AWS re:Invent

ANT316

Effective Data Lake: Design Patterns and Challenges

Radhika Ravirala
EMR Solutions Architect
Amazon Web Services

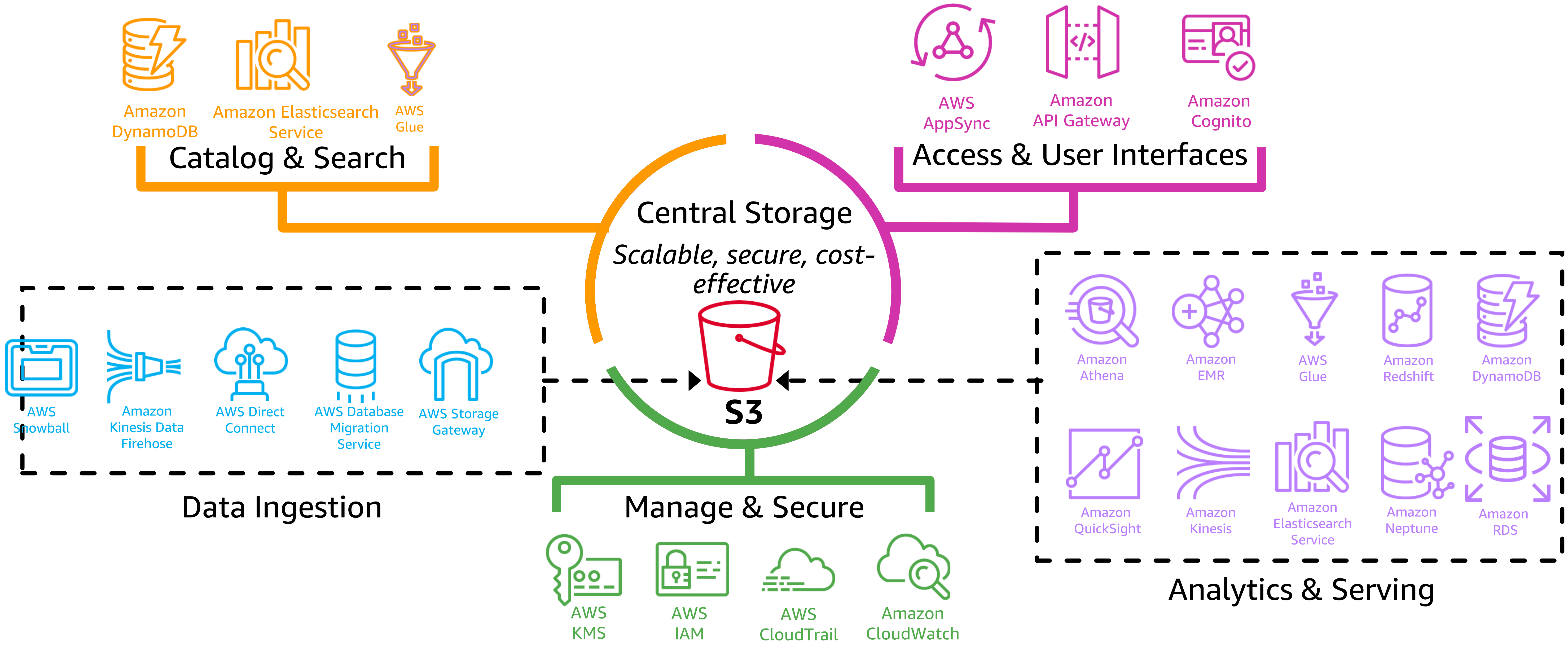
Moataz Anany
Solutions Architect
Amazon Web Services

Agenda

- Why a Data Lake?
- Data Lake concepts
- Common asks and challenges
- Data Lake design patterns
- Security and governance patterns
- Q & A

Why a Data Lake?

Data lake on AWS



Amazon DynamoDB

Amazon Elasticsearch Service

AWS Glue

Catalog & Search

AWS AppSync

Amazon API Gateway

Amazon Cognito

Access & User Interfaces

Central Storage

Scalable, secure, cost-effective

S3

AWS Snowball

Amazon Kinesis Data Firehose

AWS Direct Connect

AWS Database Migration Service

AWS Storage Gateway

Data Ingestion

Manage & Secure

AWS KMS

AWS IAM

AWS CloudTrail

Amazon CloudWatch

Amazon Athena

Amazon EMR

AWS Glue

Amazon Redshift

Amazon DynamoDB

Amazon QuickSight

Amazon Kinesis

Amazon Elasticsearch Service

Amazon Neptune

Amazon RDS

Analytics & Serving

The core of a Data Lake

Versatile Compute Layers

Athena

Amazon EMR

Amazon Redshift Spectrum

Data Lake

Data & Metadata

Amazon S3

AWS Glue Data Catalog

The concept of a Data Lake

- All data in one place, a single source of truth
- Handles structured/semi-structured/unstructured/raw data
- Supports fast ingestion and consumption
- Schema on read
- Designed for low-cost storage
- Decouples storage and compute
- Supports protection and security rules

Data Lake concepts

Tier 1 Data Lake: Ingestion

Amazon S3

Single source of truth for raw data

Use least transformations

Use lifecycle policies to Amazon Simple Storage Service (Amazon S3) IA or Amazon Glacier

Tier 2 Data Lake: Analytics

Amazon S3

Use columnar formats – Parquet/ORC

Organized into partitions

Coalescing to larger partitions over time

Optimized for analytics

Tier 3 Data Lake: Analytics

Amazon S3

Domain level DataMart

Organized by use cases

Optimized for specialized analysis

Amazon Redshift

Data Warehouse:

Fast speeds over structured schemas

Serves dashboards and reports

Fine-grained access controls

Supports joining native and external tables

Lifecycle back to S3 Data Lake

Common asks and challenges

Some customer asks

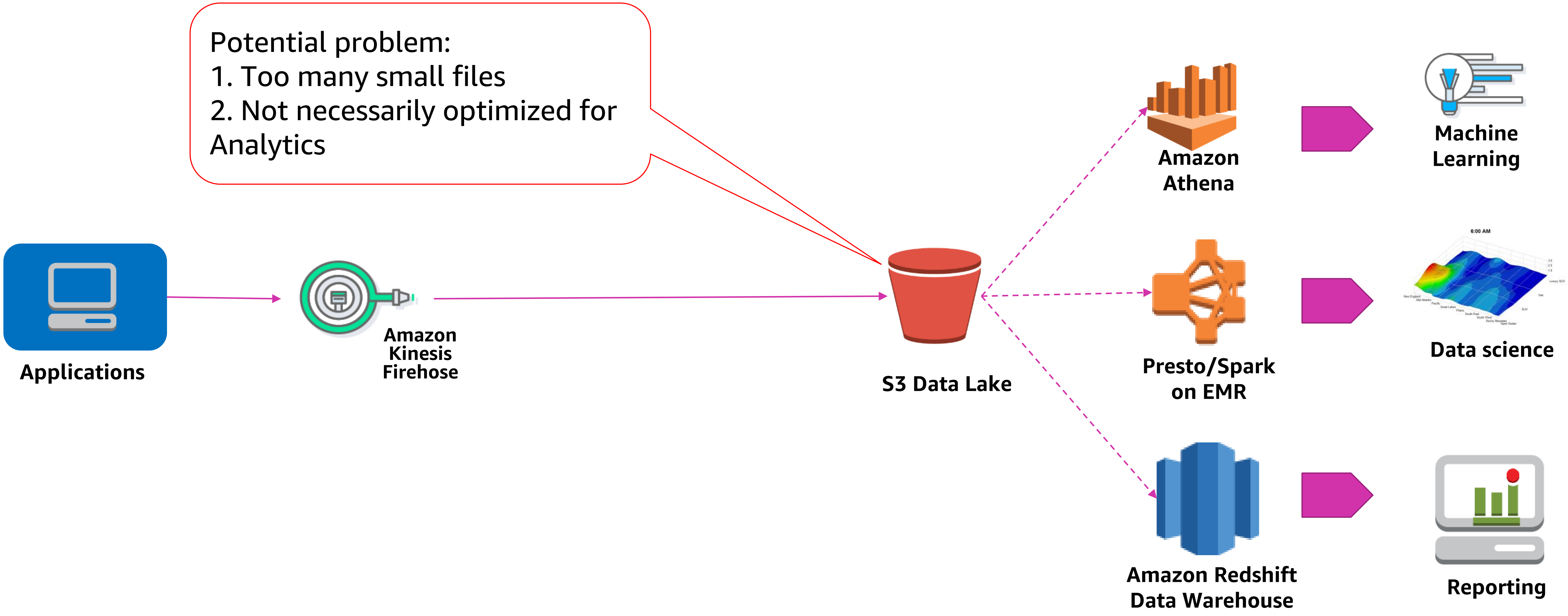
- Can I do streaming ingest into a Data Lake?
- Can a Data Lake replace our database replicas we maintain for analytics?
- How to organize data inside a Data Lake?
- How to handle late events coming in to old partitions?
- How to perform updates and deletes to the data inside a Data Lake?

Some more customer asks

- How can I run Machine Learning training on data in the Data Lake?
- How can I augment the data in my Data Lake with real-time predictions during ETL or ingestion?
- How to enforce data protection rules in the Data Lake?
- What are the authentication and authorization options available?

Data Lake design patterns

Log analytics, ClickStream analytics, IoT sensor data



Potential problem:
1. Too many small files
2. Not necessarily optimized for Analytics

