aws SUMMIT ONLINE

TRA04

Mastering your data journey one step at a time

Rada Stanic

Principal Solutions Architect
Amazon Web Services

aws SUMMIT ONLINE

Agenda

Current data challenges

Introducing data lakes

Security and governance patterns

AWS Lake Formation – making things easy

Data lake design patterns

Phased architecture build

Current data challenges

Decision making used to…

… revolve around the **Enterprise Data Warehouse** (in the 90s – 00s)

Data no longer fits

There is more data than people think

Data is more diverse

Wider range of workloads

Data scientists

Business users

Analysts

Applications

Machine learning

SQL analytics

Scientific

Real-time, streaming

There are more people accessing data

That want to analyse it in different ways

And there are more rules around data use

Introducing data lakes

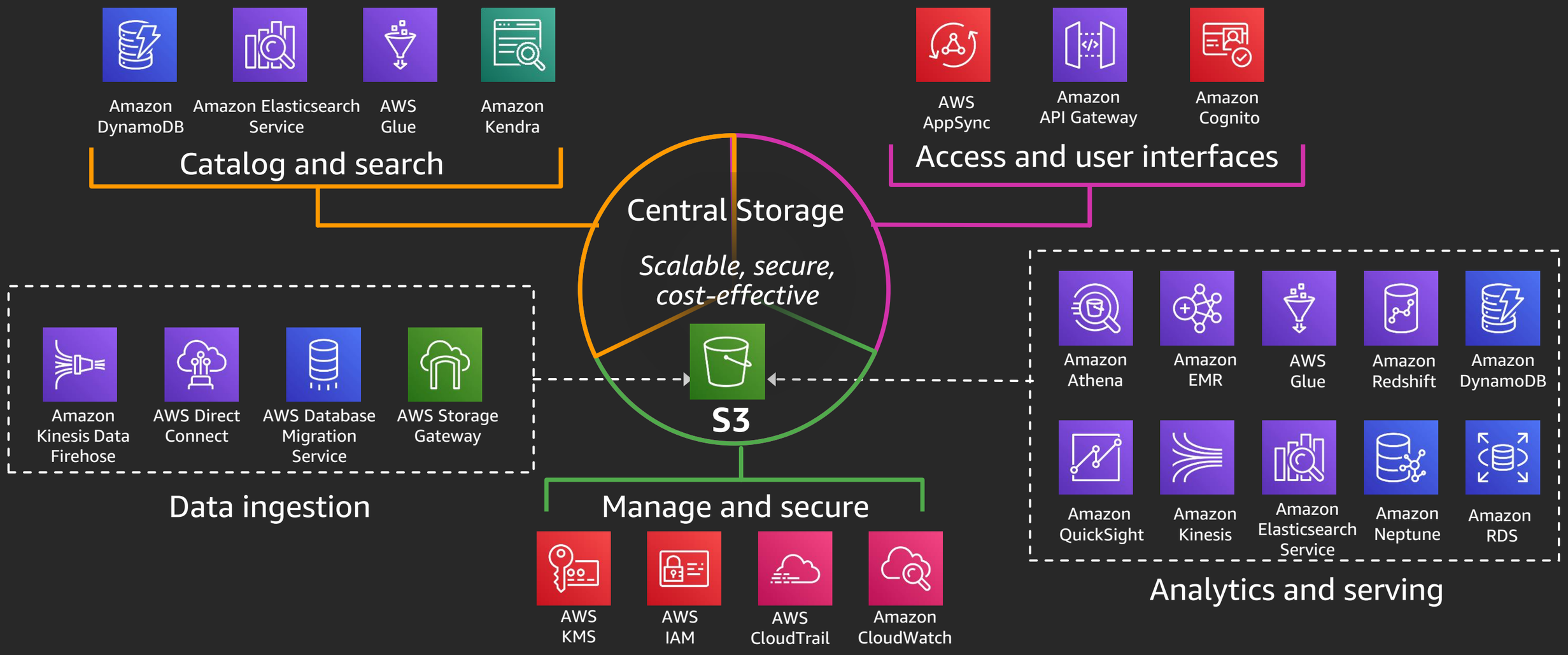
The data lake is the new information hub

A centralised repository that enables you to secure, discover, share, and analyse structured and unstructured data at any scale

