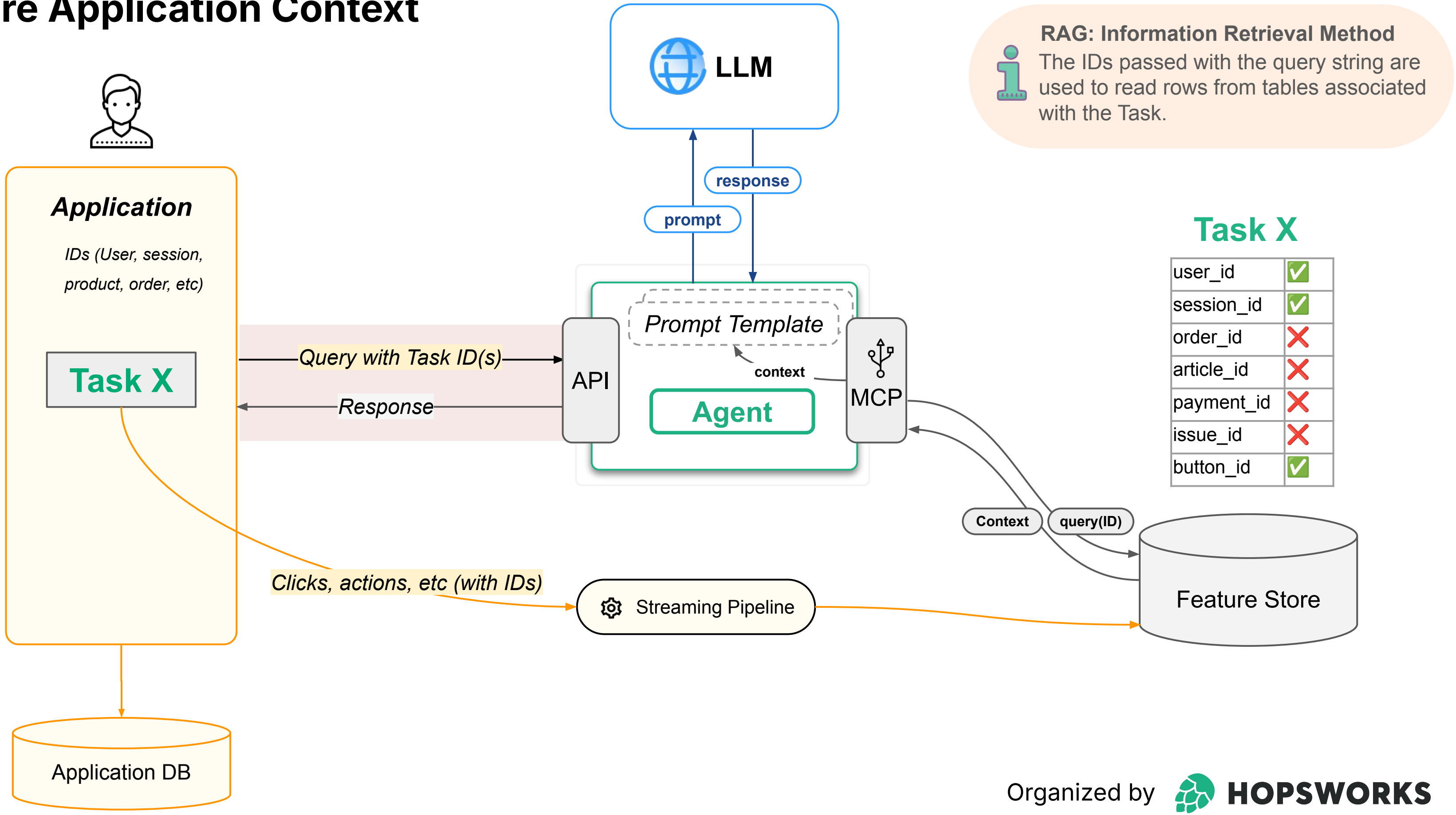
Task 1

user_id	✅
session_id	❌
order_id	✅
article_id	❌
payment_id	❌
issue_id	❌

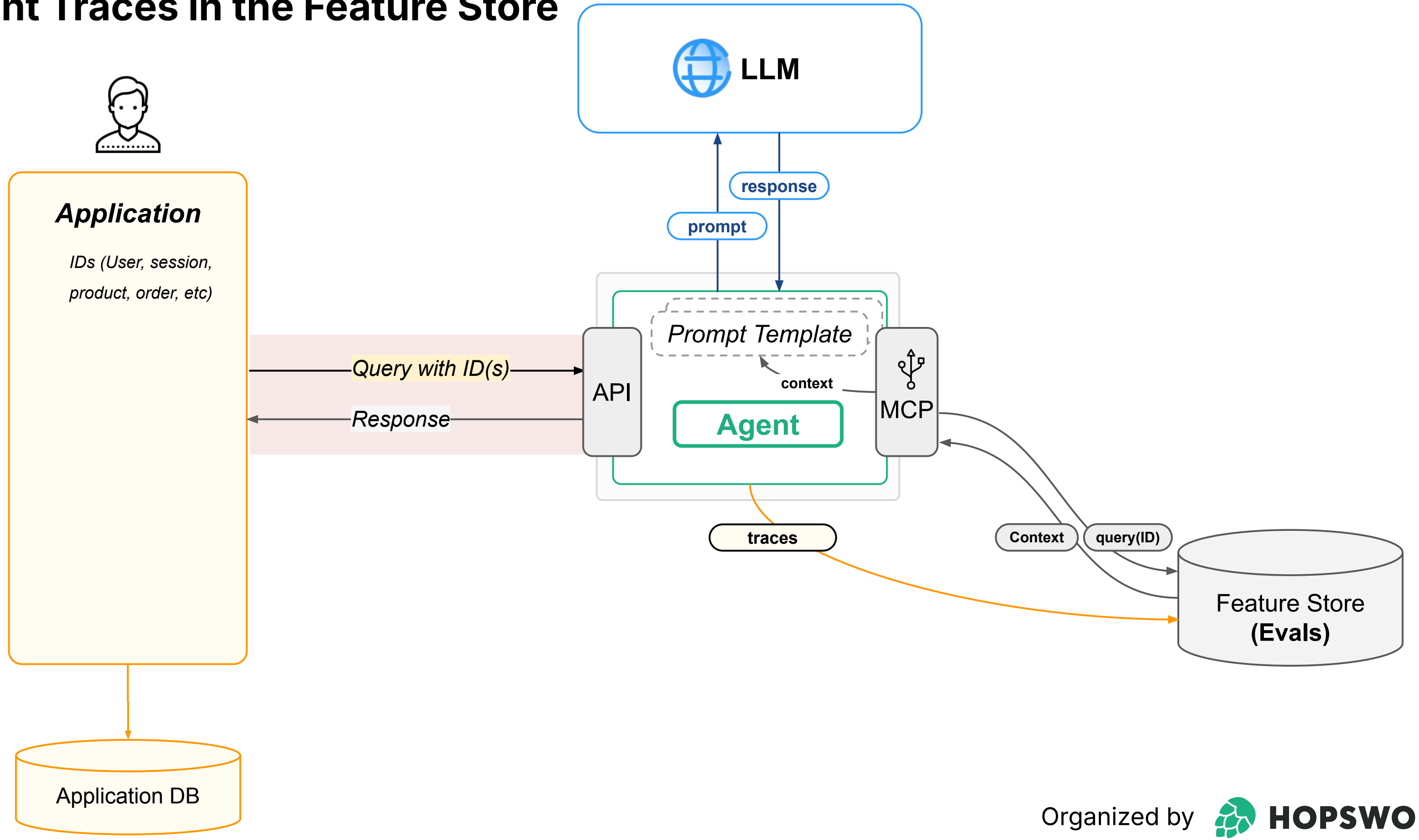
Task N

user_id	✅
session_id	❌
order_id	❌
article_id	✅