Applications

Amazon Kinesis Firehose

S3 Data Lake

Amazon Athena

Presto/Spark on EMR

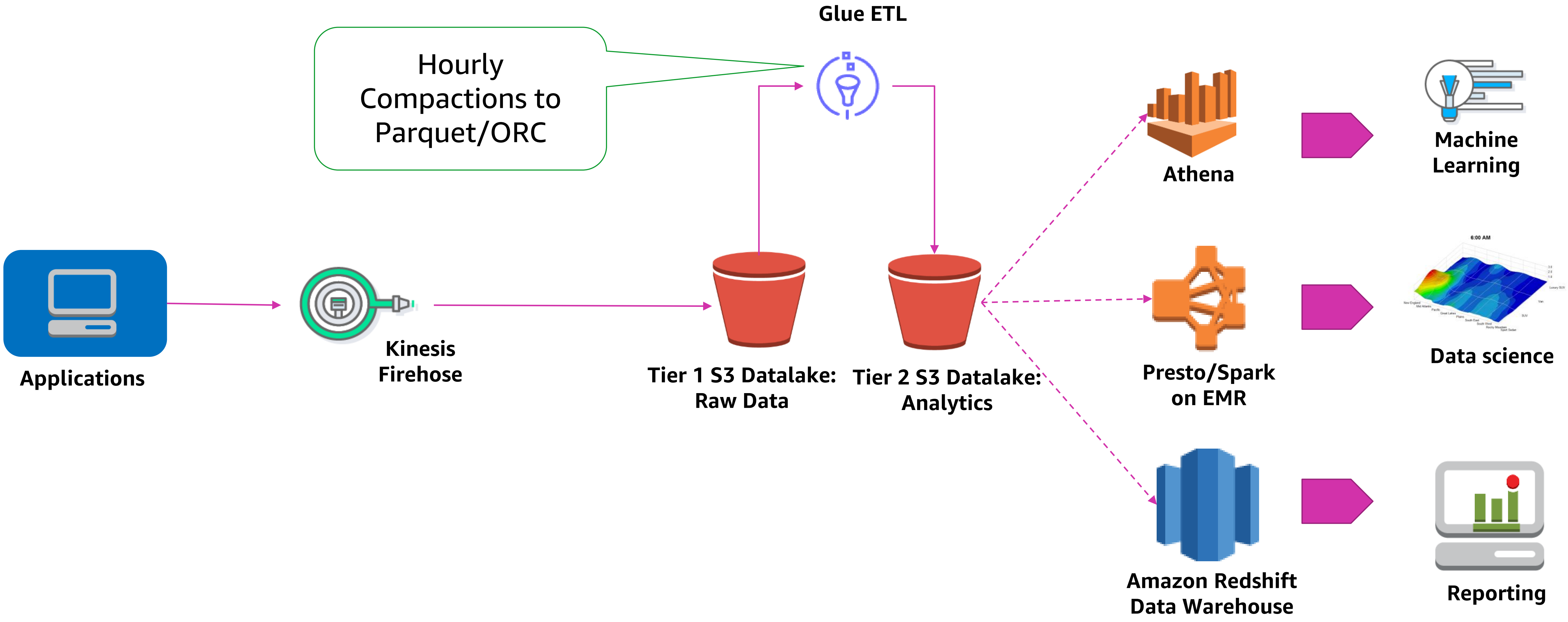
Amazon Redshift Data Warehouse

Machine Learning

Data science

Reporting

Log analytics, ClickStream analytics, IoT sensor data



Hourly Compactions to Parquet/ORC

Glue ETL

Applications

Kinesis Firehose

Tier 1 S3 Datalake: Raw Data

Tier 2 S3 Datalake: Analytics

Athena

Presto/Spark on EMR

Amazon Redshift Data Warehouse

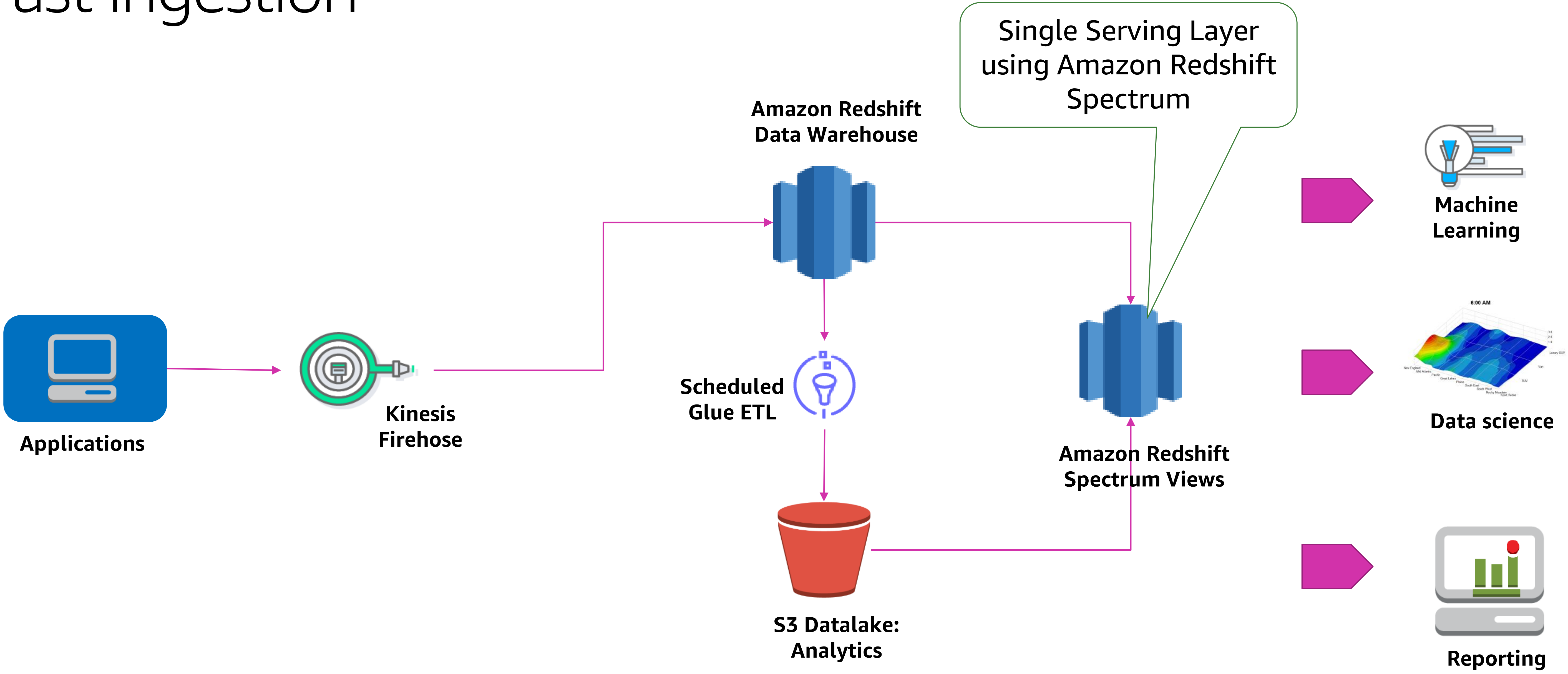
Machine Learning

Data science

Reporting

AWS re:Invent

Fast ingestion



Single Serving Layer using Amazon Redshift Spectrum

Applications

Kinesis Firehose

Amazon Redshift Data Warehouse

Scheduled Glue ETL