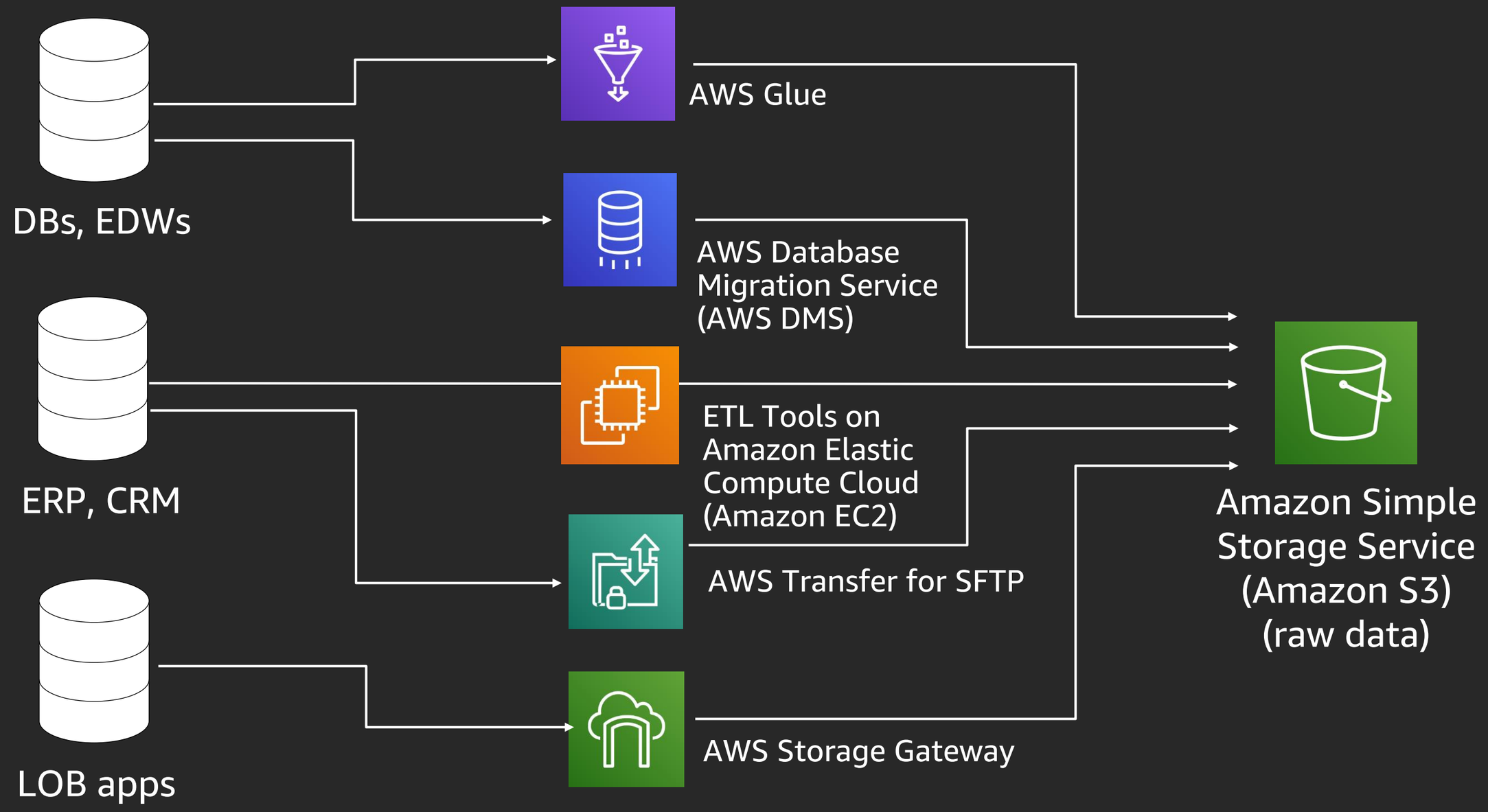
The concept of a data lake

- 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

Building a data lake on AWS



Options for structured data ingestion



The core of a data lake

Versatile compute layers

Amazon Athena

Amazon EMR

Amazon Redshift Spectrum

Data lake

Data and metadata

Amazon S3

AWS Glue Data Catalog

Tiered approach

Amazon S3

Tier 1 data lake: Ingestion

- Single source of truth for raw data
- Organised by Ingestion Time
- Use least transformations
- Use lifecycle policies to Amazon S3 IA or Amazon S3 Glacier

Tiered approach

Amazon S3

Tier 2 data lake: Analytics

- Use columnar formats – Parquet/ORC
- Organised into partitions
- Organised by Event Time
- Combine to larger partitions over time
- Optimised for analytics