payment_id	❌
issue_id	❌

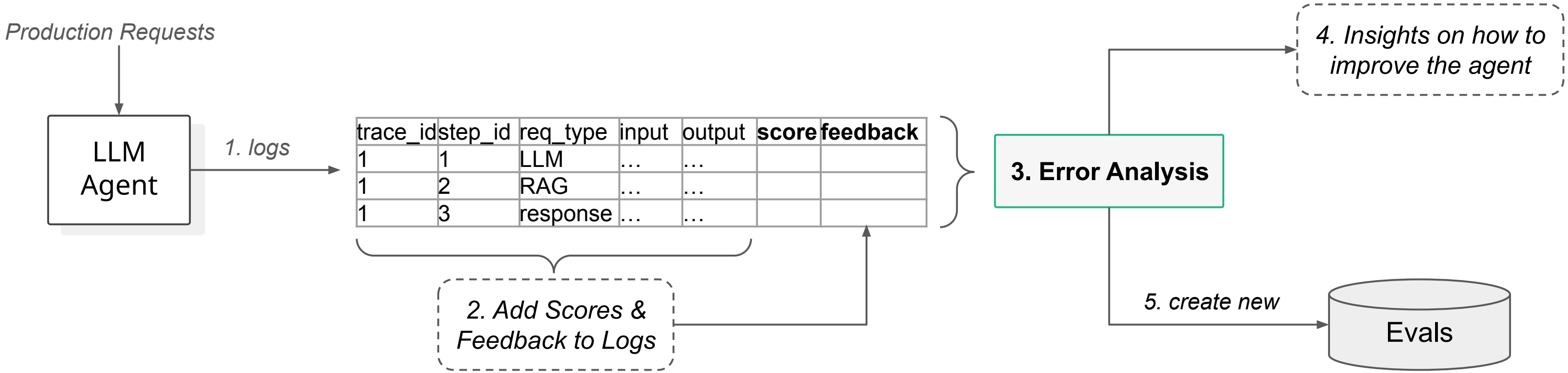
Task-Aware Application Context



Store Agent Traces in the Feature Store



Error Analysis with Agent Log Traces



trace_id	step_id	req_type	input	output	**score**	**feedback**