S3 Datalake: Analytics

Amazon Redshift Spectrum Views

Machine Learning

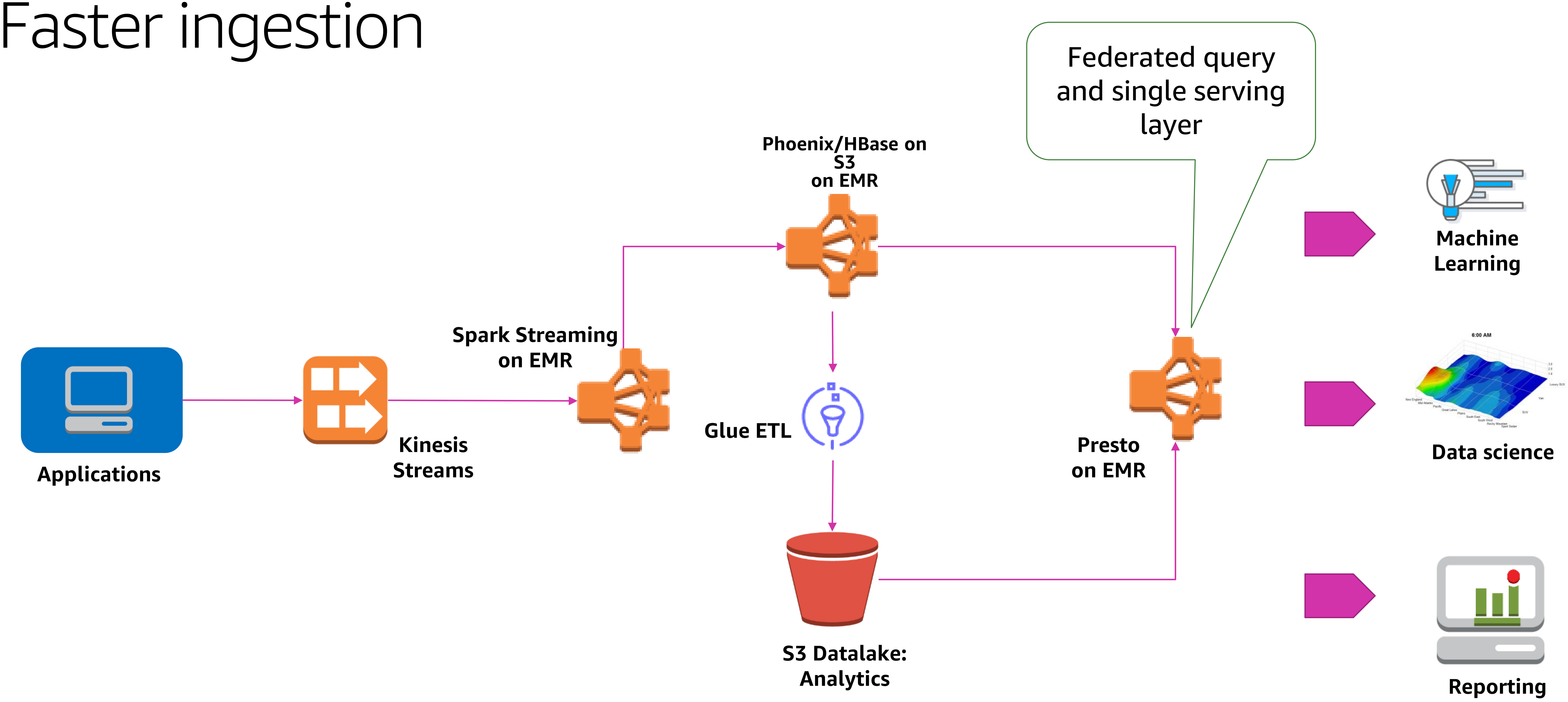
Data science

Reporting

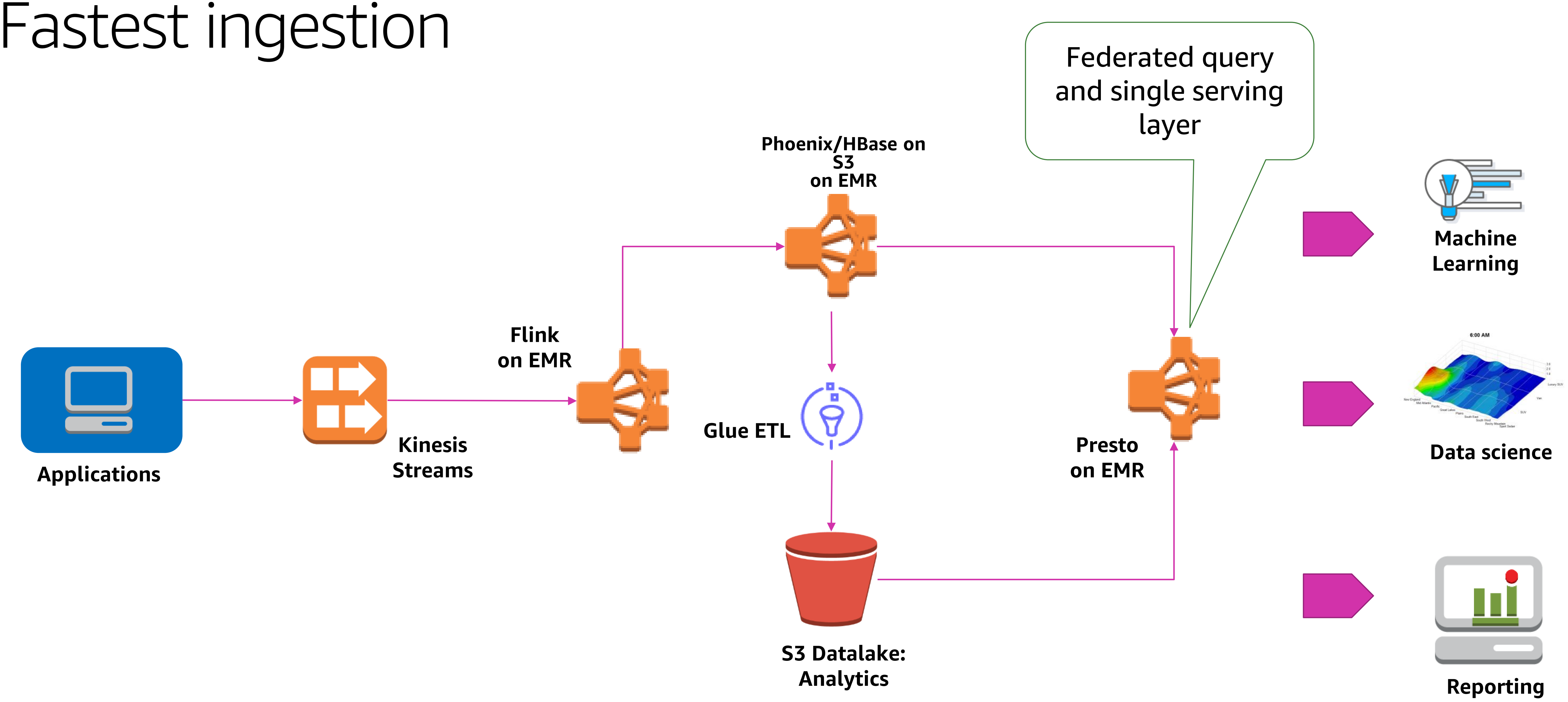
AWS re:Invent

aws

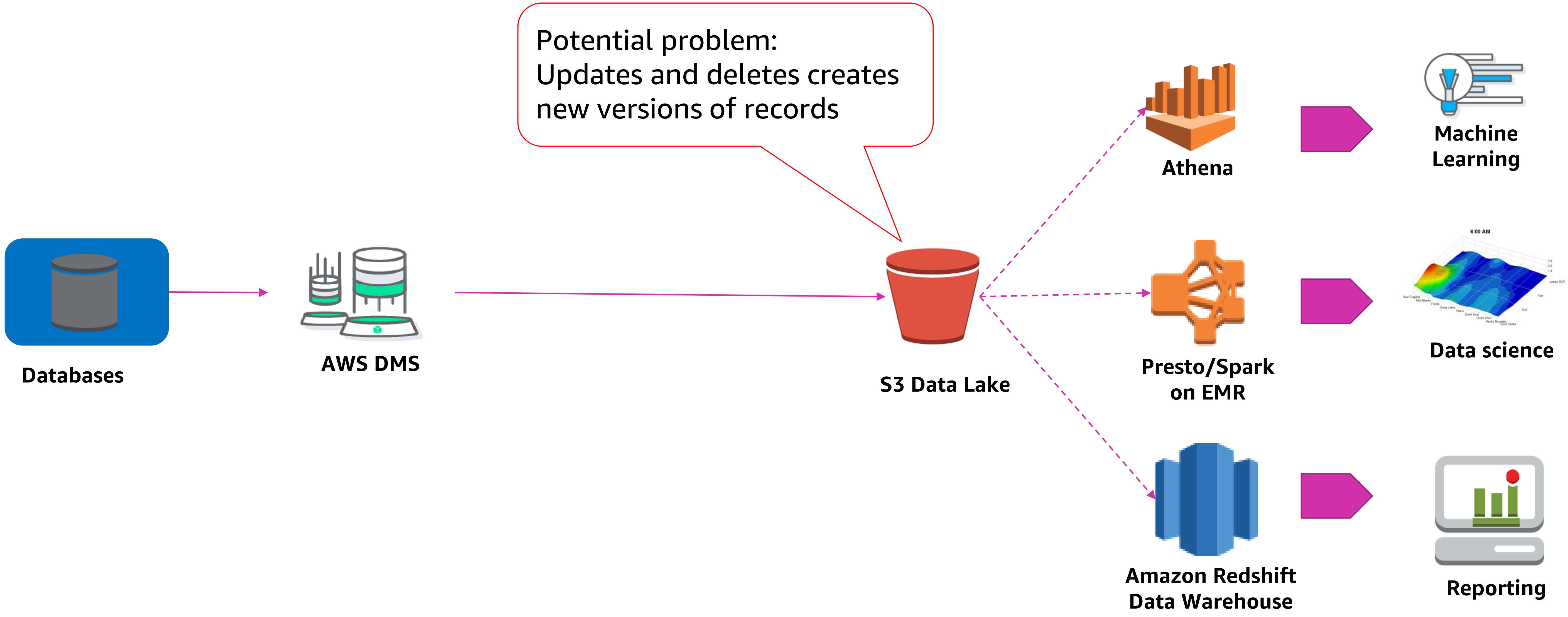
Faster ingestion



Fastest ingestion

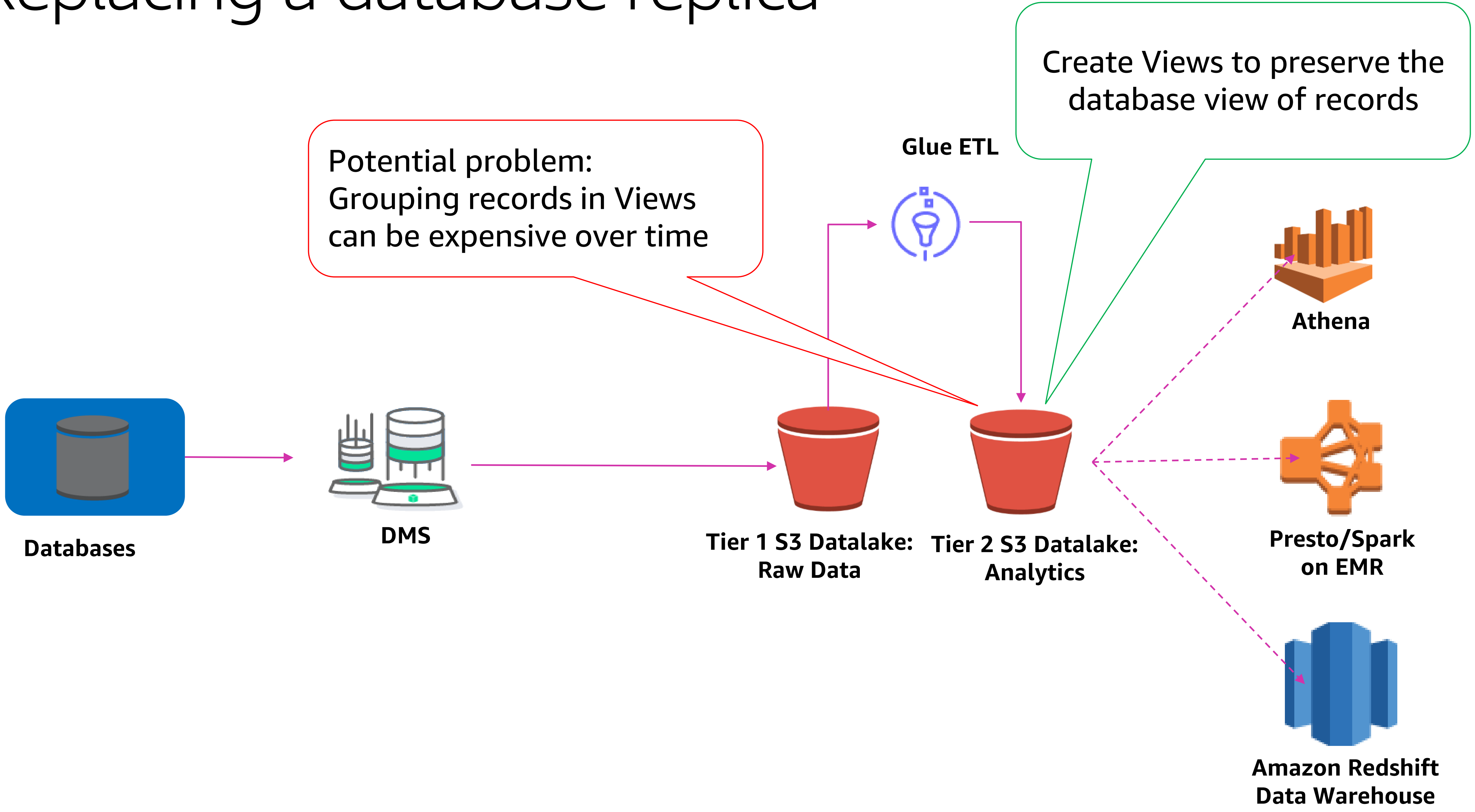


Replacing a database replica

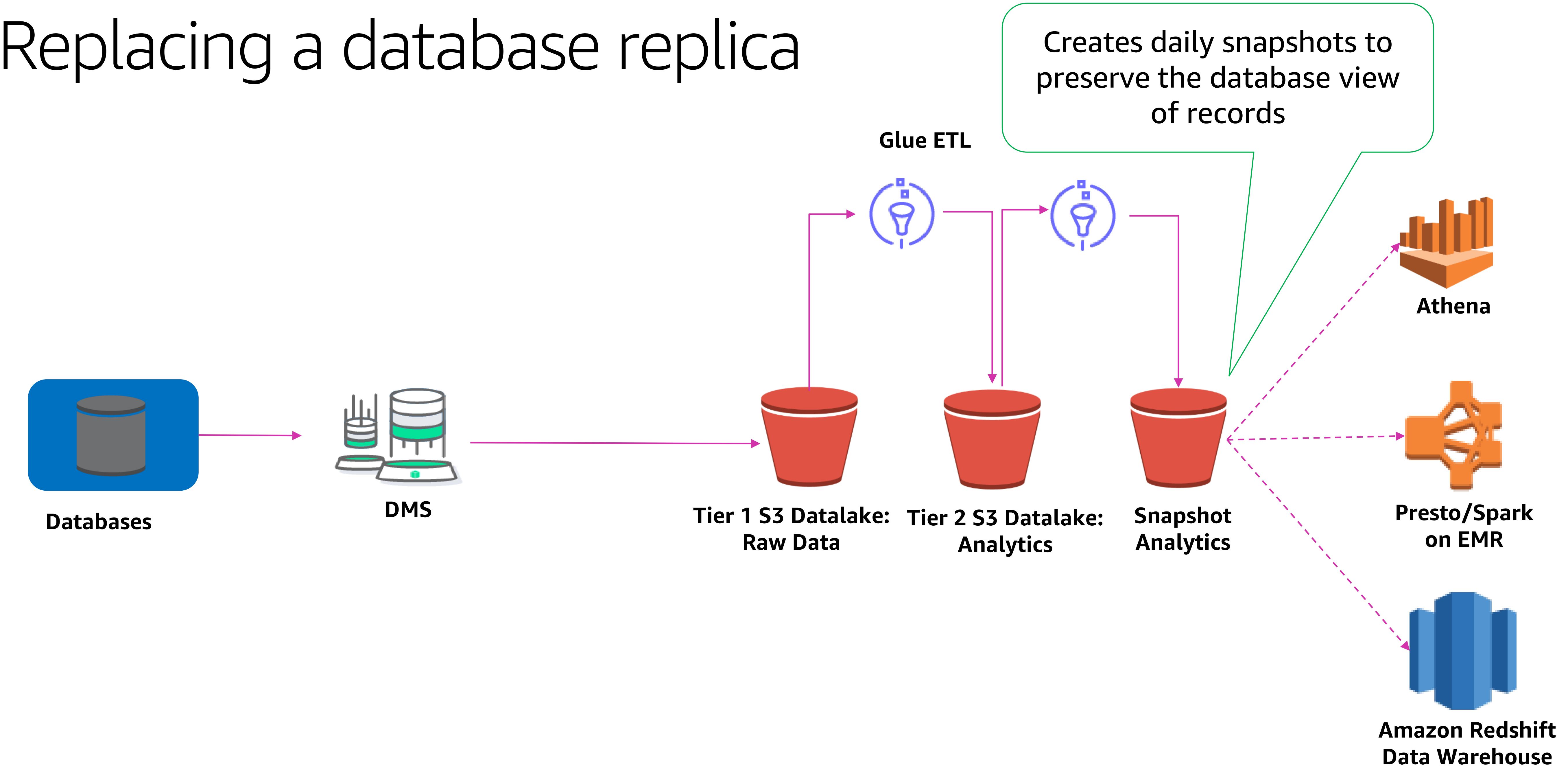


Potential problem:
Updates and deletes creates new versions of records

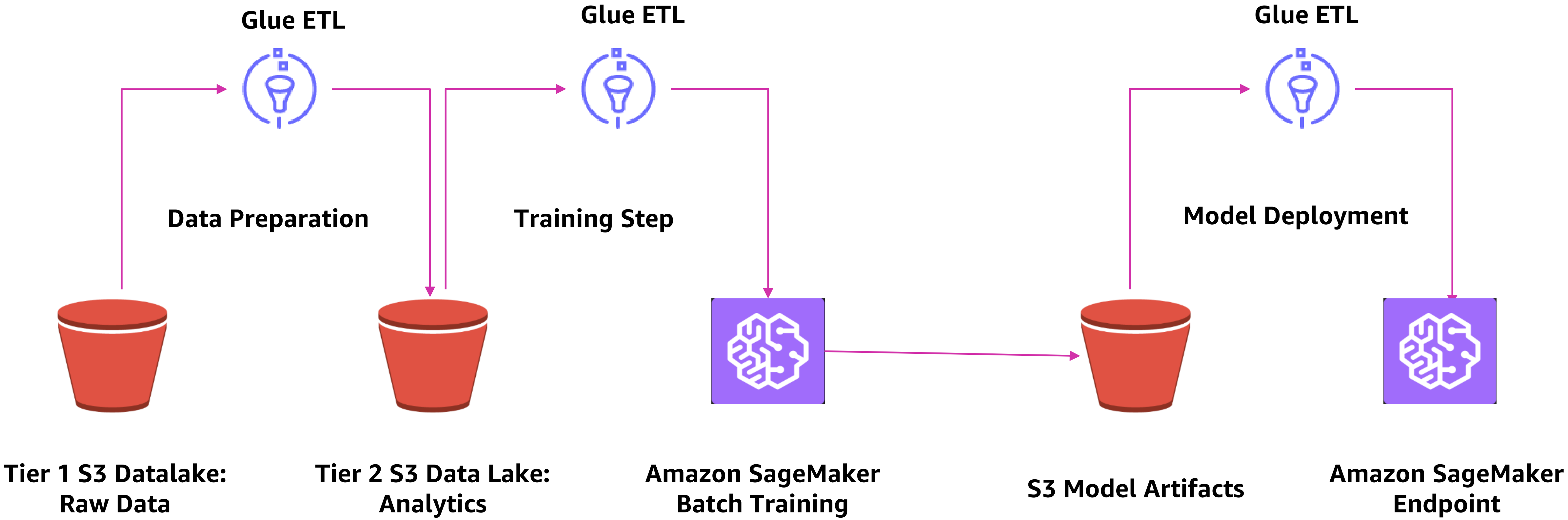
Replacing a database replica


Create Views to preserve the database view of records
Potential problem: Grouping records in Views can be expensive over time
Glue ETL
Athena
Databases
DMS
Tier 1 S3 Datalake: Raw Data
