Lyft's Feature Store: Architecture, Optimization, and Evolution

Rohan Varshney, Senior Software Engineer

FEATURE STORE SUMMIT 2025

Organized by HOPSWORKS

Lyft's Feature Store, a core infrastructure component in its Data Platform, **optimizes the management & deployment of ML features at scale**. It centralizes feature engineering & ensures uniformity across models & workflows by streamlining feature creation & storage for both offline/online model training & inference, facilitating low-latency access & high-throughput processing. This presentation covers its architecture, practical uses, performance, developer experience, optimization efforts, & evolution over the last 5+ years. We hope to demonstrate its role in empowering Lyft engineers to **develop service components & models more effectively, including for future AI/LLM applications**.

Use Cases

Team / Function	Description	Impact
Fulfillment	Houses ML models to match drivers to rides & generate + rank rider offers.	Empowers dispatch and offering services.
Orchestration		
Pricing		
Integrity		
Growth Platform & Market Expansion		

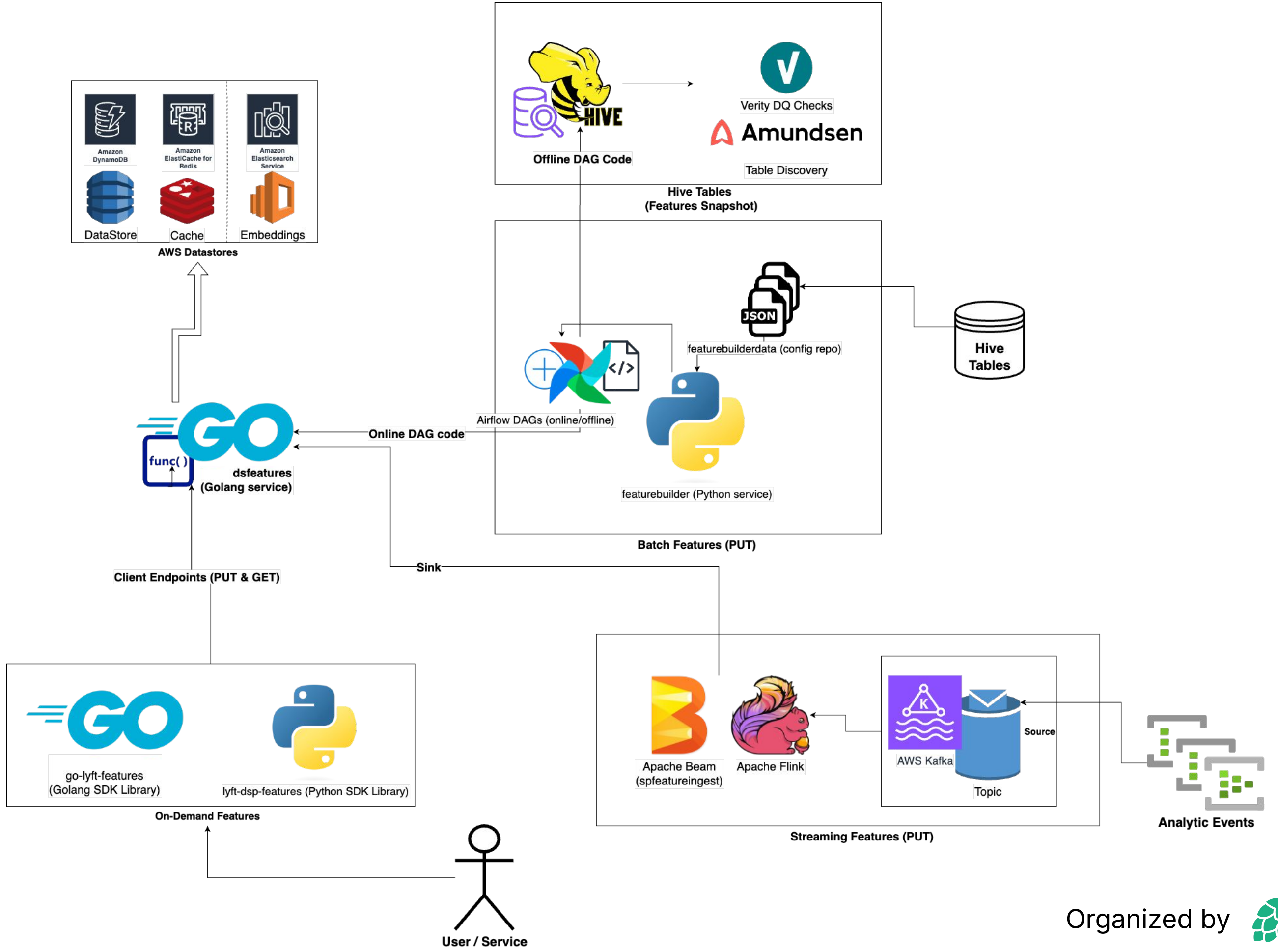
Team / Function	Description	Impact
Fulfillment	Houses ML models to match drivers to rides & generate + rank rider offers.	Empowers dispatch and offering services.
Orchestration	Platform for rider & driver incentive programs. Launch ML models for batch campaigns.	Produce incremental driver hour engagement & growth.
Pricing		
Integrity		
Growth Platform & Market Expansion		

