

End-to-End Data Management in Support of an ML RecSys

David Cohen

The Intel logo is located in the bottom left corner of the slide. It consists of the word "intel" in a lowercase, white, sans-serif font, followed by a registered trademark symbol (®). The logo is positioned to the right of a decorative graphic of three overlapping squares in shades of blue.




intel®

Macro Trends

Machine Learning is a dominant focus for revenue generating businesses across the Digital Services market. Year-on-year, exponential data growth, model demand for computations to process, and memory to access feature embeddings derived from this data drives investments across this market.

Machine Learning is a Dominant Force in the Industry

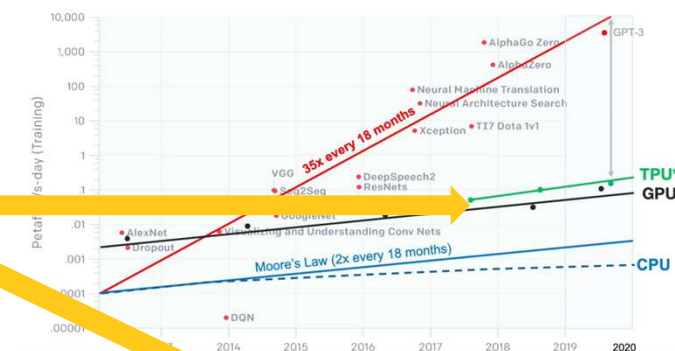
Industry Priorities for Optimizations

	The Top		
Technology			
Opportunity	Software	Algorithms	Hardware Architecture
	<ul style="list-style-type: none"> Software performance engineering 	<ul style="list-style-type: none"> New algorithms 	<ul style="list-style-type: none"> Hardware streamlining
Examples	<ul style="list-style-type: none"> Removing software bloat Tailoring software to hardware features 	<ul style="list-style-type: none"> New problem domains New machine models 	<ul style="list-style-type: none"> Processor simplification Domain specialization

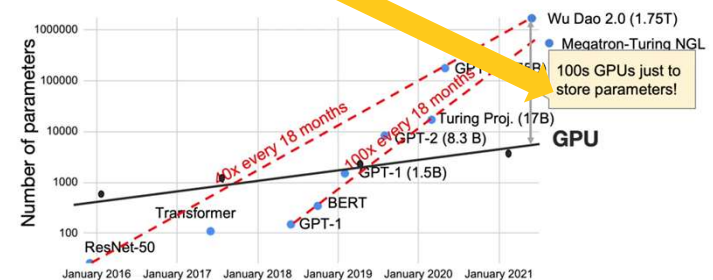
The Bottom
for example, semiconductor technology

Demand from ML Training outstripping Tech Capacity

ML Training Demand vs Computational Capacity



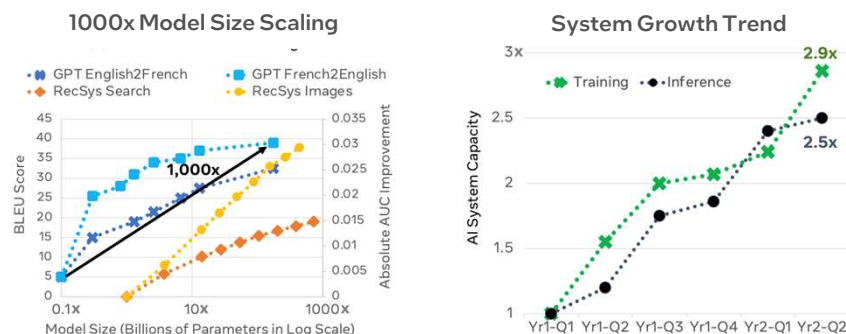
Memory Demand far exceeds vs GPU memory capacity



Recommendation Systems at Meta – a Case Study

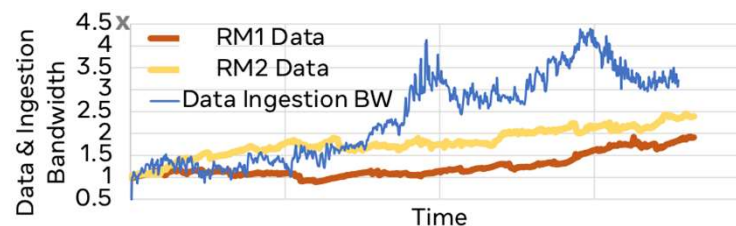
Background

- In the past decade, Meta has seen an exponential increase in AI training data and model capacity.
- data storage and the ingestion pipeline accounts for a significant portion of the infrastructure and power capacity compared to ML training and end-to-end machine learning life cycles.
- Recommendation Systems are one of Meta’s fastest growing areas of ML usage.



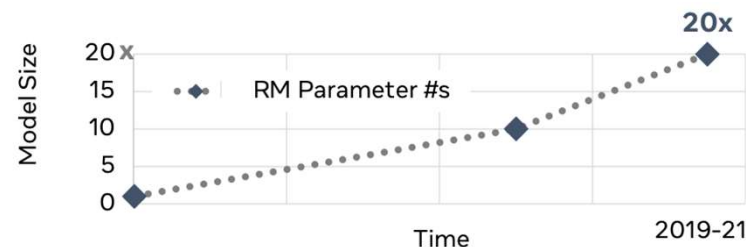
Recommendation System Data Growth Trend

Between 2019 and 2021 the amount of data for recommendation use cases has roughly doubled, leading to a 3.2x increase in the data ingestion bandwidth demand.



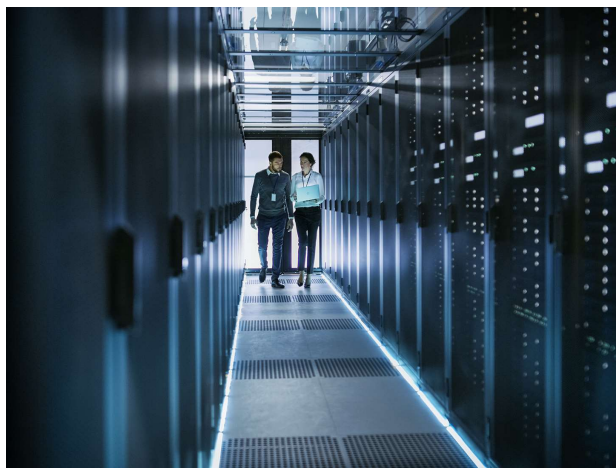
Recommendation System Model Growth Trend

Meta’s recommendation (aka “Recall”) and ranking model sizes have increased 20x during the same time period.



What Does a Modern Recommendation System Look Like?

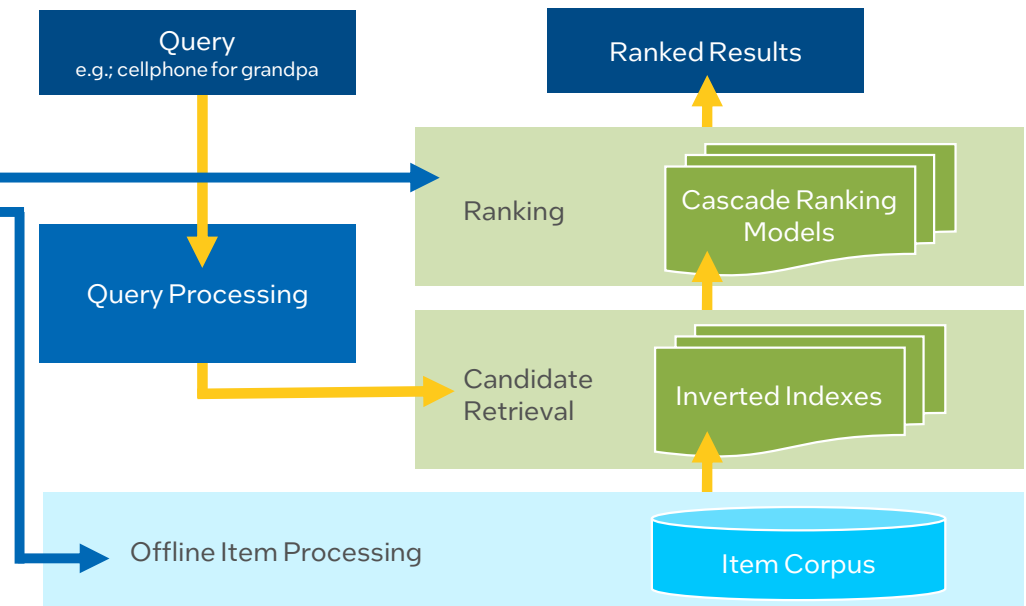
1. Cloud-Native Supercomputing for Training