Tier 2 S3 Datalake: Analytics
Presto/Spark on EMR
Amazon Redshift Data Warehouse

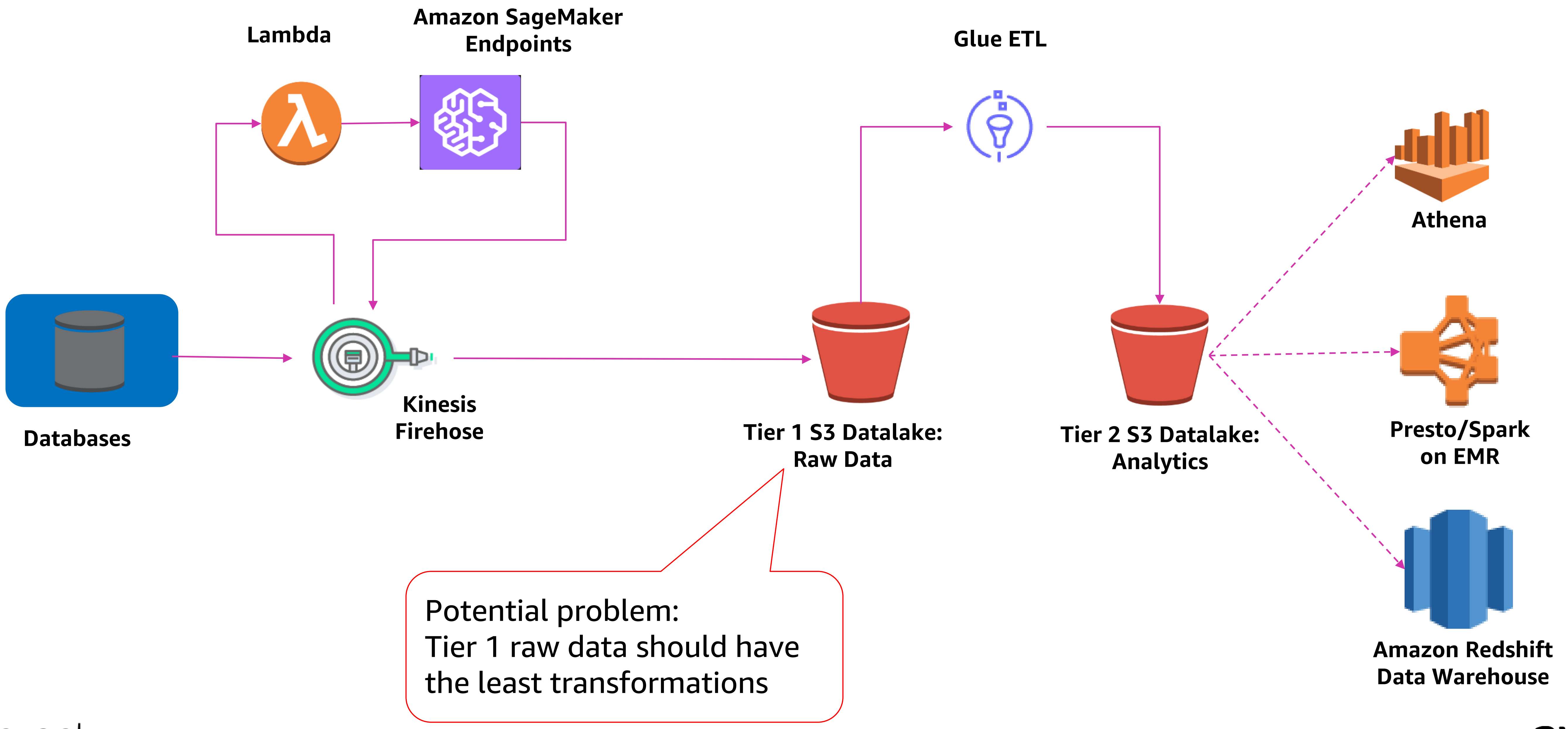
Replacing a database replica


Creates daily snapshots to preserve the database view of records
Glue ETL
Athena
Databases
DMS
Tier 1 S3 Datalake: Raw Data
Tier 2 S3 Datalake: Analytics
Snapshot Analytics
Presto/Spark on EMR
Amazon Redshift Data Warehouse

Machine Learning—Batch training pipeline

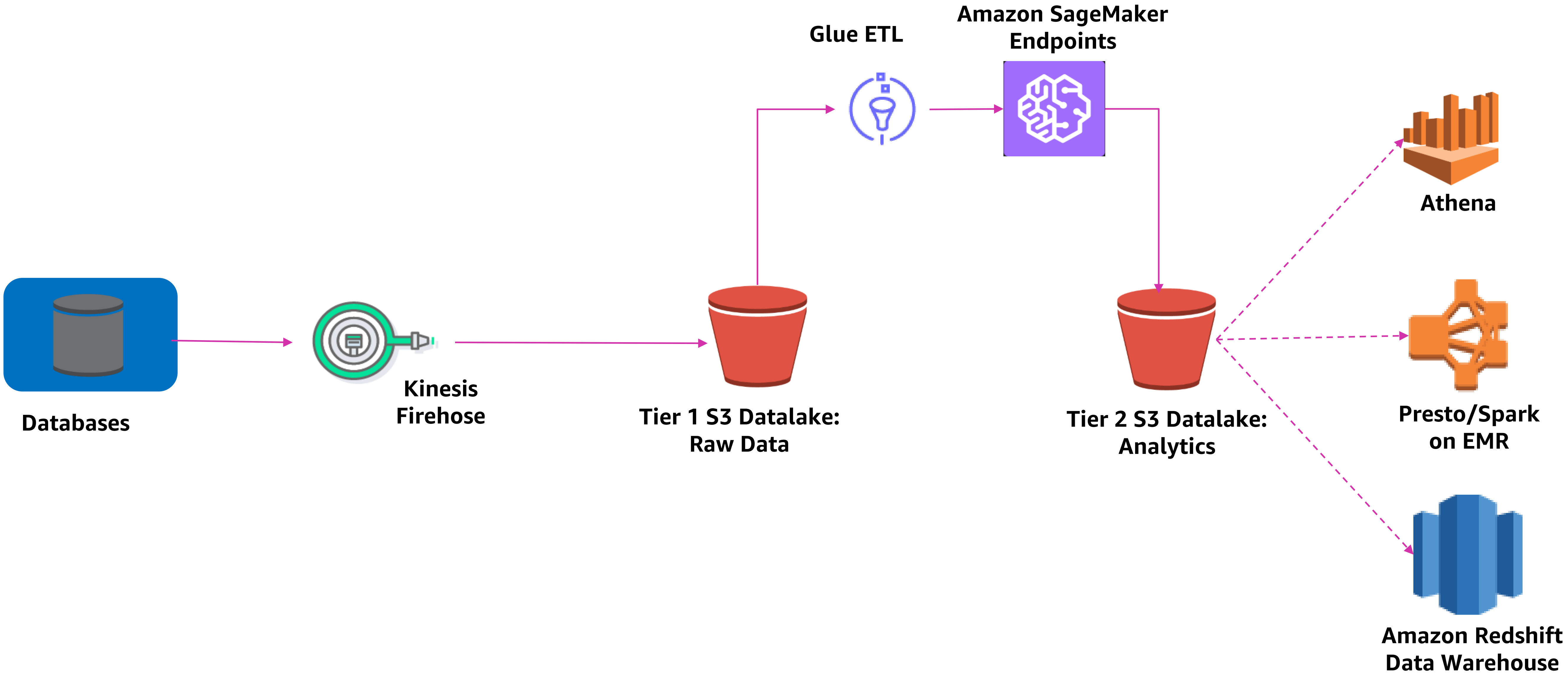


Machine Learning—Predictions on streaming data


Lambda
Amazon SageMaker Endpoints
Glue ETL
Athena
Databases
Kinesis Firehose
Tier 1 S3 Datalake: Raw Data
Tier 2 S3 Datalake: Analytics
Presto/Spark on EMR
Amazon Redshift Data Warehouse