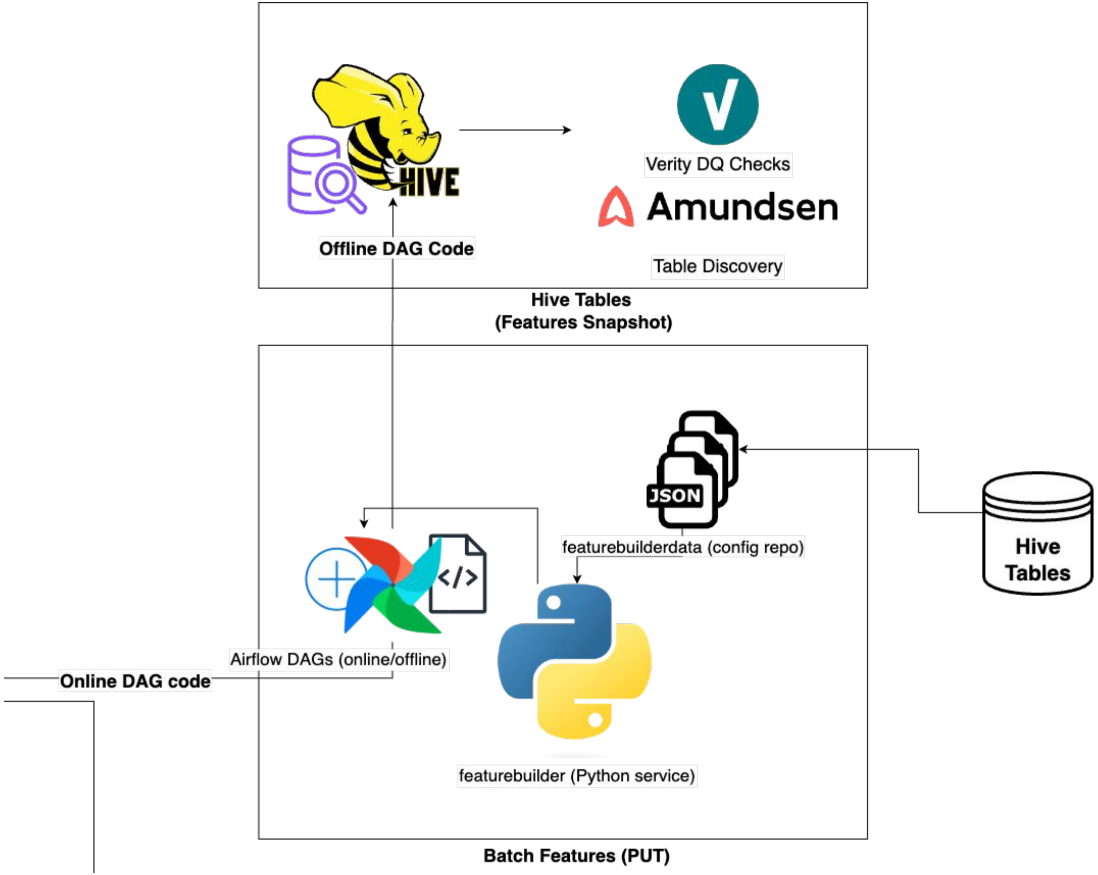
Team / Function	Description	Impact
Fulfillment	Houses ML models to match drivers to rides & generate + rank rider offers.	Empowers dispatch and offering services.
Orchestration	Platform for rider & driver incentive programs. Launch ML models for batch campaigns.	Produce incremental driver hour engagement & growth.
Pricing	ML models that generate prices, coupons, etc.	💰💰💰
Integrity		
Growth Platform & Market Expansion		