Two
Pretrained
Models

- Alibaba's Super Computing Cluster (SCC)
- Amazon's Virtual Supercomputer
- Google Pathways
- Meta's AI Research SuperCluster (RSC)
- Microsoft's ZeRO-Infinity

2. Recommendation System Software Architecture (Online Serving)

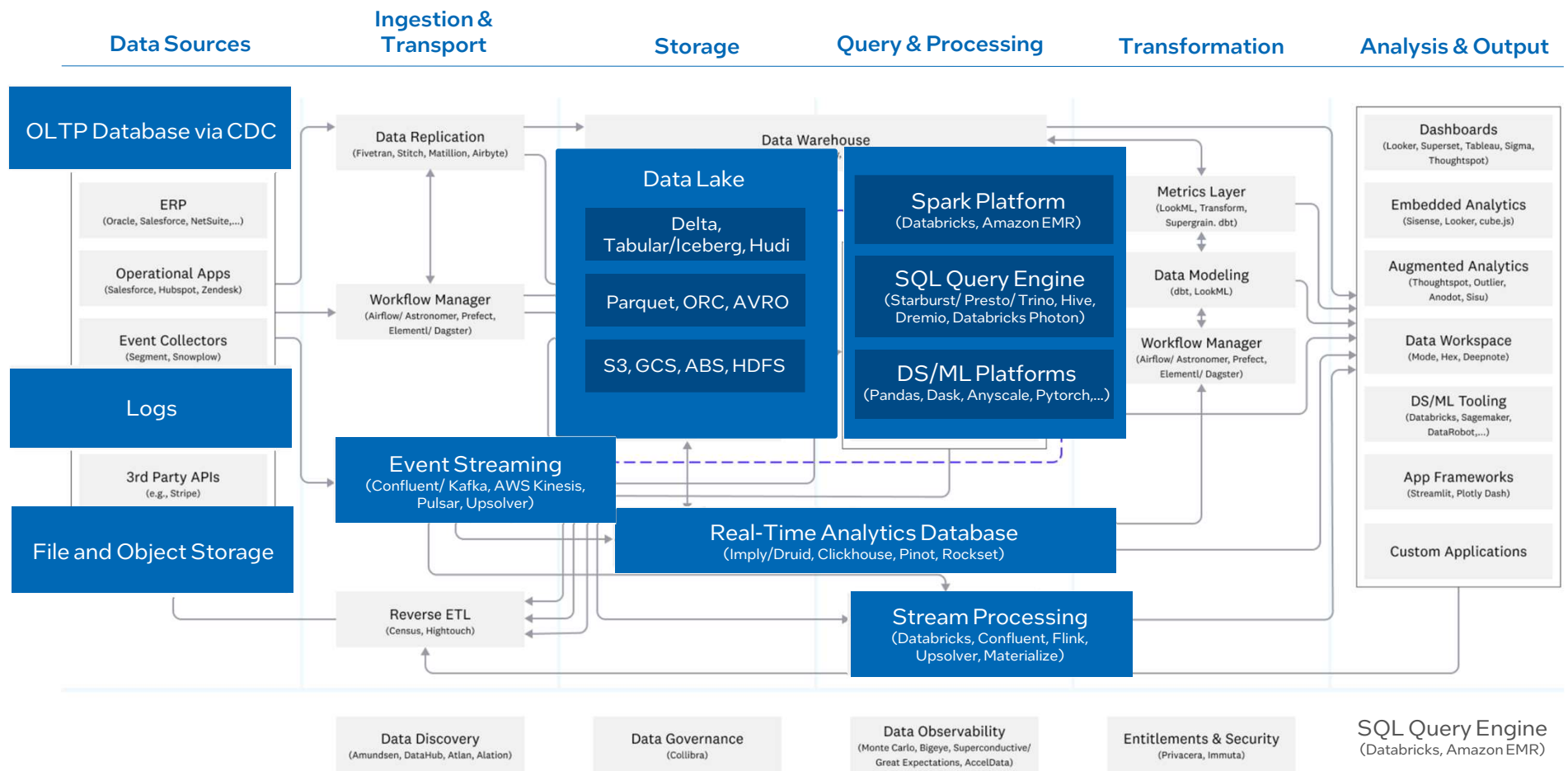


What affect does this have on the Data Management discipline at companies that rely on Recommendation Systems?

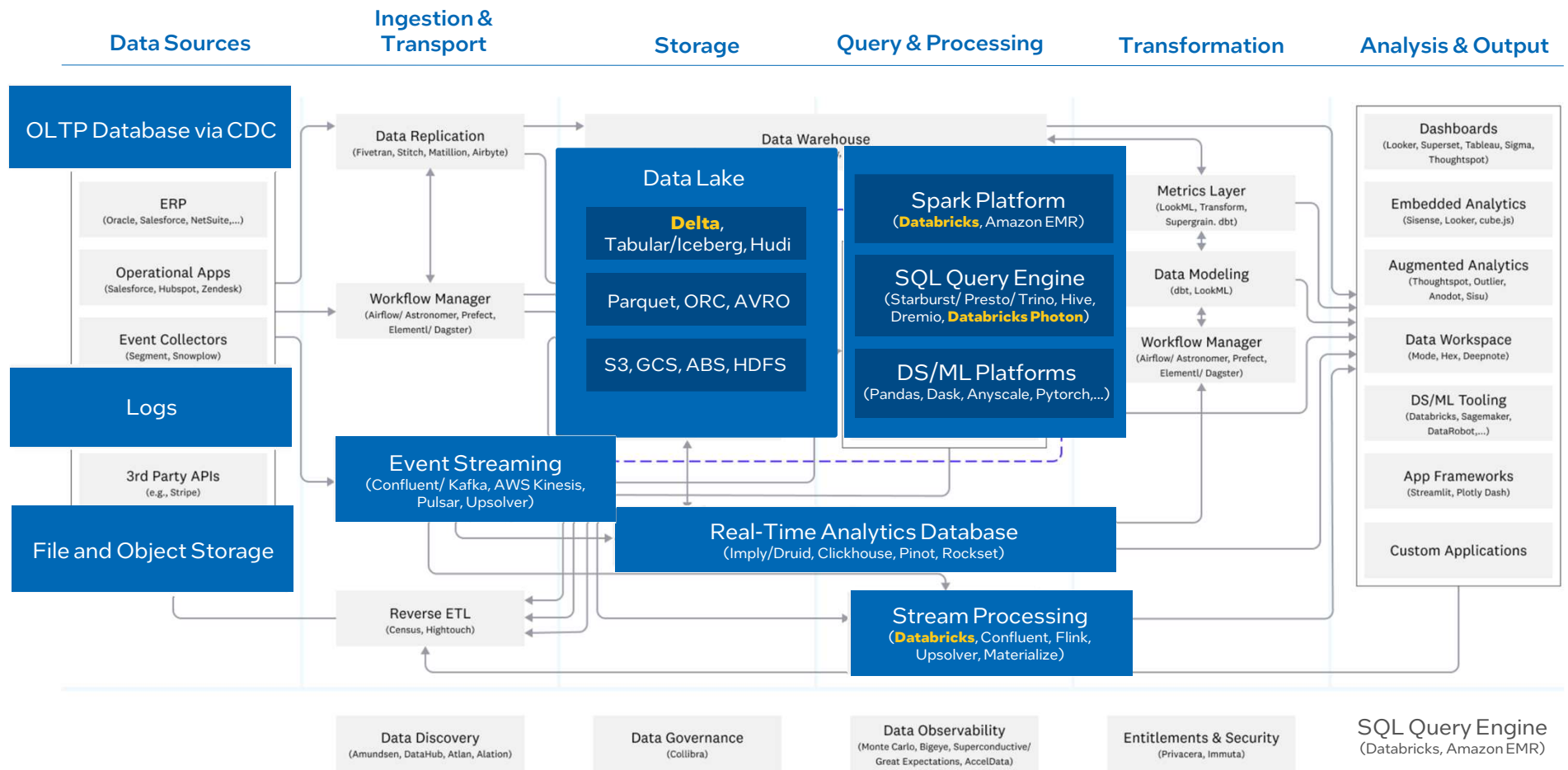
Data Management to support ML

Not surprisingly, the data management discipline of companies in the Digital Services segment is a major departure from that of traditional enterprises. Data Curation to feed the process to produce pretrained models with billions to trillions of feature/embeddings and enabling these embeddings to evolve independently of the training process has shifted the focus to near-real-time, memory-centric processing. In this section we'll look the impact this is having on the Data Management landscape.