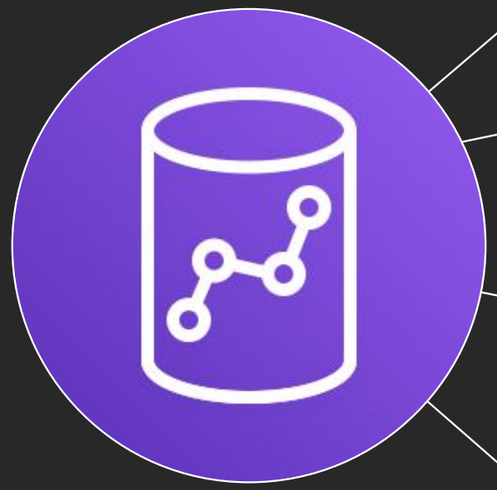
Tiered approach

Amazon S3

Tier 3 data lake: Analytics

- Domain-level data mart
- Organised by use cases
- Optimised for specialised analysis

Data warehouse



Amazon Redshift

- Fast speeds over structured schema
- Concurrency scaling for bursts of activity
- Data lake integration for security and control
- Query across Redshift and S3

Security and governance patterns

Data storage security

Implement access control in a multi-team environment

- Fine-grained
- Coarse-grained

Secure and segregated access to

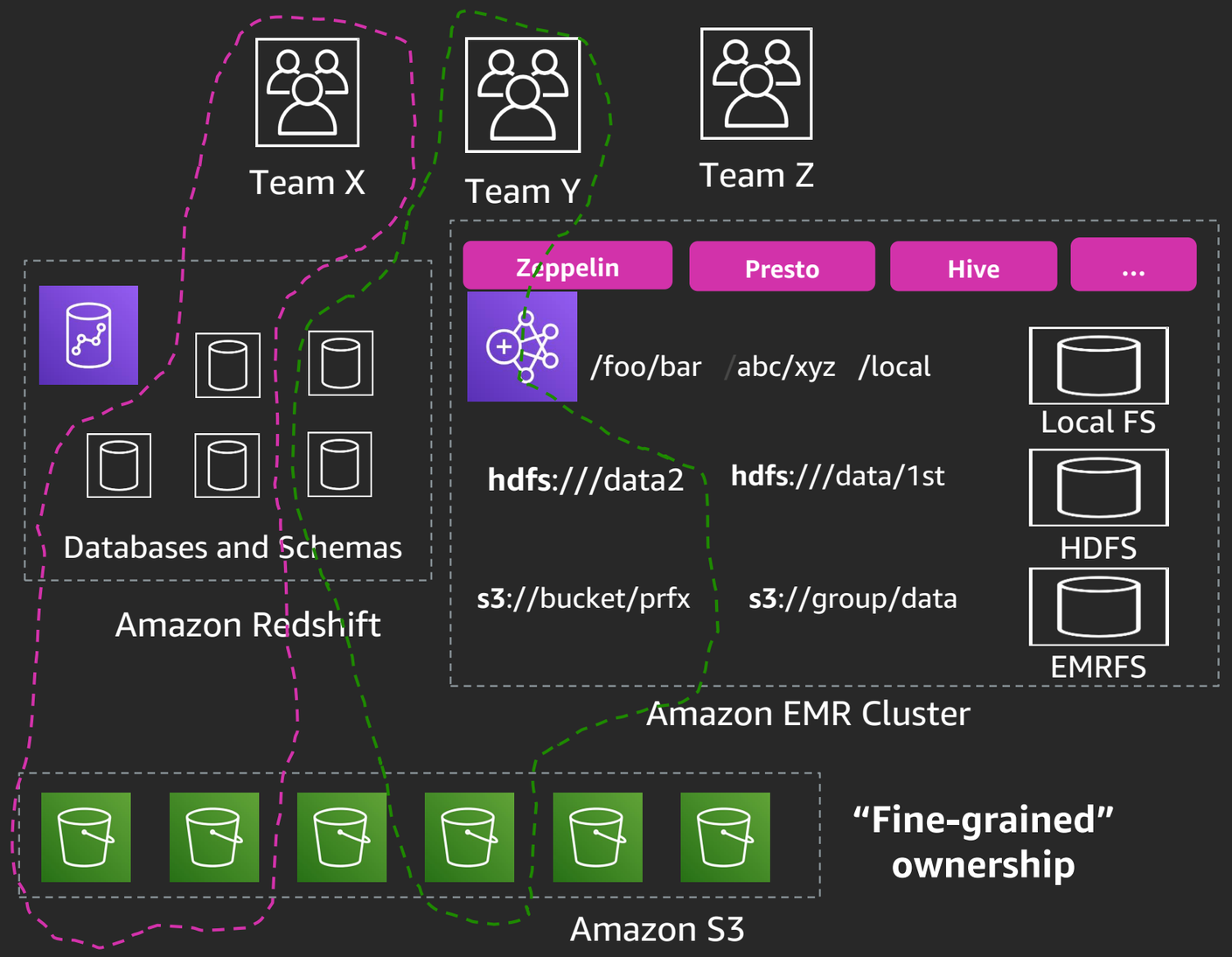
- Amazon S3
- Amazon EMR clusters
- Amazon Redshift clusters
- Serverless analytics services and other tools used in the pipeline