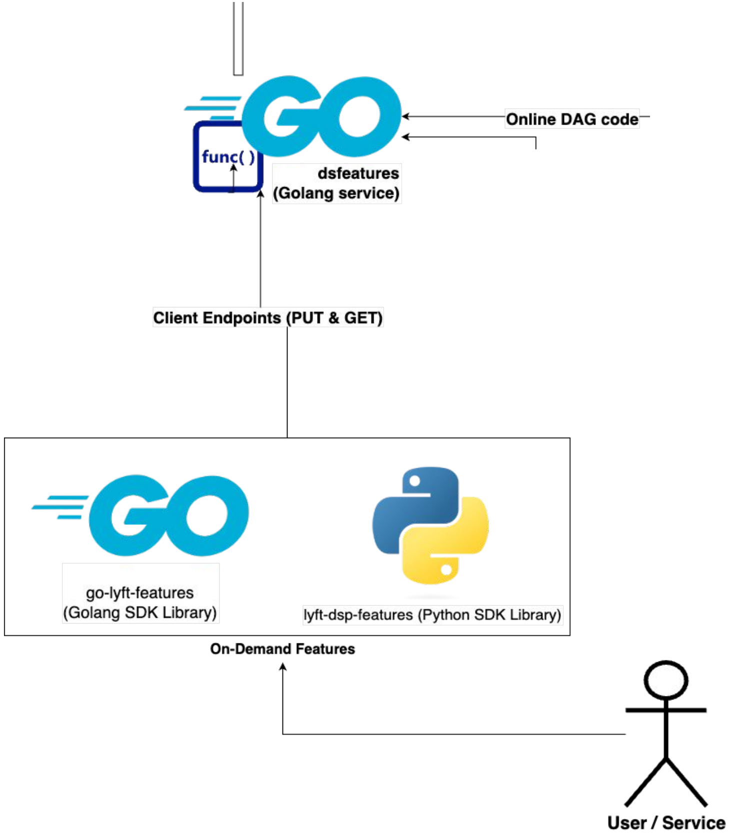
Team / Function	Description	Impact
Fulfillment	Houses ML models to match drivers to rides & generate + rank rider offers.	Empowers dispatch and offering services.
Orchestration	Platform for rider & driver incentive programs. Launch ML models for batch campaigns.	Produce incremental driver hour engagement & growth.
Pricing	ML models that generate prices, coupons, etc.	💰💰💰
Integrity	Fraud detection platform.	Potential blocker of fraudulent activity, real-time and retroactive.
Growth Platform & Market Expansion		

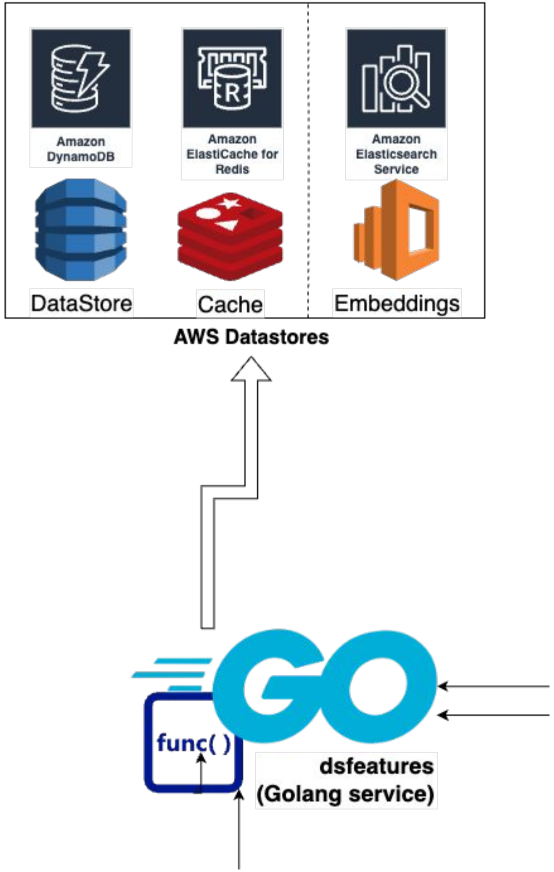
Team / Function	Description	Impact
Fulfillment	Houses ML models to match drivers to rides & generate + rank rider offers.	Empowers dispatch and offering services.
Orchestration	Platform for rider & driver incentive programs. Launch ML models for batch campaigns.	Produce incremental driver hour engagement & growth.
Pricing	ML models that generate prices, coupons, etc.	💰💰💰
Integrity	Fraud detection platform.	Potential blocker of fraudulent activity, real-time and retroactive.
Growth Platform & Market Expansion	Comms platform, building “audiences” for targeting marketing messages.	Growth of rider & driver engagement, + incremental ride count & revenue.