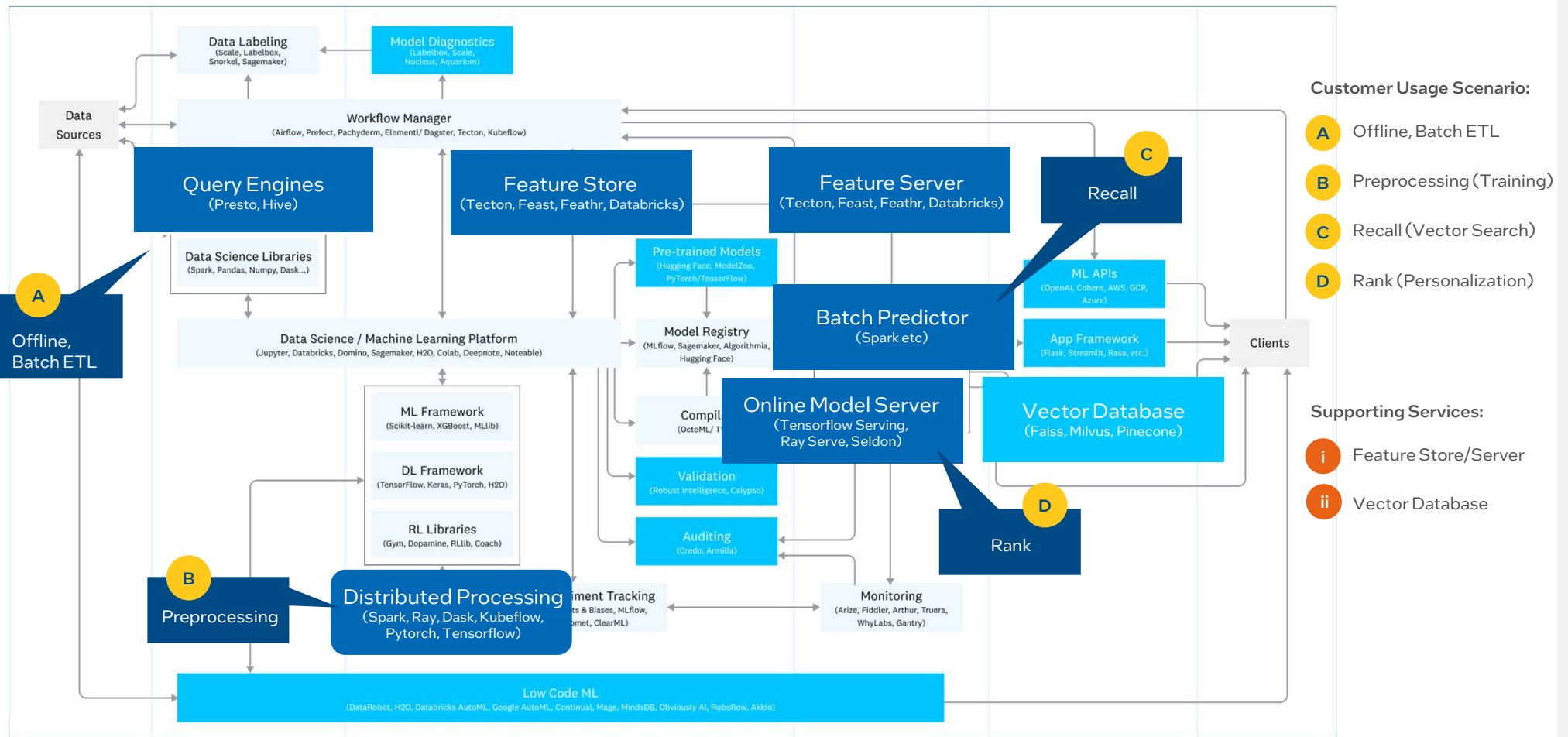
Data Infrastructure Landscape in Support of AI/ML



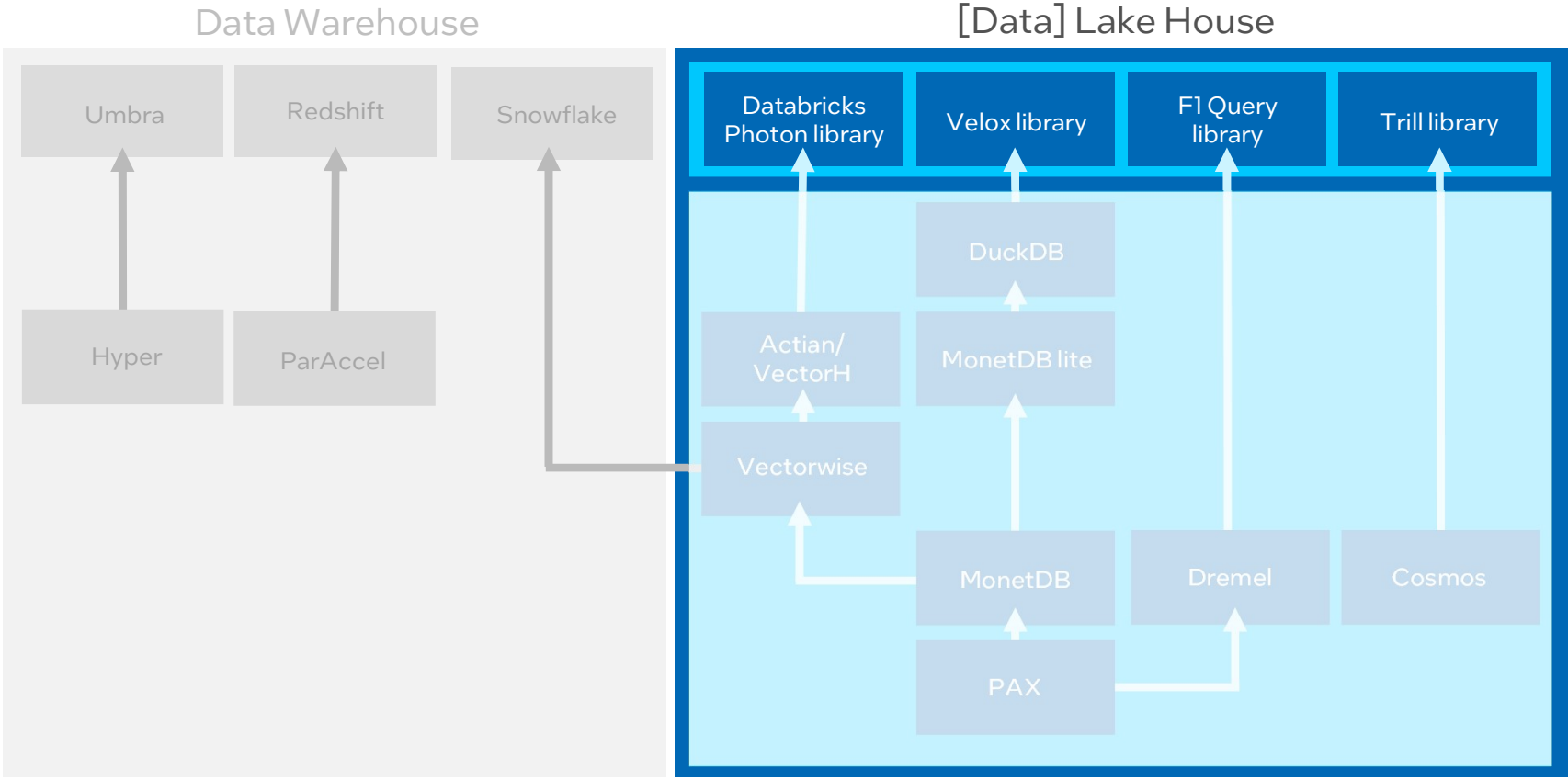
Positioning Databricks Within this Data Infrastructure Landscape