1	1	LLM	…	…		
1	2	RAG	…	…		
1	3	response	…	…		

Objective Evals and Subjective Evals

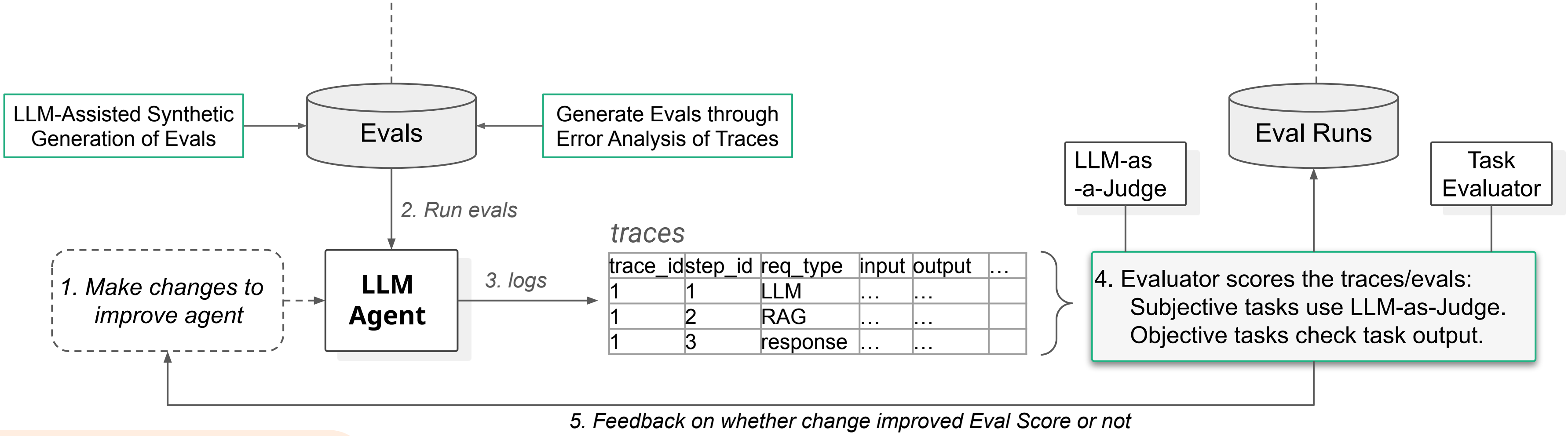
Automated Testing: Evaluate Changes to Agents with Evals