Encrypt data assets

Control access to data – fine grained ACL

"Fine-grained" data and resource ownership

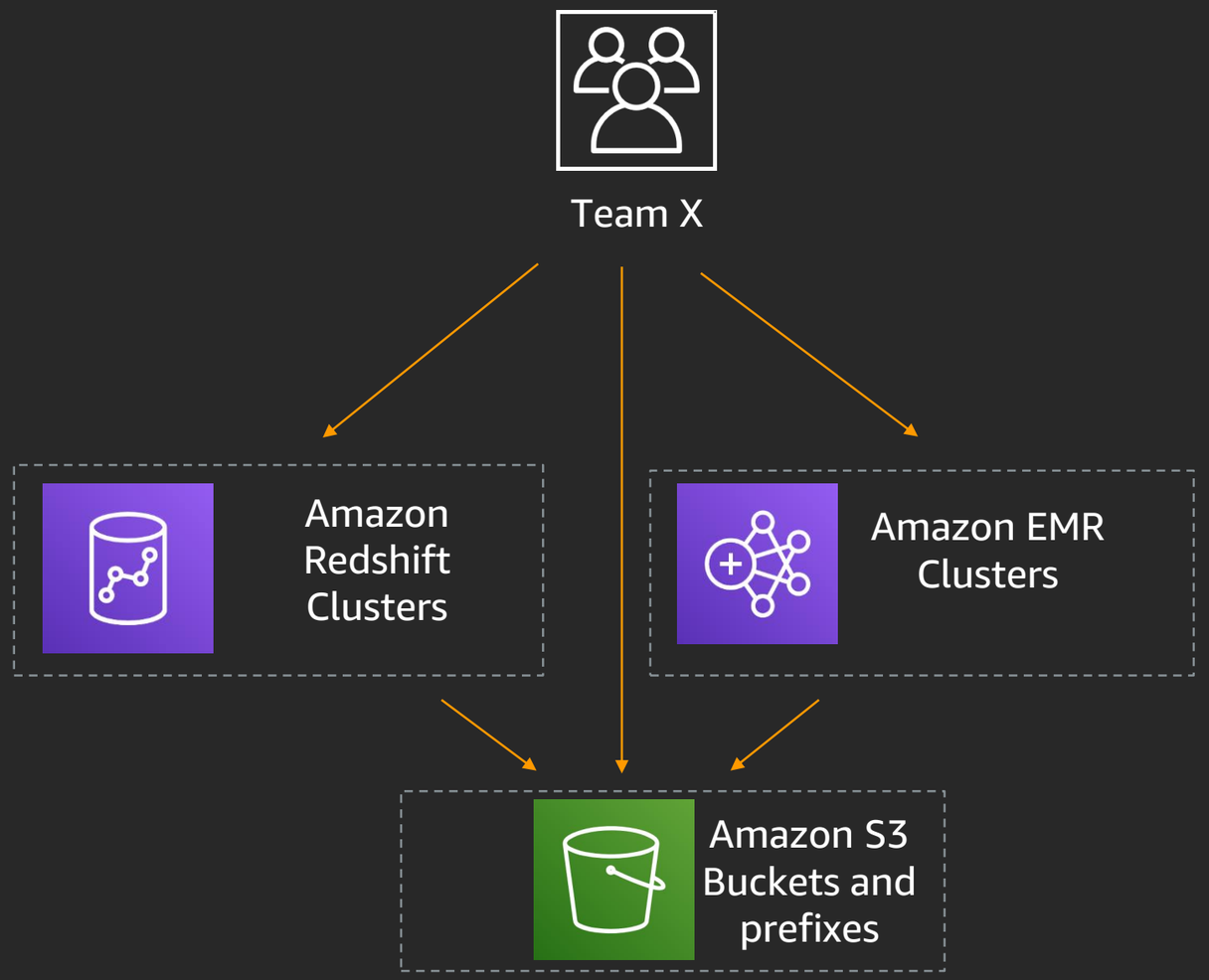
- Teams **share S3 buckets and clusters**
- Access control is complex to set up and maintain
- Common in a "**shared services**" architecture