Potential problem:
Tier 1 raw data should have the least transformations

Machine Learning—Predictions on streaming data


Glue ETL
Amazon SageMaker Endpoints
Athena
Databases
Kinesis Firehose
Tier 1 S3 Datalake: Raw Data
Tier 2 S3 Datalake: Analytics
Presto/Spark on EMR
Amazon Redshift Data Warehouse

Data Lake design principles

- **Ingestion location and frequency**: Decide on a location for ingestion. Select a frequency and ingestion mechanism as meets your needs.

- **Partition data**: Partition the data with keys that align with common query filters used. This enables partition pruning and increases query performance.

- **File Size**: Choose optimal file sizes to reduce S3 roundtrips. Recommended : 256 MB to 1GB files in columnar format per partition.

- **Compactions**: Compact data on a scheduled basis to get the file sizes above e.g., daily compactions into daily partitions if hourly files are small.

How to choose partitioning columns?

- Aim for optimum files sizes—256 MB to 1GB
- Identify the typical query scan range—One year, five years, etc.
- Know your query filters and Group By columns that should align with partition columns

How to choose partitioning columns? An example

Use case: Aggregation of time series data

Number of devices: 100

Partition format: device/year/month/day/hour

Data retention/query scan range: Five years

File per partition: One

File Size:10 MB

5*365*24*100= 4.3M partitions

How to choose partitioning columns? An example

Number of devices: 100

Partition format: year/month/day/~~hour~~

Bucketed by: Device, 50 buckets

Data retention/query scan: Five years

File per partition: 50

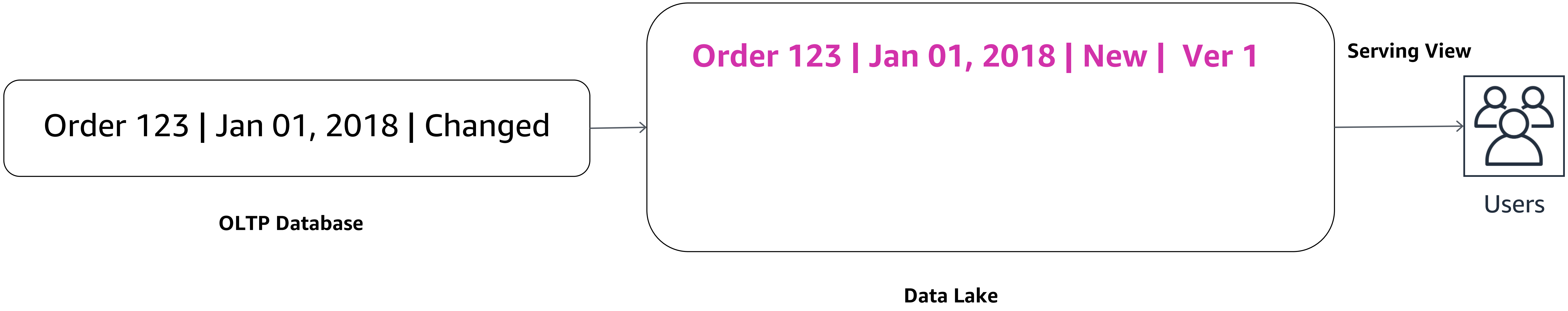
File size: 480 MB

5*365 = 1825 partitions

Data Lake design principles

- **Mutable data**: For mutable uses cases i.e., to handle updates/deletes
  - Either use a database like Amazon Redshift/HBase for the time the data can mutate and offload to S3 once data becomes static
  - Or append to delta files per partition and compact on a scheduled basis using AWS Glue or Spark on EMR

Serving mutable data



Order 123 | Jan 01, 2018 | Changed

OLTP Database

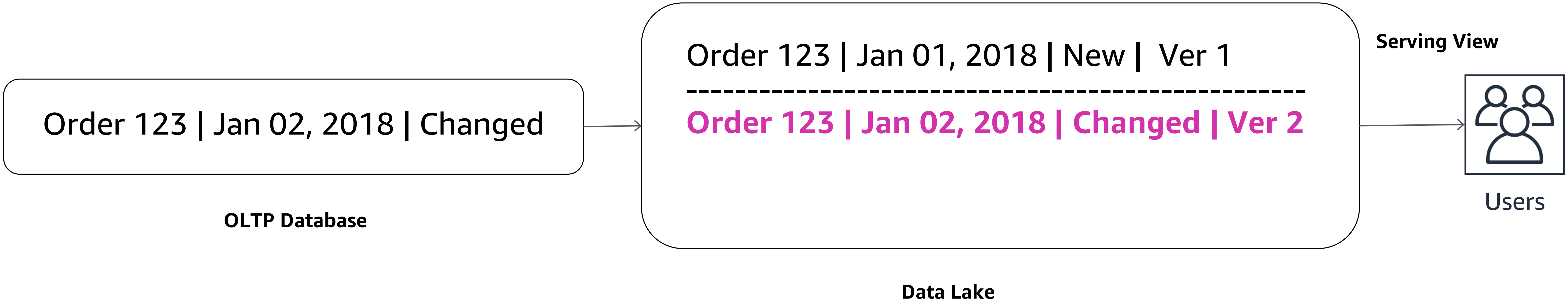
Order 123 | Jan 01, 2018 | New | Ver 1

Data Lake

Serving View

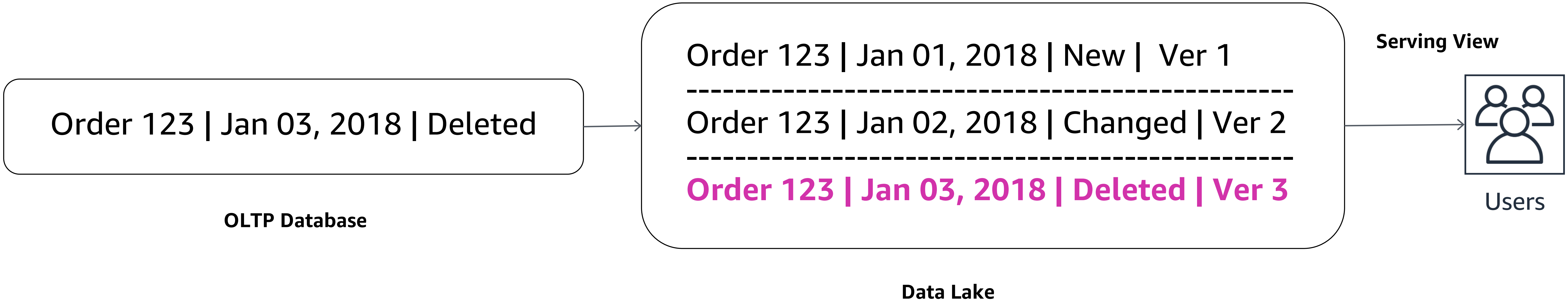
Users

Serving mutable data


Order 123 | Jan 02, 2018 | Changed
OLTP Database
Order 123 | Jan 01, 2018 | New | Ver 1

Order 123 | Jan 02, 2018 | Changed | Ver 2
Data Lake
Serving View
Users

Serving mutable data



Order 123 | Jan 03, 2018 | Deleted

OLTP Database

Order 123 | Jan 01, 2018 | New | Ver 1

Order 123 | Jan 02, 2018 | Changed | Ver 2

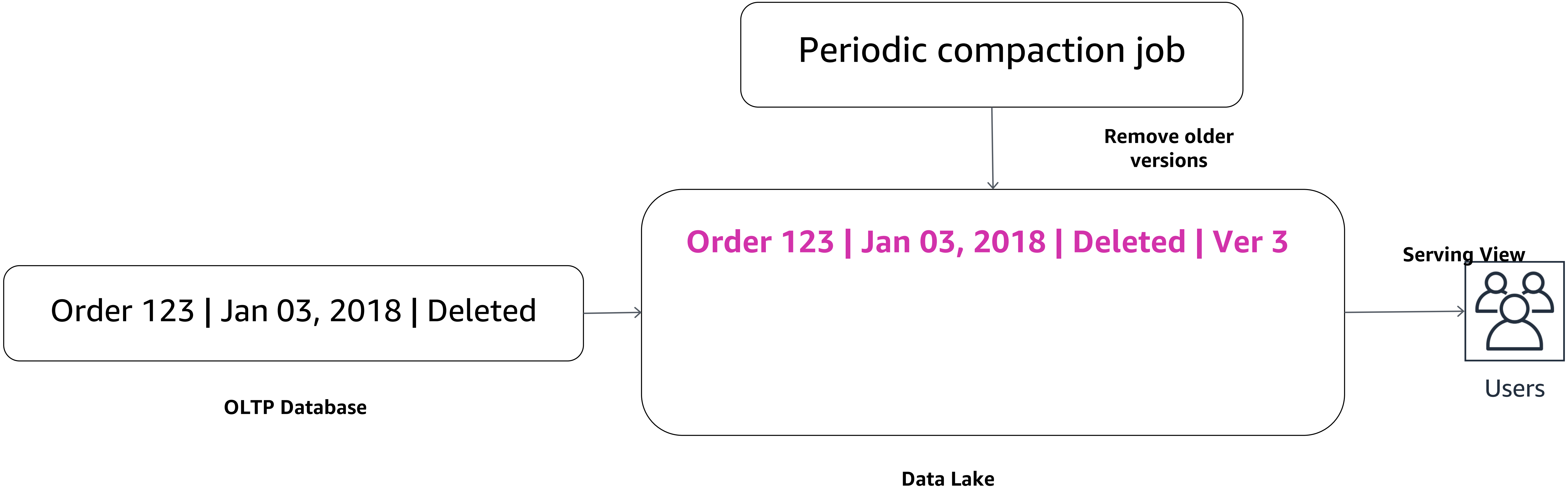
Order 123 | Jan 03, 2018 | Deleted | Ver 3