End-to-End Data Management in Support of an ML RecSys



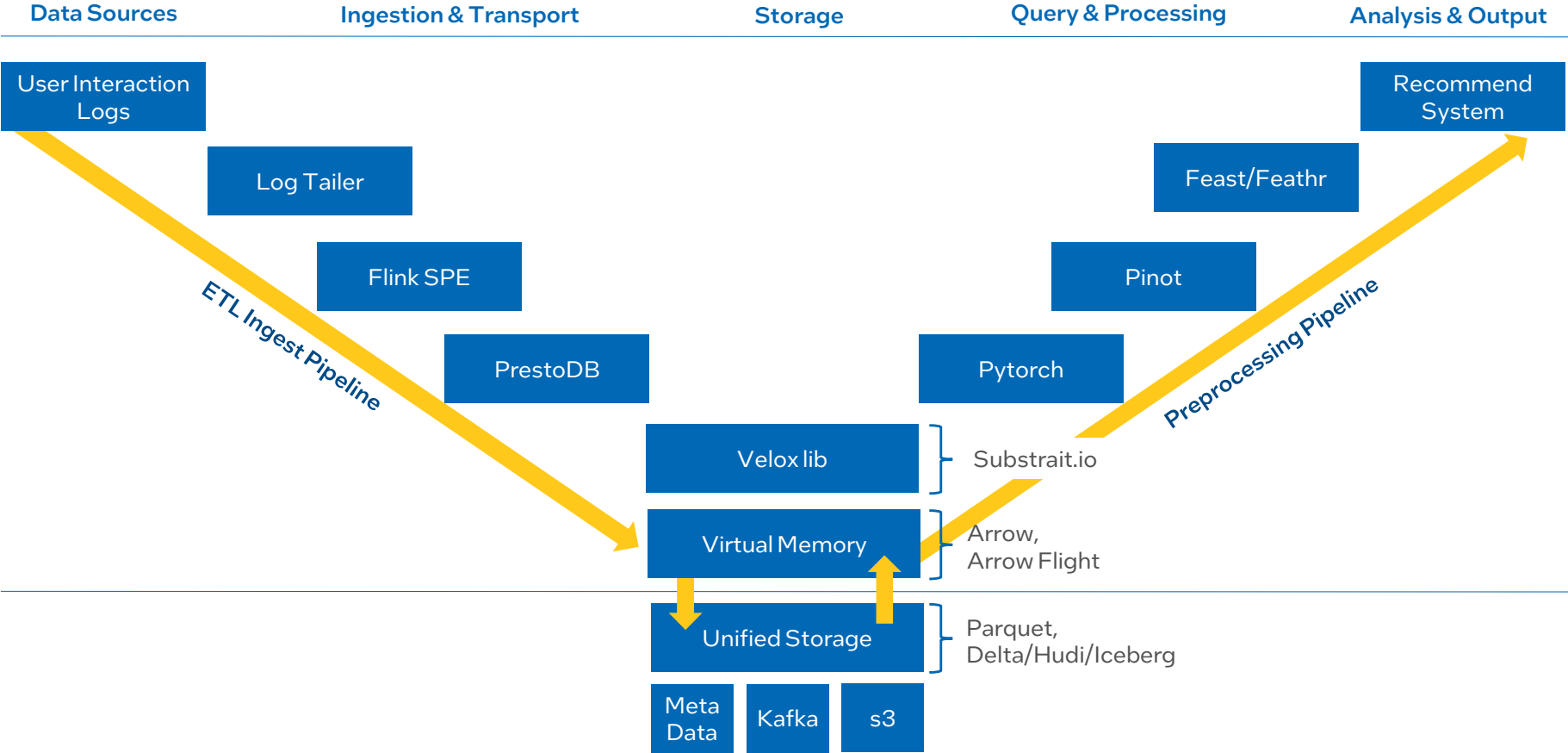
Evolving from Big-Data to the Lake House in Support of Machine Learning



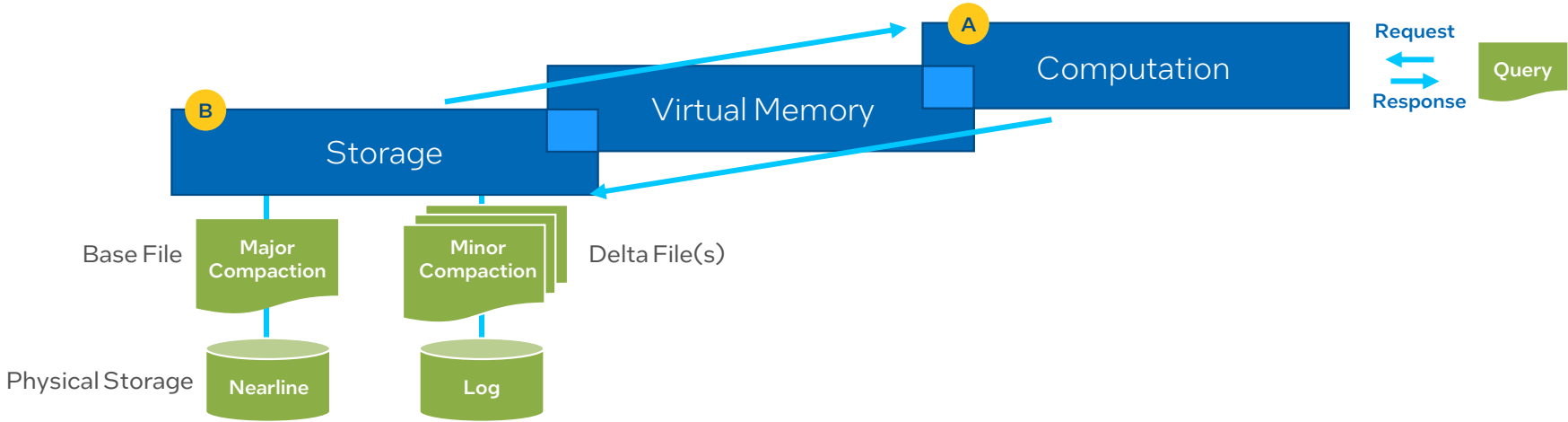
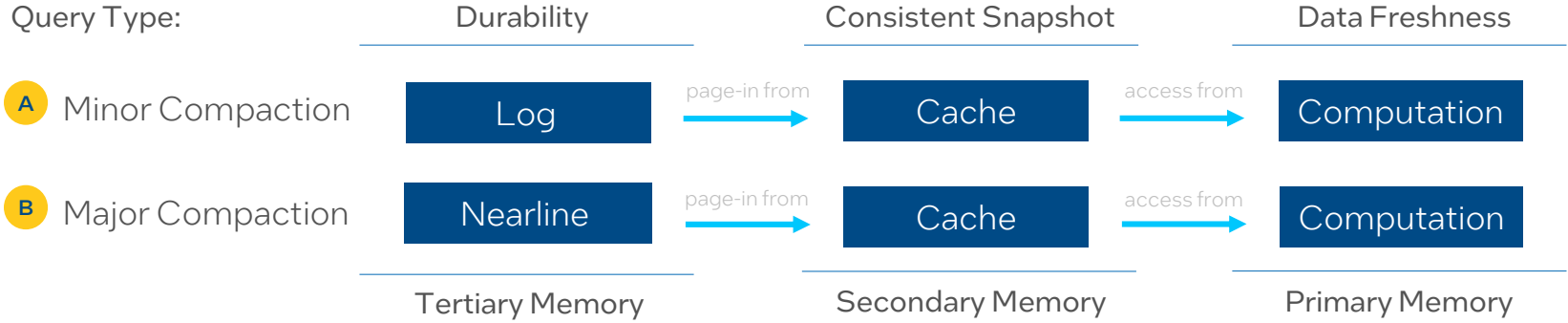
How are these Components Applied?

The software components from the previous section enable Feature Engineering, making Feature Embeddings available during training as well as serving time, and keeping these embeddings up-to-date, independently of the pretrained model.