Architecture


Amazon DynamoDB
Amazon ElastiCache for Redis
Amazon Elasticsearch Service
DataStore
Cache
Embeddings
AWS Datastores
HIVE
Offline DAG Code
Verity DQ Checks
Amundsen
Table Discovery
Hive Tables
(Features Snapshot)
JSON
featurebuilderdata (config repo)
Hive Tables
Airflow DAGs (online/offline)
featurebuilder (Python service)
Batch Features (PUT)
Online DAG code
func( )
dsfeatures
(Golang service)
Sink
Client Endpoints (PUT & GET)
go-lyft-features
(Golang SDK Library)
lyft-dsp-features (Python SDK Library)
On-Demand Features
Apache Beam
(spfeatureingest)
Apache Flink
AWS Kafka
Source
Topic
Streaming Features (PUT)
Analytic Events
User / Service


HIVE
Verity DQ Checks
Amundsen
Offline DAG Code
Table Discovery
Hive Tables
(Features Snapshot)
JSON
featurebuilderdata (config repo)
Hive
Tables
Airflow DAGs (online/offline)
Online DAG code
featurebuilder (Python service)
Batch Features (PUT)


func( )
Online DAG code
dsfeatures
(Golang service)
Client Endpoints (PUT & GET)
go-lyft-features
(Golang SDK Library)
lyft-dsp-features (Python SDK Library)
On-Demand Features
User / Service


Amazon DynamoDB
Amazon ElastiCache for Redis
Amazon Elasticsearch Service
DataStore
Cache
Embeddings
AWS Datastores
GO
func( )
dsfeatures
(Golang service)

Developer Experience

Software Engineer

FE • BE • Infra

Data Modeler

Data Engineer • Data Scientist

ML Modeler

ML SWE • Data Scientist

Researcher

Data Scientist • Analyst

```sql
SELECT
    user_id,
    COUNT(requested_at) AS rides_requested_l7,
    COUNT(completed_at) AS rides_completed_l7
FROM core.fact_rides
WHERE ds BETWEEN '{ds}' - INTERVAL '7' DAY AND '{ds}'
GROUP BY 1
```

SparkSQL

+

?