Data Lake

Serving View

Users

Serving mutable data



Periodic compaction job

Remove older versions

Order 123 | Jan 03, 2018 | Deleted

OLTP Database

Order 123 | Jan 03, 2018 | Deleted | Ver 3

Data Lake

Serving View

Users

Data Lake optimizations

- **Bucketed data**: For additional performance, bucket data in each partition on a high cardinality key. This is honored by Presto/Athena, Hive and so on, and improves query filter performance on that key.

```
df.write.bucketBy(numBuckets,”col1”).parquet(…)
```

- **Order Data**: For additional performance, sort data in each partition by a secondary key. This allows engines to skip part of files to get to the requested data faster.

```
df.repartition(100).sortWithinPartitions([‘order_id’]
,ascending=True).parquet(…)
```

Data Lake optimizations

- **Bloom filters**: Bloom Filters are space-efficient probabilistic data structures that is used to test whether an element is a member of a set

```
CREATE TABLE
STORED AS ORC
TBLPROPERTIES('orc.bloom.filter.columns’=‘ORDER_ID')
```

Security and governance patterns

Security and governance concerns

- Authentication
- Authorization on data (and metadata)
- Encryption of data at rest and in transit
- Audit and monitoring
- Centralized management
- Compliance

AWS helps you secure

Customer need to have multiple levels of security, identity and access management, encryption, and compliance to secure their data lake

Security

Amazon GuardDuty

AWS Shield

AWS WAF

Amazon Macie

Amazon VPC

Identity

AWS IAM

AWS SSO

Amazon Cloud Directory

AWS Directory Service

AWS Organizations

Encryption

AWS Certification Manager

AWS Key Management Service

Encryption at rest

Encryption in transit

Bring your own keys, HSM support

Compliance

AWS Artifact

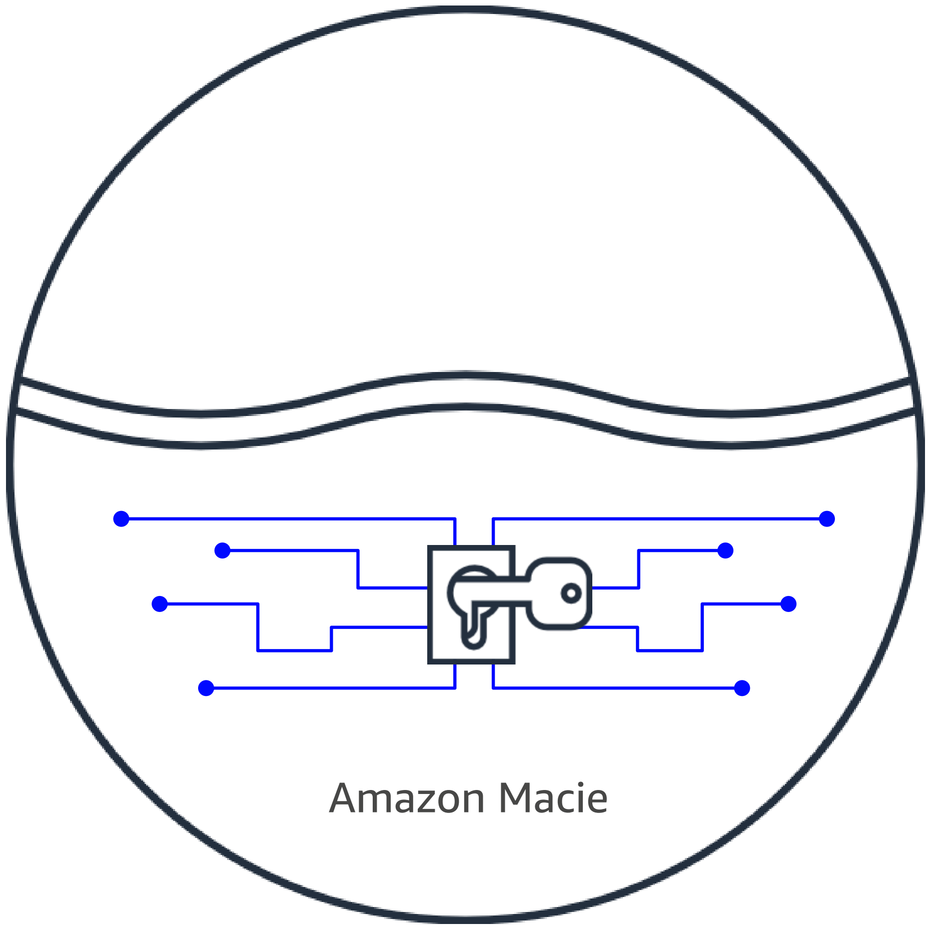
Amazon Inspector

Amazon Cloud HSM

Amazon Cognito

AWS CloudTrail

Security: Machine Learning-powered security



- Machine learning to discover, classify, and protect data
- Continuously monitors data access for anomalies
- Generates alerts when it detects unauthorized access
- Recognizes PII or intellectual property

Data Lake security

- Data storage
- Metadata

Data storage security

Data storage security

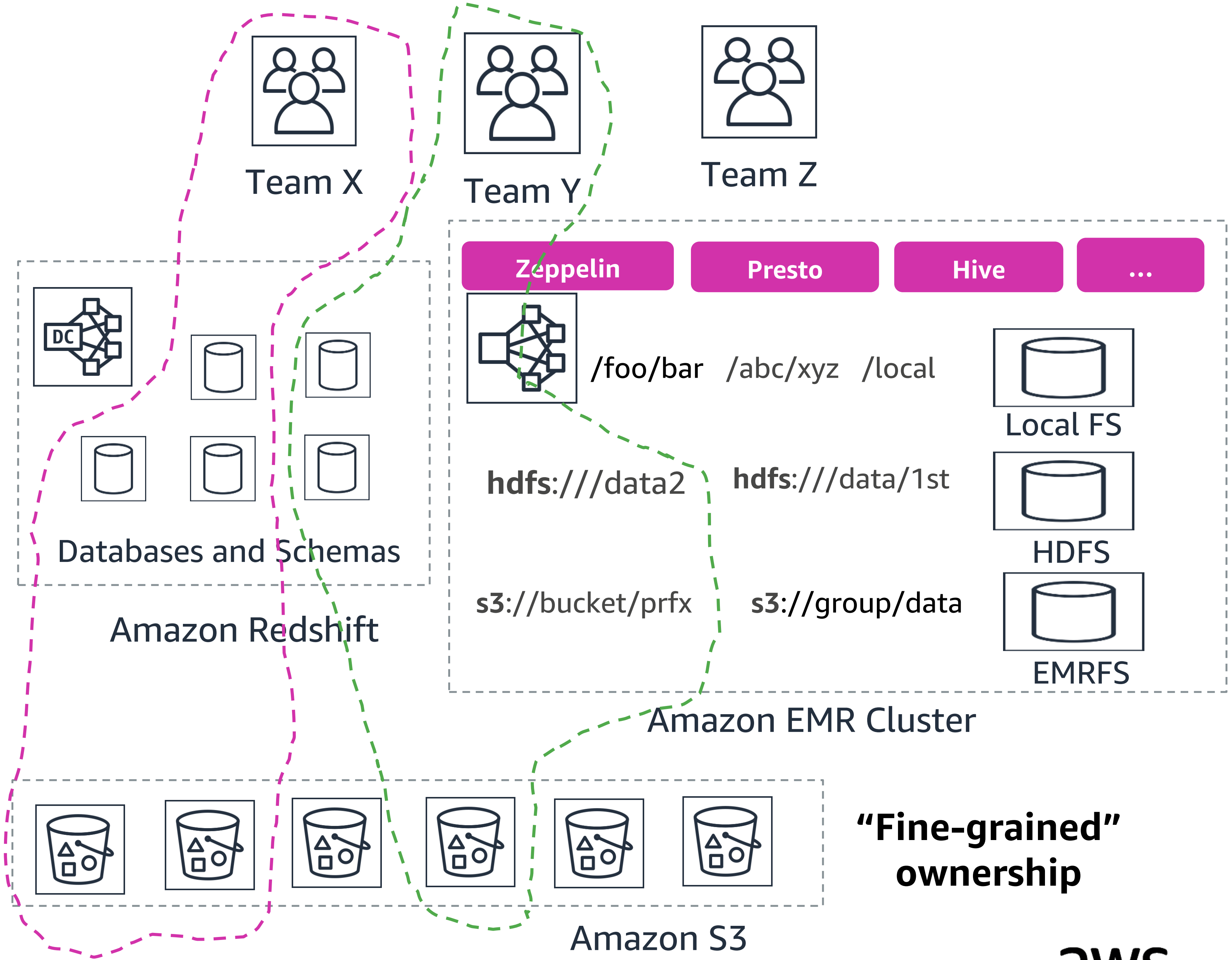
Key learnings

- Implement access control in a multi-team environment
  - Fine-grained
  - Coarse-grained
- Secure and segregated access to
  - Amazon S3
  - Amazon EMR clusters
  - Amazon Redshift clusters
  - Serverless analytics services and other tools used in the pipeline
- Encrypt data assets