End-to-End Data Management in Support of RecSys



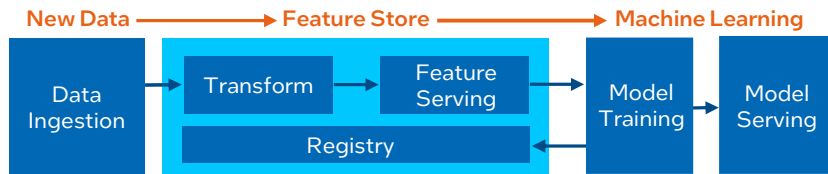
Optimizing the Lake House



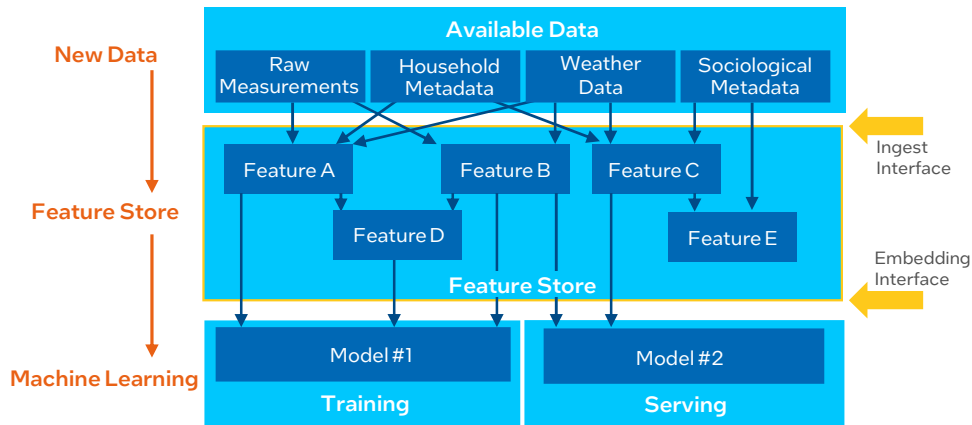
Cloud-Native Feature Engineering

1. The Data Management Portion of Feature Engineering

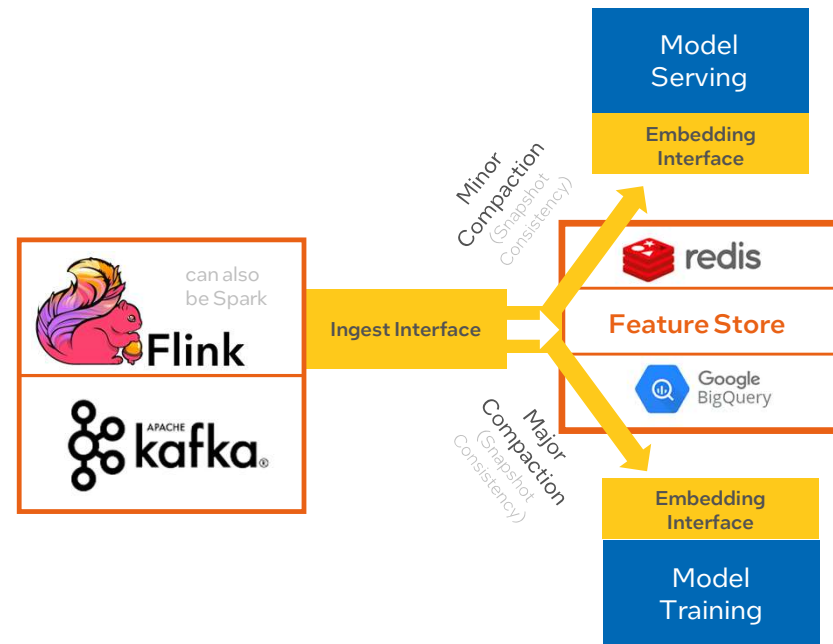
a. Dataflow in Feature Engineering



b. Data Management Ingest and Embedding Interfaces in the Dataflow



2. Positioning the Feature Store in the Feature Engineering Dataflow

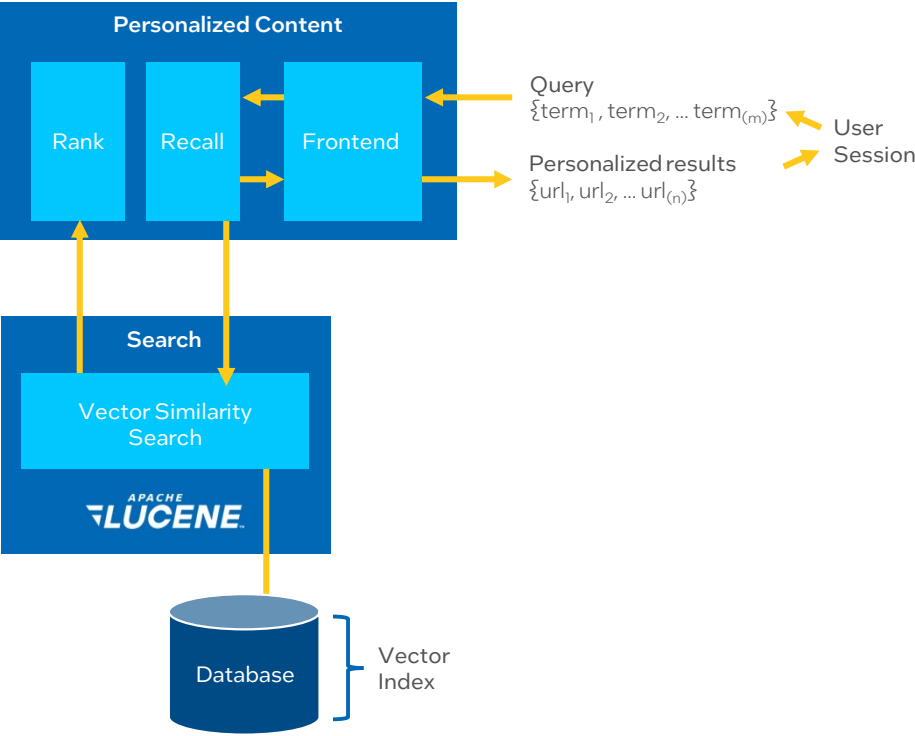


Illustrative Feature Store Implementations:

- [Feast \(e.g., Twitter\)](#)
- [Feathr \(e.g., LinkedIn\)](#)

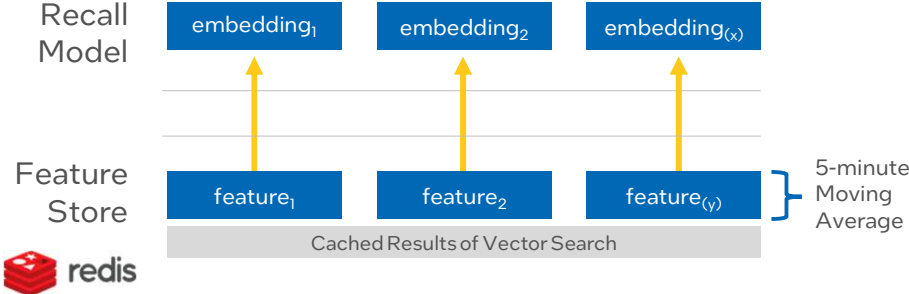
How do Feature Computations Get Executed at Serving Time?

1. ElasticSearch support for “Approximate-Nearest-Neighbor-(ANN)” search



2. Online Feature Store (e.g., Feast, Feathr)

When a “Feature” is computed over a window (e.g., “5-minute moving average”) the values of the associated embedding must be recomputed online.



Customer Proxy Workloads

The data management processing covered thus far is not addressed by traditional benchmarks, from the Transaction Processing Council (TPC). As an example, the use of a denormalized data model, complex data types (e.g. array, map, etc.), and user-defined, nested data structures is not covered by any TPC benchmark. We have started the process of addressing this by developing “Customer Proxy Workloads.” This work is motivated by this gap with the goal to try to address this through the community.