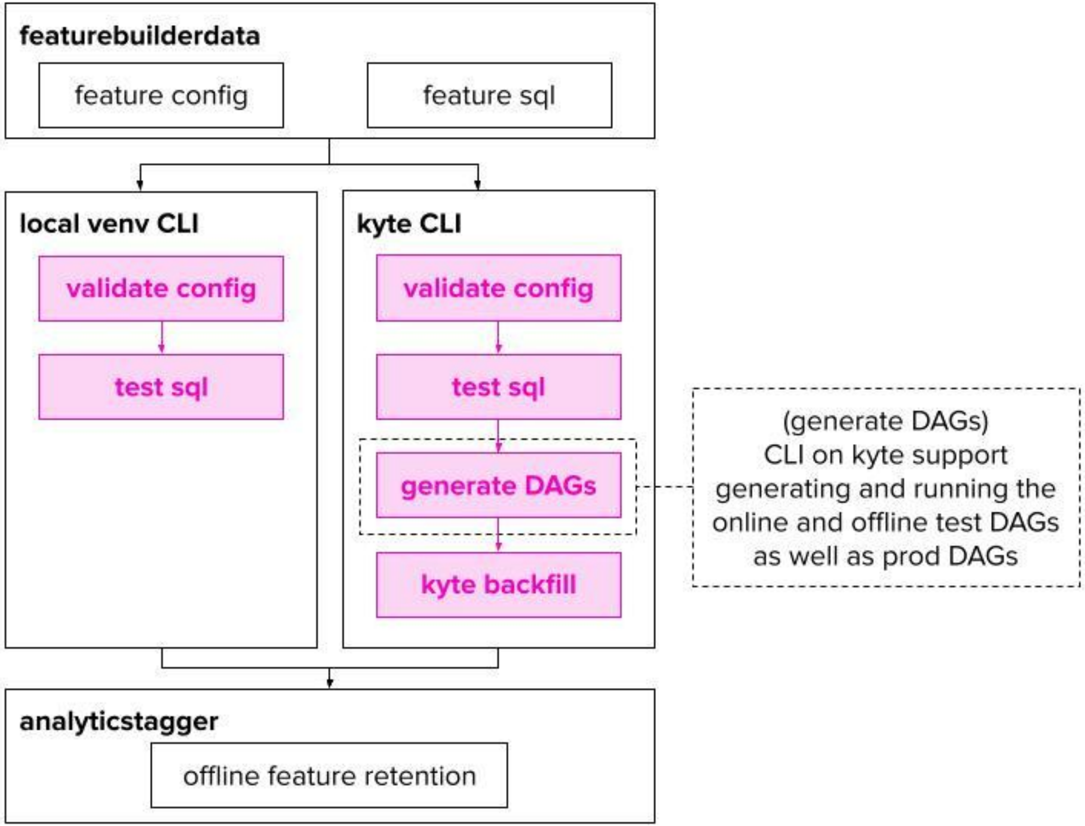
```
{
    "version": 1,
    "entity": "RIDER",
    "feature_group": "rider_stats_l7",
    "query_file": "rider_stats_l7.sql",
    "pagerduty_emails": ["rider@lyft.pagerduty.com"],
    "owner_emails": ["rider-science@lyft.com"],
    "tier": "Tier2",
    "amundsen_tags": [
        "rider"
    ],
    "etl_dependency": [
        {
            "type": "partition",
            "schema": "core",
            "table": "fact_rides"
        }
    ],
    "online_access": true,
    "offline_access": true,
    "features": {
        "rides_requested_l7": {
            "description": "Number of rides requested by the rider in the last 7 days",
            "type": "int",
            "has_personal_data": false
```

```
SELECT
    user_id,
    COUNT(requested_at) AS rides_requested_l7,
    COUNT(completed_at) AS rides_completed_l7
FROM core.fact_rides
WHERE ds BETWEEN '{ds}' - INTERVAL '7' DAY AND '{ds}'
GROUP BY 1
```

SparkSQL

+

JSON configuration


featurebuilderdata
feature config
feature sql
local venv CLI
validate config
test sql
kyte CLI
validate config
test sql
generate DAGs
kyte backfill
(generate DAGs)
CLI on kyte support
generating and running the
online and offline test DAGs
as well as prod DAGs
analyticstagger
offline feature retention

Amundsen

rider

Resource	
○ Datasets	10000
○ Dashboards	10000
○ People	430
● ML Features	5676

Availability

- [ ] Hive (offline)
- [ ] Dynamo (online)