Control access to data—Fine-grained ACL

“Fine-grained” data and resource ownership

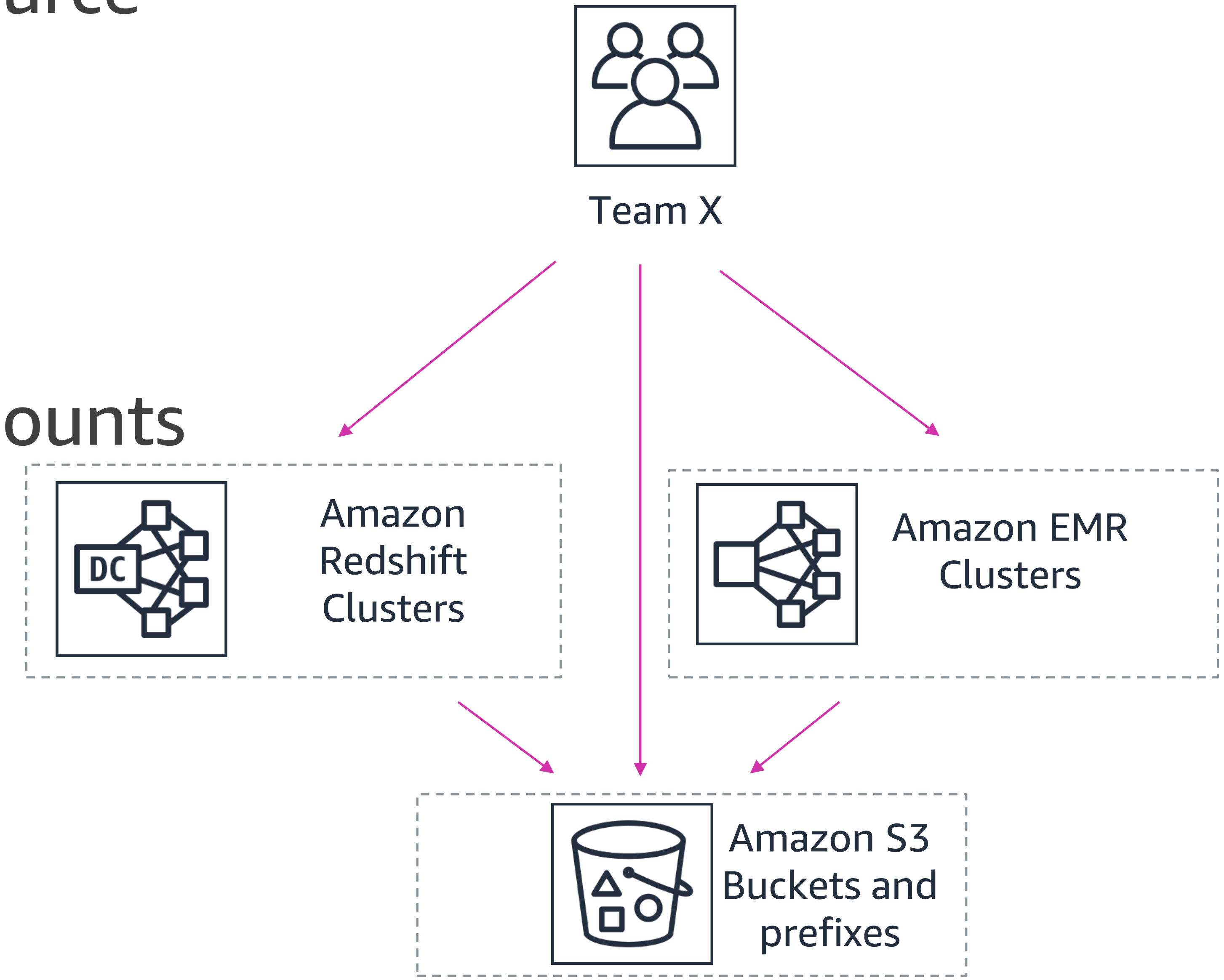
- Teams **share S3 buckets and clusters**
- Access control complex to set up and maintain
- Common in a “**shared services**” architecture



Control data access—Coarse grained

Prefer "coarse-grained" data and resource ownership

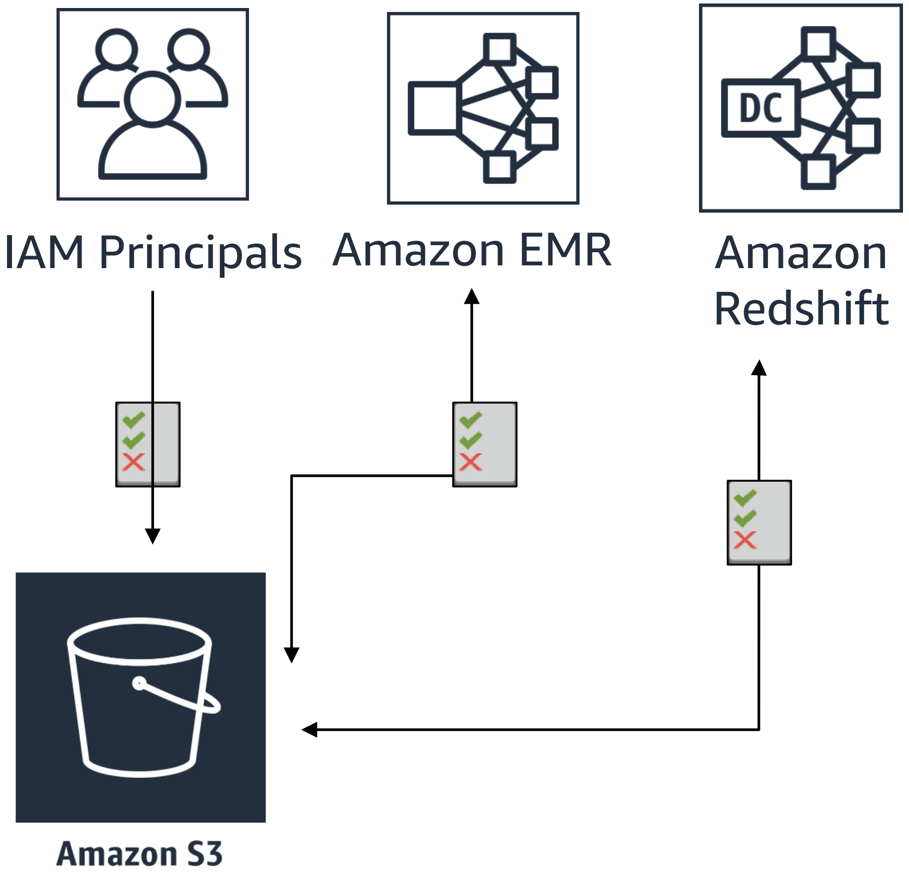
- Teams own **entire S3 buckets and clusters**
- Ownership segregated by AWS accounts
- Access control easier to setup and maintain
- Suitable for **autonomous teams**



Control access to data

Configure **Amazon S3** permissions

- Implement your access control matrix using **IAM policies**
- Use **S3 bucket policies** for easy cross-account data sharing
- Limit role-based access from an **Amazon EMR** cluster’s **Amazon Elastic Compute Cloud** (Amazon EC2) **instance profile**
- Authorize access from other tools such as **Amazon Redshift** using IAM roles



Block public access to Amazon S3

Amazon S3 provides four settings

- `BlockPublicAcls` – **rejects new** public object or bucket ACLs
- `IgnorePublicAcls` – **ignores existing** public object or bucket ACLs
- `BlockPublicPolicy` – **rejects new** public bucket access policy
- `RestrictPublicBuckets` – **restricts access** to only AWS services and authorized users within the bucket owner's account

But, what is “public”?

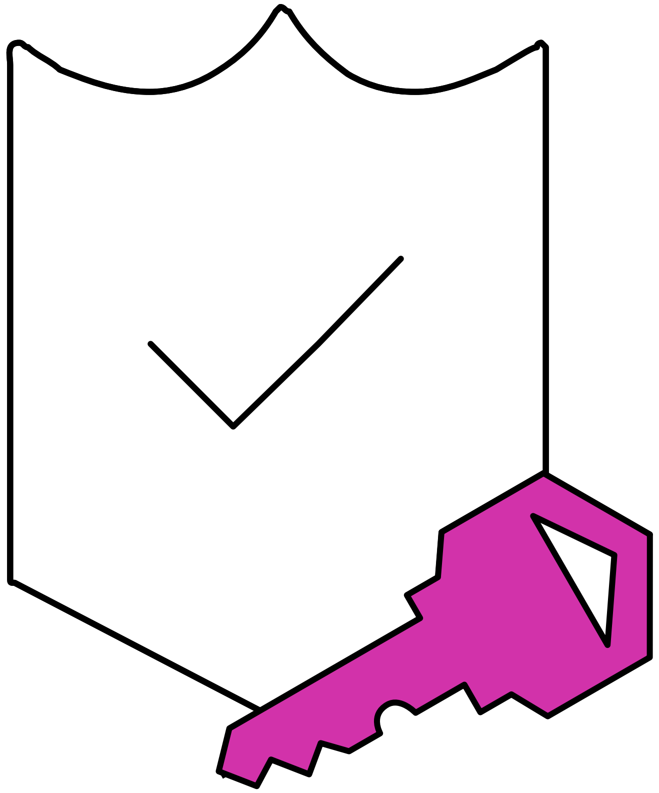
- **Public object (or bucket) ACL** → grants permissions to members of the predefined ***AllUsers*** or ***AuthenticatedUsers*** groups (grantees)

- **Public bucket policy** → **doesn’t** grant permissions to **only fixed values in Principal** and **Condition** elements

AWS re:Invent

aws

Encryption: Data-at-rest and in-motion



- Amazon S3 offers multiple forms of encryption
  - Server-side and Client-side encryption
  - Encryption with keys managed by S3 or AWS Key Management Service
  - Encryption with keys that customers manage
- Encrypts data in transit when replicating across regions
- Data movement services can use the same AWS Key Management Service
- SSL endpoints

Metadata security

Metadata security

AWS Glue Data Catalog

- **Apache Hive metastore compatible**
- Track data evolution using **schema versioning**
- **Integrates with** Hive, Spark, Presto, Amazon Athena and Amazon Redshift spectrum
- Use crawlers **classify** your data in one central list that is **searchable**

AWS Glue

Metadata security

Key learnings

- Create and maintain centralized data catalog
- Enable cross account access
- Use IAM policies to control catalog access—similar to S3 bucket policies
- Encrypt metadata in AWS Glue Data Catalog

Glue Data Catalog—Cross account access

Catalog Policy in Account A

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:GetDatabase"
      ],
      "Principal": {"AWS": [
        "arn:aws:iam::account-B-id:user/Bob"
      ]},
      "Resource": [
        "arn:aws:glue:us-east-1:account-A-id:catalog",
        "arn:aws:glue:us-east-1:account-A-
id:database/db1"
      ]
    }
  ]
}
```

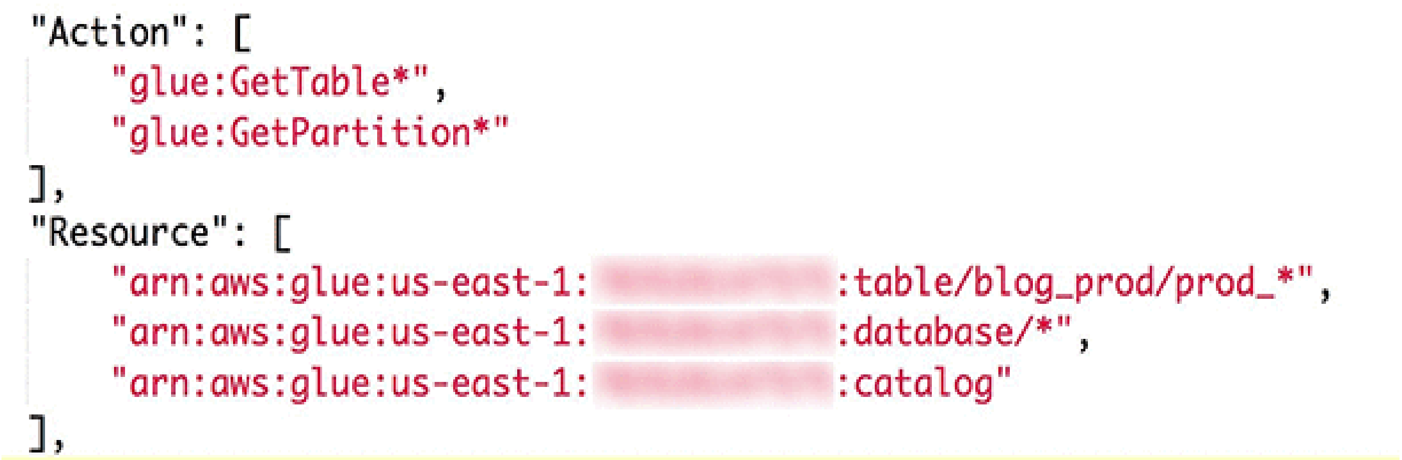
Bob’s IAM Policy in Account B

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:GetDatabase"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:account-A-
id:catalog",
        "arn:aws:glue:us-east-1:account-A-
id:database/db1"
      ]
    }
  ]
}
```