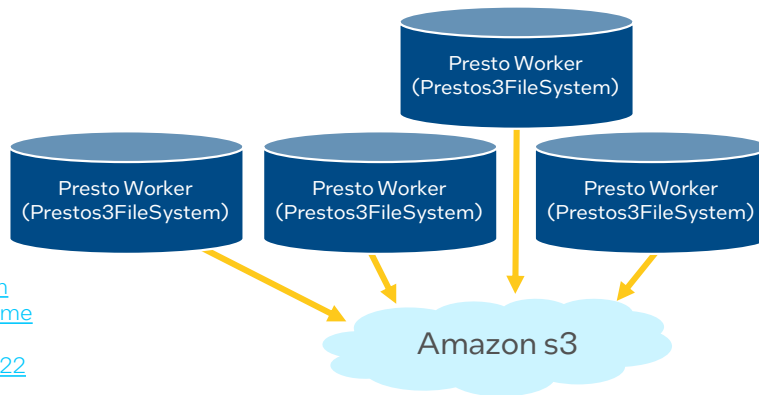
Data Management in support of Recommendation Systems

Customer	Customer Classification	Geo-Region	Extract, Transform, & Load (ETL)	Preprocessing	Recall		Ranking
					Embedding Index for Similarity Search	Candidate Generation Similarity Search	Real-time Update Feature Store
1. Alibaba	Hyperscaler	PRC			5,11	5,11	5,11
2. ByteDance	Strategic	PRC	12	12	12	12	12
3. JD.com	Strategic	PRC	20	20	20	20	20
4. <tbd>	Strategic	PRC					
5. Meta	Hyperscaler	NA	8,13,21	21	4,13	4,13	4,13
6. Pinterest	Strategic	NA	8,17,19	10	1	1	1,3,6
7. Twitter	Strategic	NA	8,14	18	2	2	
8. Uber	Strategic	NA	7,8,9	7	15	15	15,16

- [Baltescu et al. "ItemSage: Learning Product Embeddings for Shopping Recommendations at Pinterest." 2022](#)
- [El-Kishky et al. "TwHIN: Embedding the Twitter Heterogeneous Information Network for Personalized Recommendation." 2022](#)
- [He et al. "Pinterest Home Feed Unified Lightweight Scoring: A Two-tower Approach." 2021](#)
- [Huang et al. "Embedding-based Retrieval in Facebook Search." 2020](#)
- [Jiang et al. "Alibaba hologres: a cloud-native service for hybrid serving/analytical processing." 2020](#)
- [Kenny. "Real-time ranking at Faire part 2: the feature store." 2022](#)
- [Laptev et al. "Time-series Extreme Event Forecasting with Neural Networks at Uber." 2017](#)
- [Luo et al. "From Batch Processing to Real Time Analytics: Running Presto® at Scale." 2022](#)
- [Luo. "Engineering Data Analytics with Presto and Apache Parquet at Uber."](#)
- [Pancha et al. "SearchSage: Learning Search Query Representations at Pinterest." 2021](#)
- [Shaoyuan et al. "Proxima: A Vector Retrieval Engine independently developed by Alibaba DAMO Academy." 2021](#)
- [Shi et al. "IPS: Unified Profile Management for Ubiquitous Online Recommendations." 2021](#)
- [Tang et al. "MSURU: Large Scale E-commerce Image Classification with Weakly Supervised Search Data." 2019](#)
- [VijayaRenu et al. "Twitter Sparrow: Reduce Event Pipeline latency from hours to seconds." 2021](#)
- [Wang et al. "Food Discovery with Uber Eats: Recommending for the Marketplace." 2018](#)
- [Wang et al. "Optimal Feature Discovery: Better, Leaner Machine Learning Models Through Information Theory." 2021](#)
- [Wang et al. "Pinterest's Analytics as a Platform on Druid \(Part 3 of 3\)." 2021](#)
- [Zhang et al. "Processing billions of events in real time at Twitter." 2021](#)
- [Zhang et al. "Efficient Resource Management at Pinterest's Batch Processing Platform." 2021](#)
- [Zhang et al. "Towards Personalized and Semantic Retrieval: An End-to-End Solution for E-commerce Search via Embedding Learning." 2020](#)
- [Zhao et al. "Understanding data storage and ingestion for large-scale deep recommendation model training: industrial product." 2022](#)

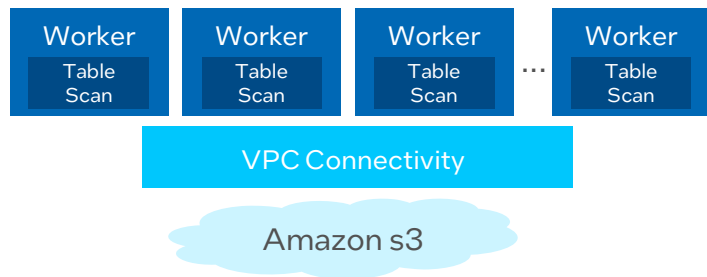
Customer Proxy Workload for Batch. Offline ETL Processing

1. General Support for s3 object store



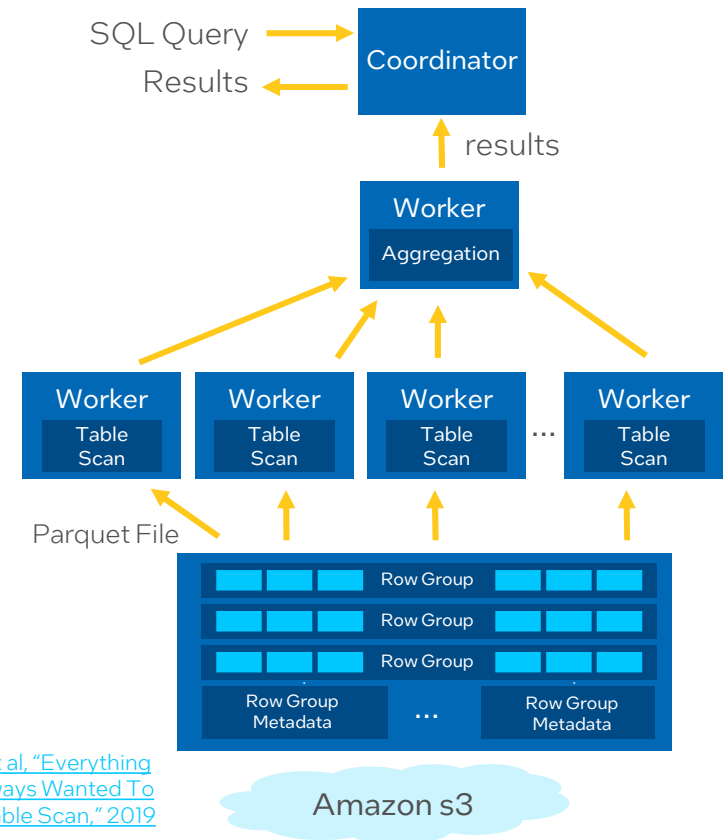
[Luo et al, "From Batch Processing to Real Time Analytics: Running Presto® at Scale," 2022](#)

3. Connectivity optimized for throughput



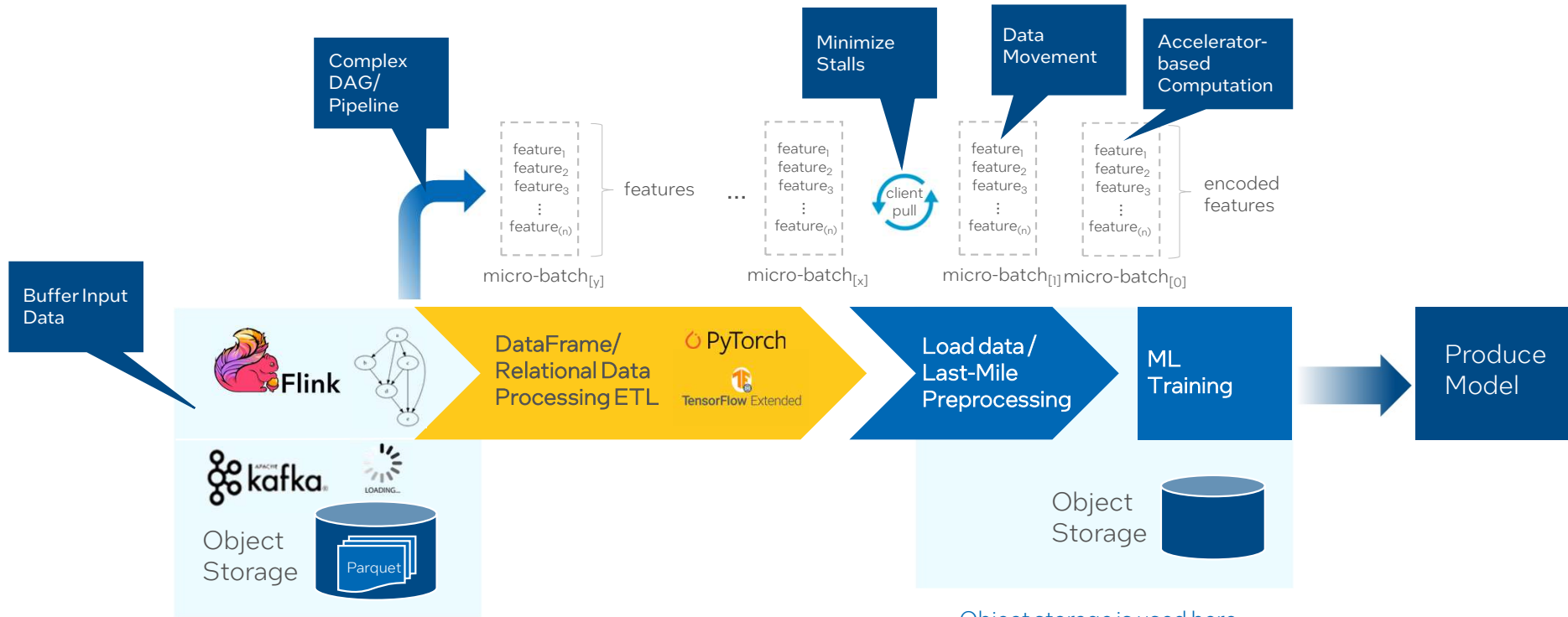
[Patiejunas et al, "Facebook's Disaggregated Storage and Compute for MapReduce," 2016](#)