evals

eval_id	event_ts	task	prompt	expected_response
1	2025-03-17 10:00	feature pipeline	Write a feature pipeline for the Titanic Dataset	Created feature group with 891 rows.
..	…	…	…	…

eval_runs

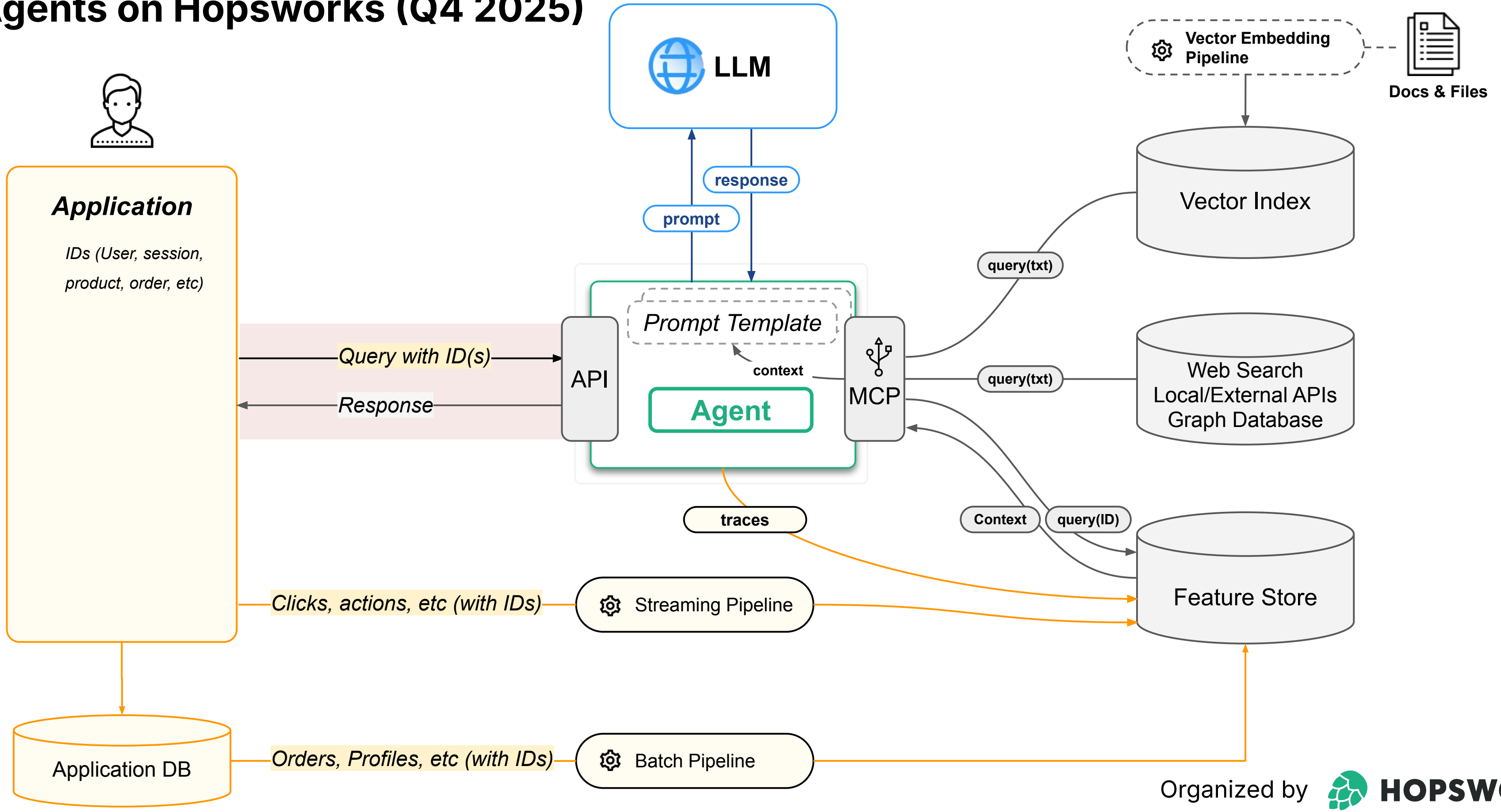
eval_id	event_ts	response	feedback	score
1	2025-03-18 10:00	Created feature group with 891 rows.	feature group creation worked.	5
..	…	…	…	…