Entity

Exact name or *wild card*

Feature Name

Exact name or *wild card*

Feature Group

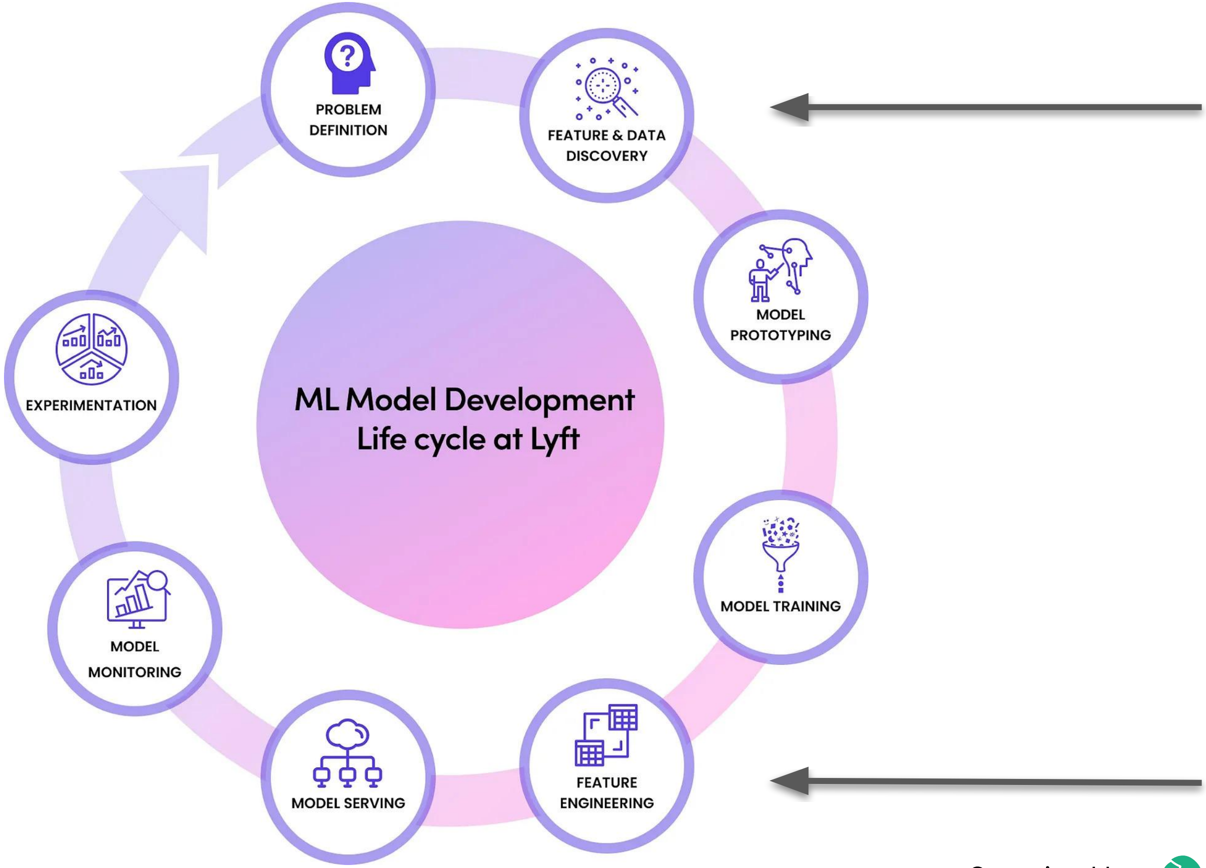
RESOURCE

rider_value.rider_value.3
Rider value = Net Revenue + Credit Card Fees + PAX Eng + PAX Acq + DVR Eng + Insurance. Updated monthly

rider_value.rider_value.3
Rider value = Net Revenue + Credit Card Fees + PAX Eng + PAX Acq + DVR Eng + Insurance. Updated monthly

rider_value.rider_value.2

rider_value.rider_value.2


ML Model Development Life cycle at Lyft
PROBLEM DEFINITION
FEATURE & DATA DISCOVERY
MODEL PROTOTYPING
MODEL TRAINING
FEATURE ENGINEERING
MODEL SERVING
MODEL MONITORING
EXPERIMENTATION

Evolution

Batch Features

The biggest family of features at Lyft.

1. Flyte → Self-managed Airflow → Astronomer
2. Deprecate HiveQL & Redshift usage
3. Support for staging (unlocks prototyping & testing)
4. Error Handling (explicit categorization of value type)
5. New Golang library & Offline SDK
6. Data Contracts

Streaming & Real-Time Features

The growing potential of fresh features.

1. Added OpenSearch integration for use of Embeddings
2. “RealtimeMLPipeline” interface for developing Flink streaming apps with minimal complexity

Optimization

Data Generation

How to make a platform of platforms more lightweight?

1. Support for more DAG scheduling windows
2. Monitoring & Dashboards for Failed DAGs (& tasks)
3. Deprecation of underutilized features by R/W activity
4. DAG for offline + DAG for online → **merged**
5. Ownership & Observability Propagation

Data Retrieval

Can we make the life of our customers significantly better?

1. ElastiCache → ValKey
2. cache payload field reduction
3. EKS pod size reduction
4. Retry + Timeout management
5. Metadata + value cache & datastore TTL
6. Redundant network calls + payload processing steps

Performance

8.2ms → 5.5ms

-32.9%

Avg. P95 Read Latency, Oct. 2024 → Oct. 2025

2,465 → 2,763

+12.1%

Num. Batch Features, Oct. 2024 → Oct. 2025

62 → 77

+24.2%

Num. Distinct Callers, Oct. 2024 → Oct. 2025

~2 trillion → ~3.2 trillion

R/W Call Volume Annual Sum, 2024 → 2025*

*projected

Q&A

https://www.linkedin.com/in/rohanvarshney/