2. s3 Select (Predicate Pushdown) and Parquet



[Erling et al, "Everything You Always Wanted To Do in Table Scan," 2019](#)

Customer Proxy Workload for “Preprocessing” for Training Model



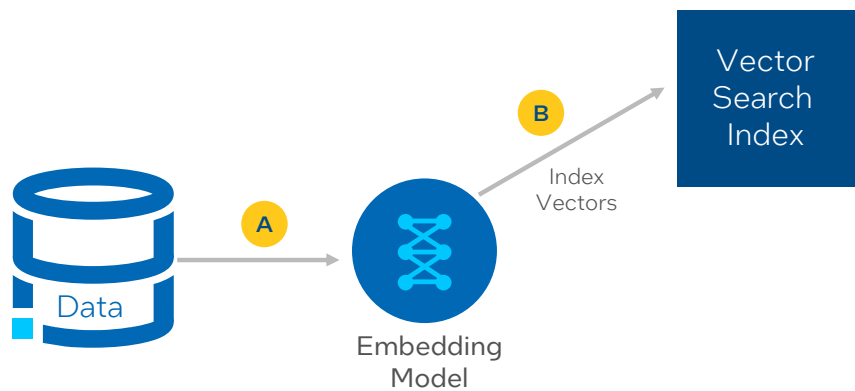
- Produced by Offline ETL pipeline (see customer proxy workload #1)
- Newly ingested dataset (“major compaction”)
- Dataset size is on order of 800TBs to 1s of PBs
- Parquet-based dataset uses denormalized data model with complex, nested data types

Object storage is used here for periodic checkpoints

[Mohan et al, “Analyzing and Mitigating Data Stalls in DNN Training,” 2021](#)
[Murray et al, tf.data: a machine learning data processing framework,” 2021](#)
[Zhao et al, “Understanding data storage and ingestion for large-scale deep recommendation model training: industrial product,” 2022](#)

Customer Proxy Workload for Vector/Embedding-Based Index

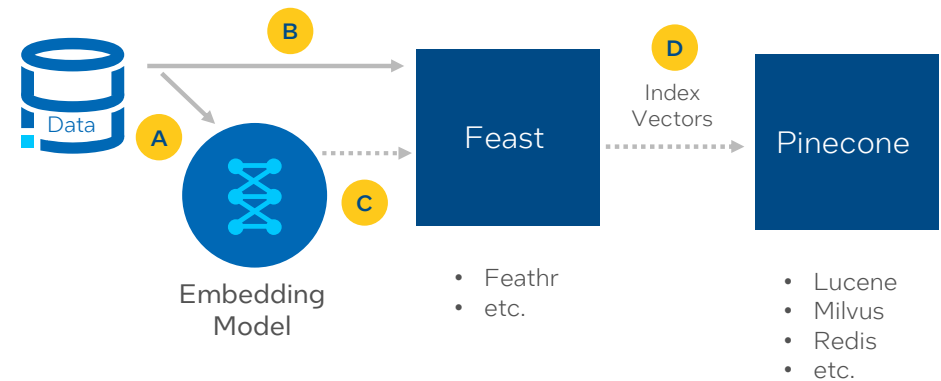
1. Vector/Embedding Index without Feature Store



- A** Pretrained Embedding Model
- B** Batch Process for Indexing

[Pinecone, "Operationalize vector search with Pinecone and Feast Feature Store," 2021](#)

2. Vector/Embedding Index without Feature Store

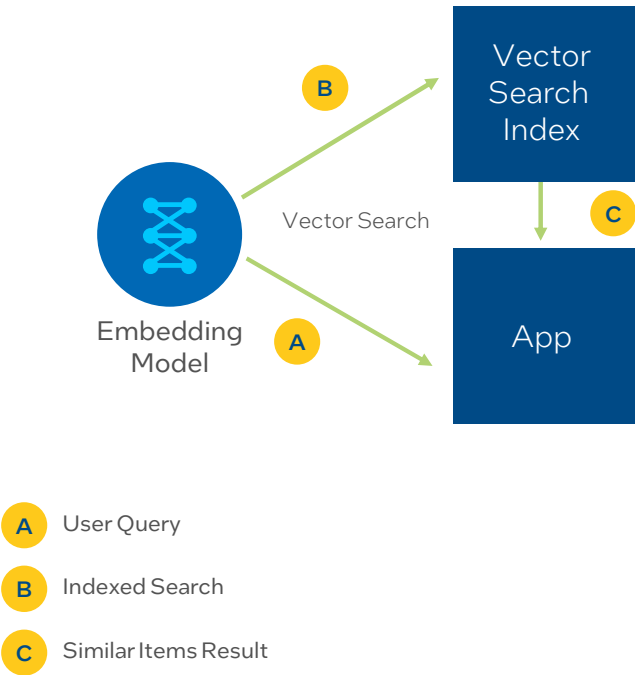


- A** Pretrained Embedding Model
- B** Populate Features/Embeddings
- C** Caching and Periodic Updates to Feature/Embedding (e.g., Moving Window)
- D** Batch process for Indexing

- Offline
- Online

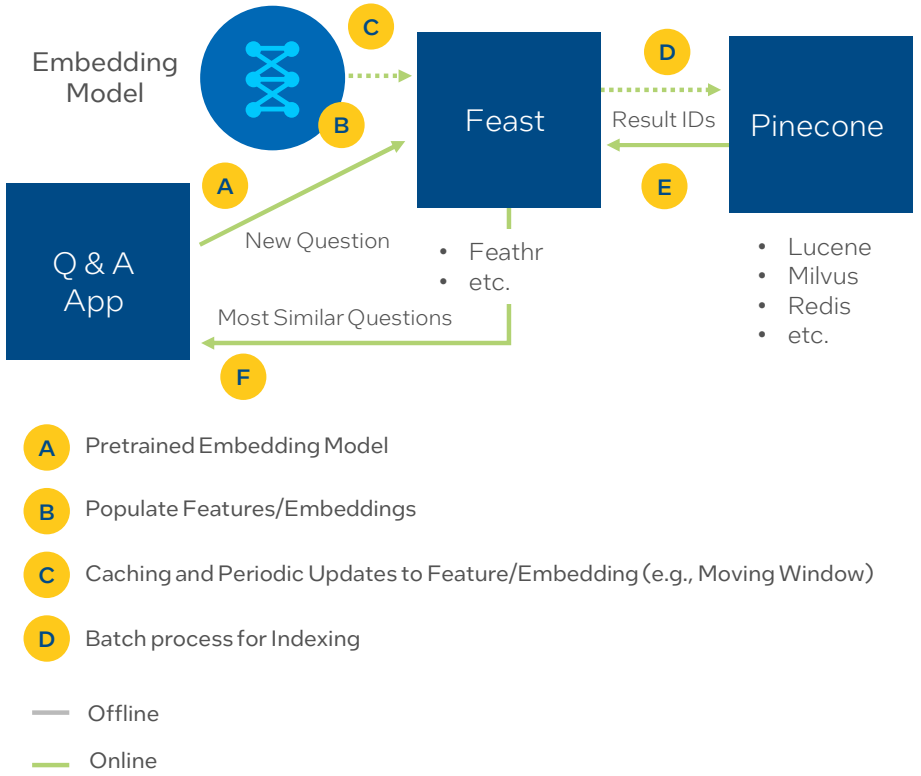
Customer Proxy Workload for Vector Similarity Search (Recall)