traces

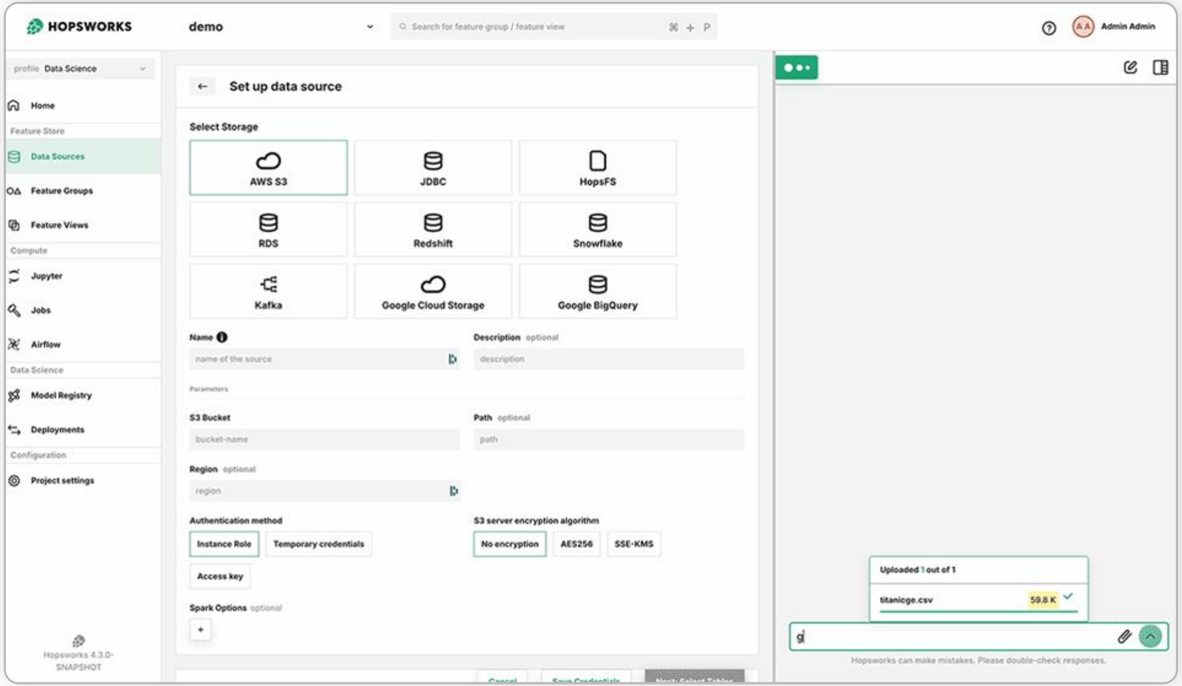
trace_id	step_id	req_type	input	output	…
1	1	LLM	…	…	
1	2	RAG	…	…	
1	3	response	…	…	