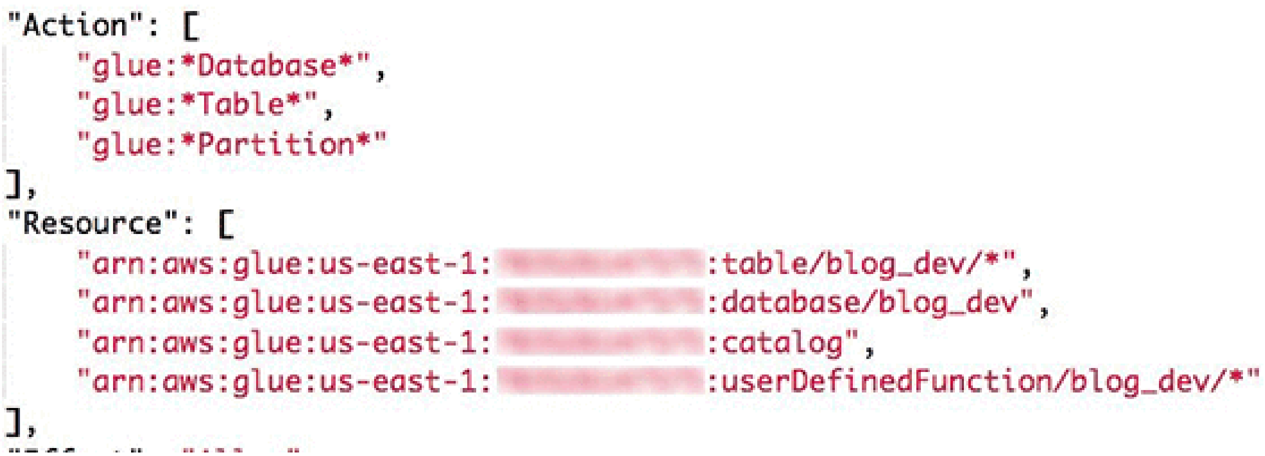
Glue Data Catalog—Resource-level permissions

AWS Glue

- Fine-grained access control to catalog using IAM policies
- Restrict what they can view and query



```
"Action": [
    "glue:GetTable*",
    "glue:GetPartition*"
],
"Resource": [
    "arn:aws:glue:us-east-1:[redacted]:table/blog_prod/prod_*",
    "arn:aws:glue:us-east-1:[redacted]:database/*",
    "arn:aws:glue:us-east-1:[redacted]:catalog"
],
```



```
"Action": [
    "glue:*Database*",
    "glue:*Table*",
    "glue:*Partition*"
],
"Resource": [
    "arn:aws:glue:us-east-1:[redacted]:table/blog_dev/*",
    "arn:aws:glue:us-east-1:[redacted]:database/blog_dev",
    "arn:aws:glue:us-east-1:[redacted]:catalog",
    "arn:aws:glue:us-east-1:[redacted]:userDefinedFunction/blog_dev/*"
],
```

Security and Governance

	Athena	EMR	Glue	Redshift
Authentication	IAM/EC2 Key pair	Kerberos/LDAP/ EC2 Key pair/IAM	IAM Role	IAM/Native
Authorization	S3 Bucket Policies	S3 Bucket Policies/ Hive Grants/ EMRFS Auth	S3 Bucket Policies/ Fine Grained	S3 Bucket Policies/ Native Grants
Encryption of data at-rest	SSE-S3/ SSE-KMS/ CSE-KMS	SSE-S3/ SSE-KMS/ CSE-KMS/ CSE-CMK	SSE-S3	Database Encryption/ SSE-S3/ SSE-KMS/ CSE-CMK
Encryption of data in-transit	SSL	Yes, through Security Config	SSL	SSL
Audit	CloudTrail	Application Logs	CloudTrail	Database Audit
Compliance	HIPAA	FedRAMP/HIPAA	HIPAA	FedRAMP/HIPAA

Compliance: Virtually every regulatory agency

Global

CSA
Cloud Security Alliance Controls

ISO 9001
Global Quality Standard

ISO 27001
Security Management Controls

ISO 27017
Cloud Specific Controls

ISO 27018
Personal Data Protection

PCI DSS Level 1
Payment Card Standards

SOC 1
Audit Controls Report

SOC 2
Security, Availability, & Confidentiality Report

SOC 3
General Controls Report

United States

CJIS
Criminal Justice Information Services

DoD SRG
DoD Data Processing

FedRAMP
Government Data Standards

FERPA
Educational Privacy Act

ISO FFIEC
Financial Institutions Regulation

FIPS
Government Security Standards

FISMA
Federal Information Security Management

GxP
Quality Guidelines and Regulations

HIPPA
Protected Health Information

ITAR
International Arms Regulations

MPAA
Protected Media Content

NIST
National Institute of Standards and Technology

SEC Rule 17a-4(f)
Financial Data Standards

VPAT/Section 508
Accountability Standards

Asia Pacific

FISC [Japan]
Financial Industry Information Systems

IRAP [Australia]
Australian Security Standards

K-ISMS [Korea]
Korean Information Security

MTCS Tier 3 [Singapore]
Multi-Tier Cloud Security Standard

My Number Act [Japan]
Personal Information Protection

Europe

C5 [Germany]
Operational Security Attestation

Cyber Essentials Plus [UK]
Cyber Threat Protection

G-Cloud [UK]
UK Government Standards

IT-Grundschutz [Germany]
Baseline Protection Methodology

AWS Lake Formation

Build a secure data lake in days

Register existing data or load new data using blueprints. Data stored in Amazon S3.

Secure data access across multiple services using single set of permissions.

No additional charge. Only pay for the underlying services used.

Quickly build data lakes

Move, store, catalog, and clean your data faster. Use ML transforms to de-duplicate data and find matching records.

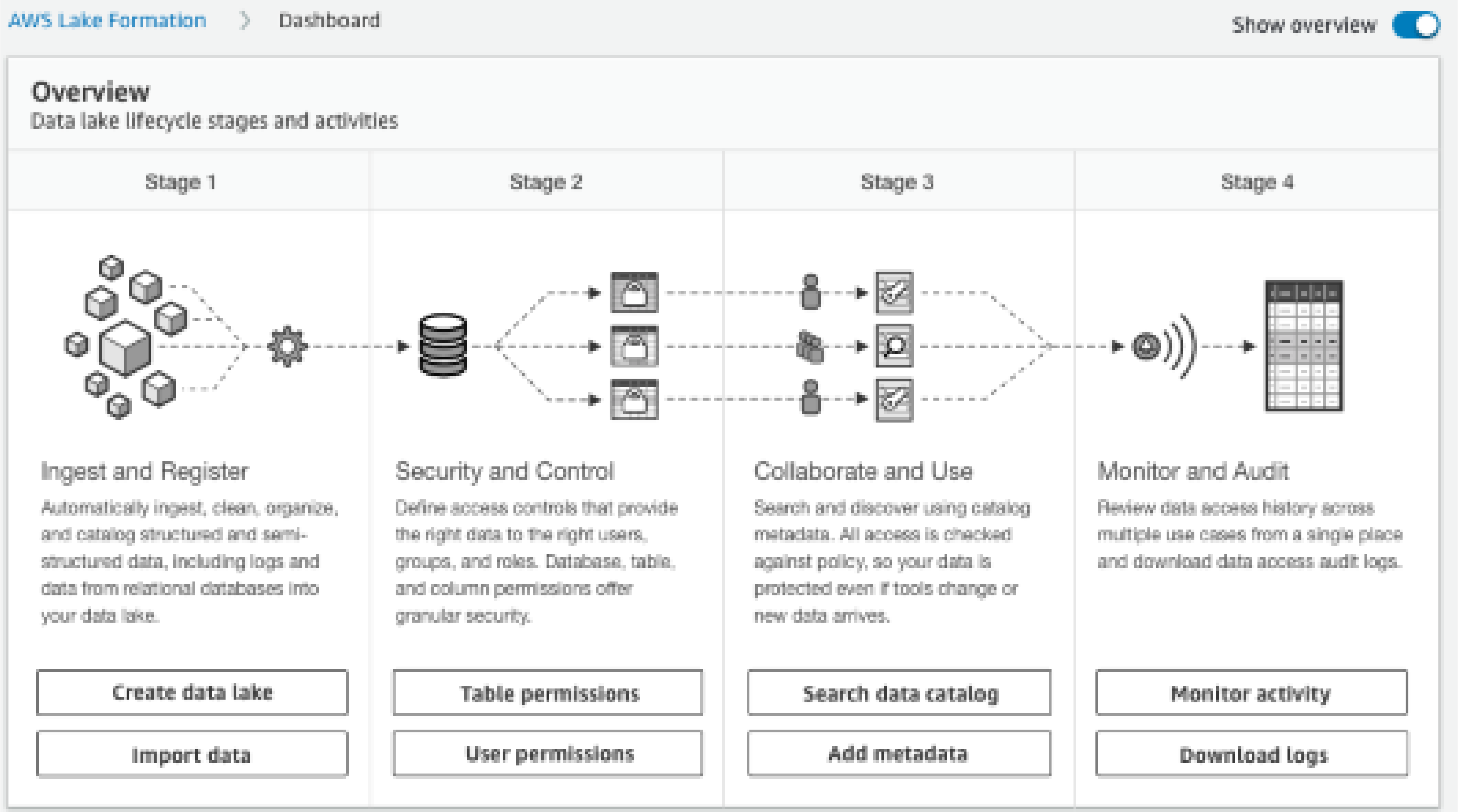
Easily secure access

Centrally define table and column-level data access and enforce it across Amazon EMR, Amazon Athena, Amazon Redshift Spectrum, Amazon SageMaker, and Amazon QuickSight

Share and collaborate

Use data catalog in Lake Formation to search and find relevant data sets and share them across multiple users and accounts

How it works



Overview

Data lake lifecycle stages and activities

Stage 1	Stage 2	Stage 3	Stage 4
Ingest and Register Automatically ingest, clean, organize, and catalog structured and semi-structured data, including logs and data from relational databases into your data lake.	**Security and Control** Define access controls that provide the right data to the right users, groups, and roles. Database, table, and column permissions offer granular security.	**Collaborate and Use** Search and discover using catalog metadata. All access is checked against policy, so your data is protected even if tools change or new data arrives.	**Monitor and Audit** Review data access history across multiple use cases from a single place and download data access audit logs.

Thank you!

Radhika Ravirala
ravirala@amazon.com

Moataz Anany
moanany@amazon.com

!

Please complete the session survey in the mobile app.