1. Vector Similarity Search without Feature Store

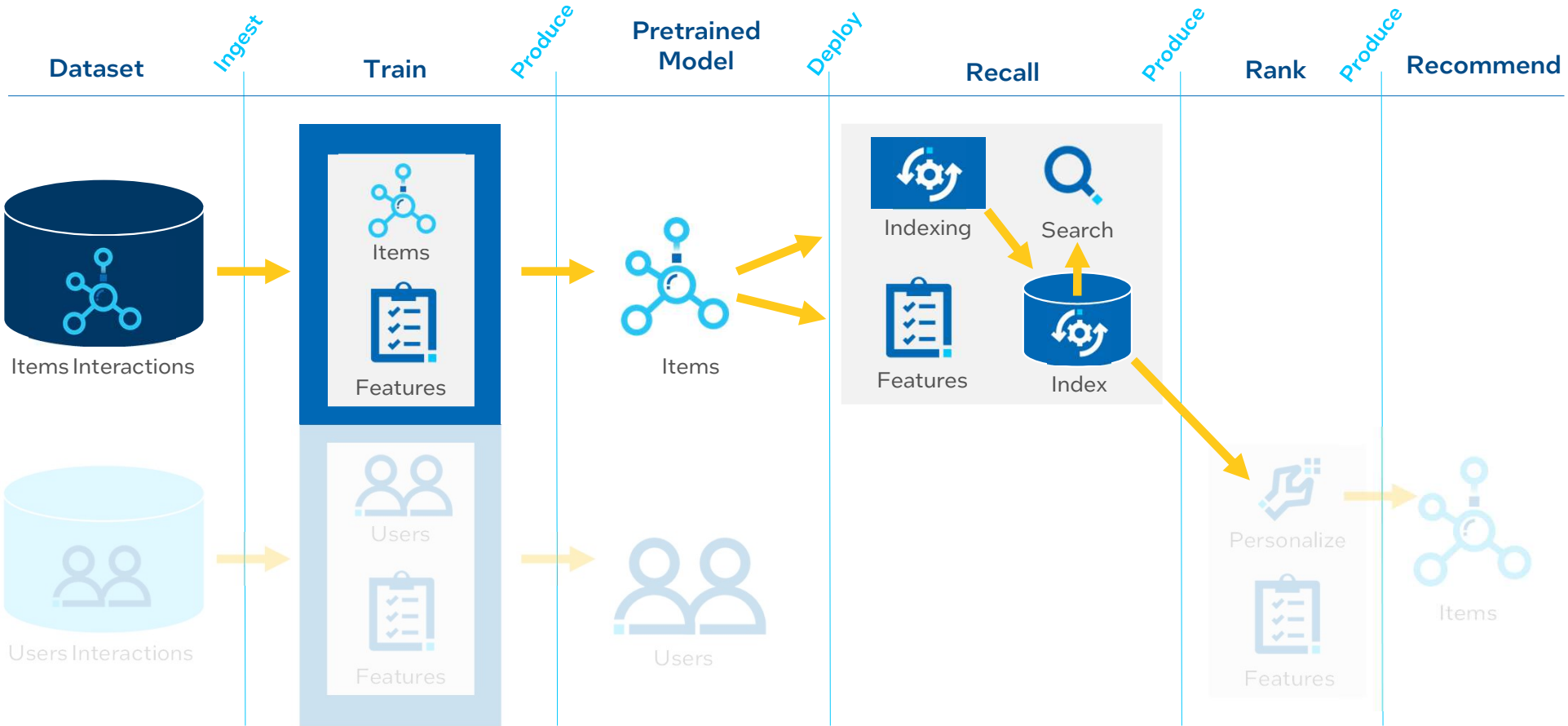


[Pinecone, "Operationalize vector search with Pinecone and Feast Feature Store," 2021](#)

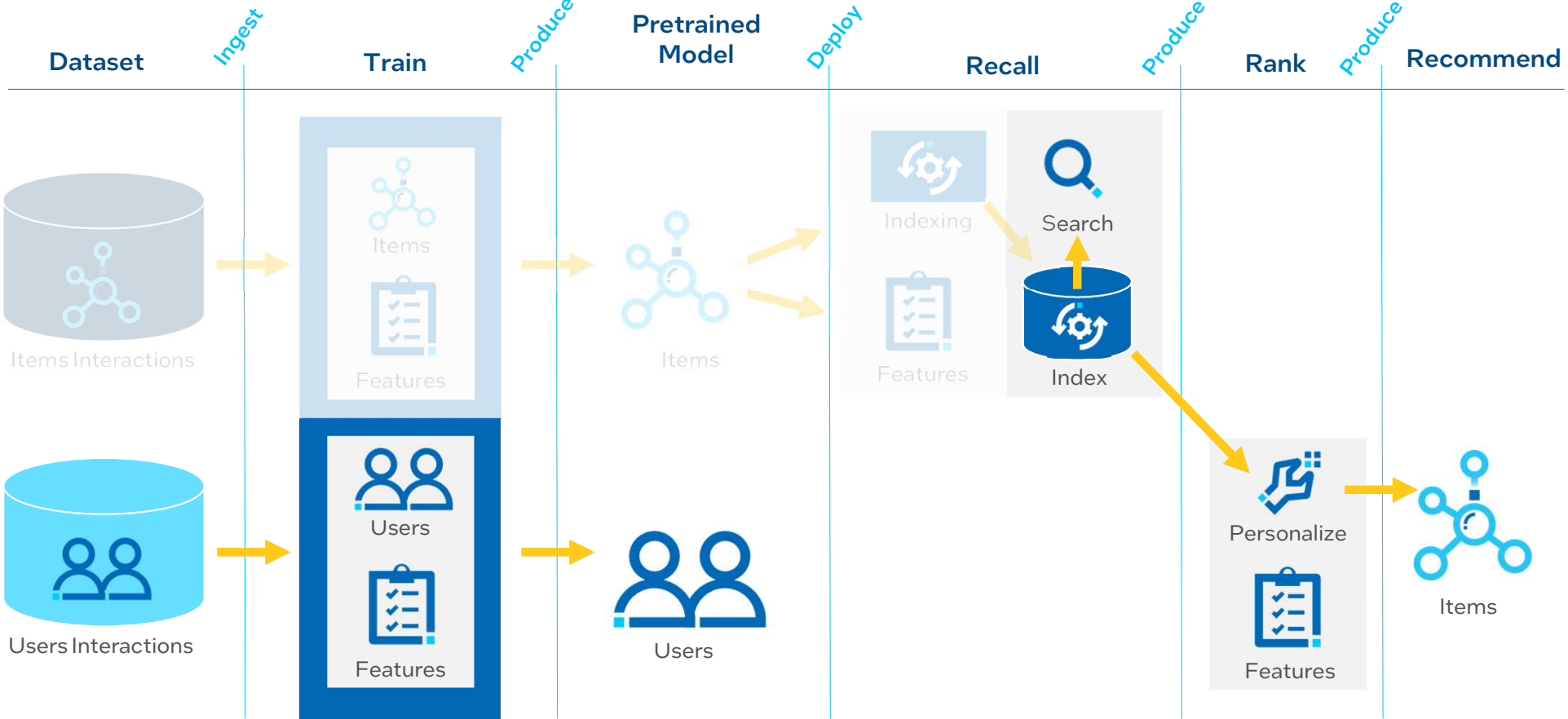
2. Vector/Embedding Index without Feature Store



Customer Proxy Workload for Recall



Customer Proxy Workload for Ranking



Conclusion

Wrap-up and Parting thoughts

Growth exceeds Capacity

Processing capacity is not keeping up with Data Growth and Growth in Model Size far exceeds Memory Capacity.

Data Management is being disrupted by Machine Learning

In this talk we focused on the impact Machine Learning is having on Data Management, the emergence of the Lake House Architecture, and the refactoring of common data management capabilities into a shared library that can be leveraged by multiple engines accessing Lake-House-resident dataset.

Emerging Trends as outcomes of this Disruption

We then looked at the emerging trends: disaggregated, Secondary Memory, Feature Engineering and Feature Stores, and Vector Similarity Search Engines.

Getting involved in the Velox and Substrait projects as well as Proxy Workloads

Finally, we talked about the use of Velox, Substrait, and the Lake House architecture along with the customer proxy workloads we are developing. These represent a refactoring of the Data Management capabilities used to support Recommendation System platforms and Machine Learning in general. The Velox and Substrait.io projects are available on github. We're working through how best to make the proxy workloads available.

intel®