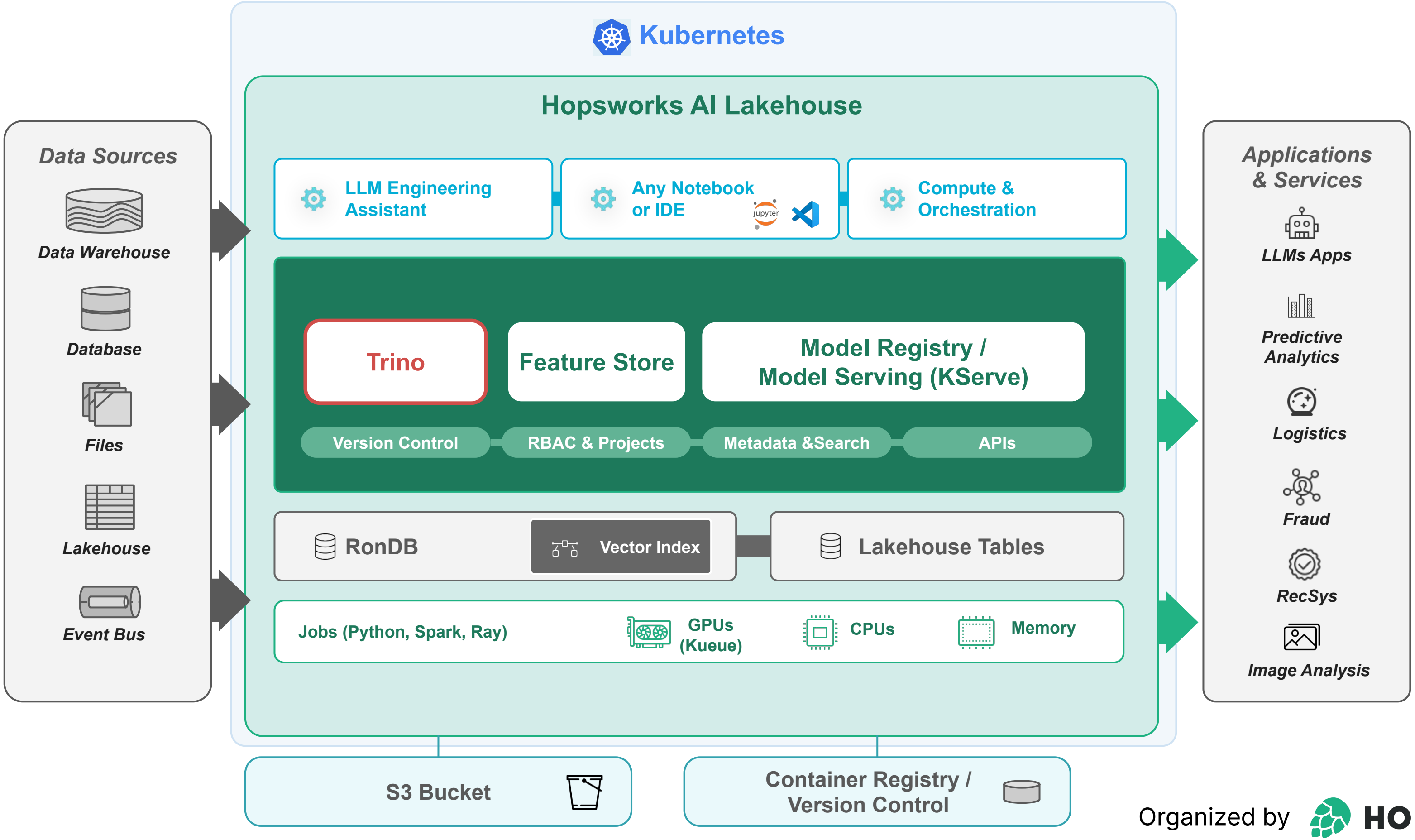
Agents on Hopsworks (Q4 2025)



What else is new in Hopsworks?

Brewer (Q4 2025)




Kubernetes
Hopsworks AI Lakehouse
Data Sources
Data Warehouse
Database
Files
Lakehouse
Event Bus
LLM Engineering Assistant
Any Notebook or IDE
jupyter
Compute & Orchestration
Trino
Feature Store
Model Registry / Model Serving (KServe)
Version Control
RBAC & Projects
Metadata &Search
APIs
RonDB
Vector Index
Lakehouse Tables
Jobs (Python, Spark, Ray)
GPUs (Kueue)
CPUs
Memory
S3 Bucket
Container Registry / Version Control
Applications & Services
LLMs Apps
Predictive Analytics
Logistics
Fraud
RecSys
Image Analysis

Building Machine Learning Systems

Batch, Real-Time, and LLM Systems

Scan the QR code and add a Promo Code: 'FS2025' to reserve your spot to receive the complete digital edition