Control data access - coarse grained

Prefer “coarse-grained” data and resource ownership

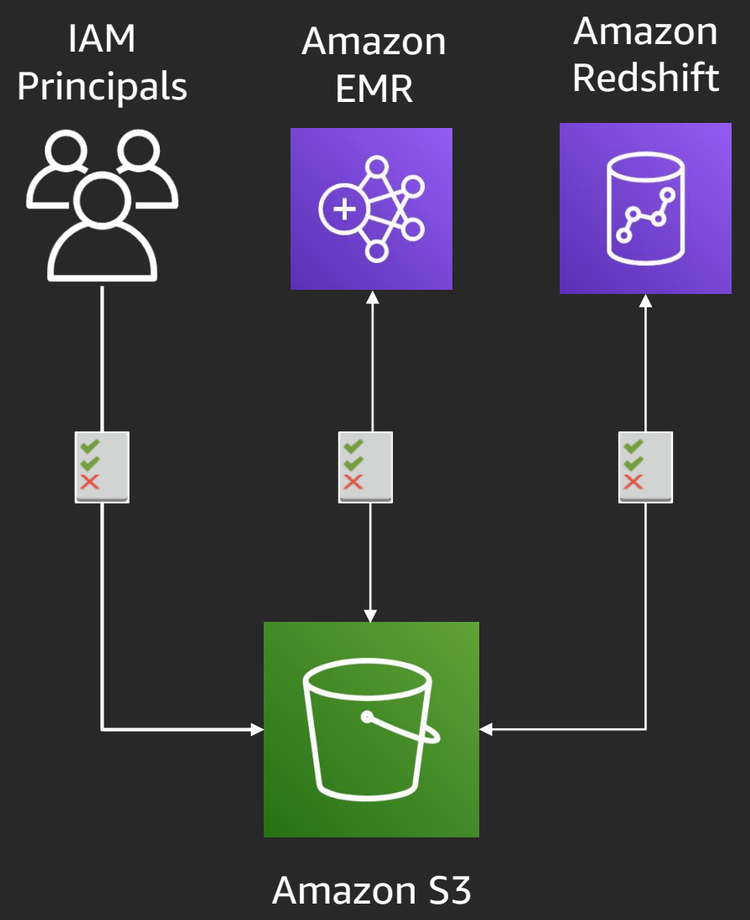
- Teams own **entire S3 buckets and clusters**
- Ownership segregated by AWS accounts
- Access control easier to setup and maintain
- Suitable for **autonomous teams**



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 EC2 **instance profile**
- Authorise access from other tools such as **Amazon Redshift** using IAM roles



Block public access to Amazon S3

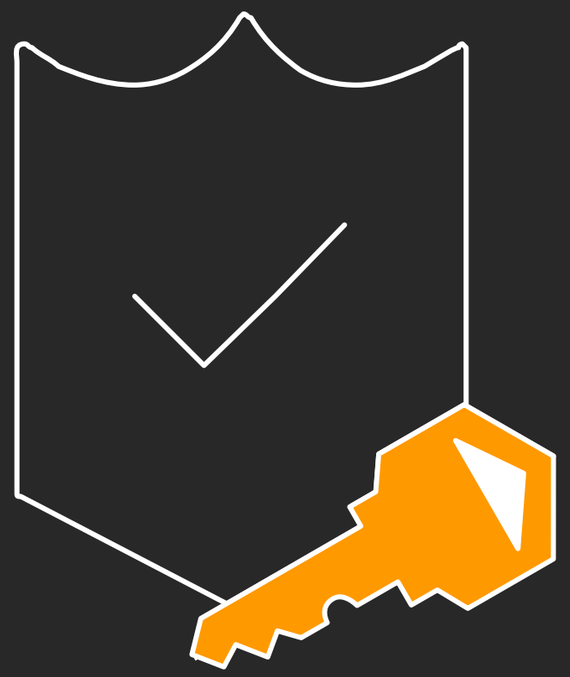
Amazon S3 provides four settings

- `BlockPublicAcls`
- `IgnorePublicAcls`
- `BlockPublicPolicy`
- `RestrictPublicBuckets`

But what is “public”?

- **Public object (or bucket) ACL** → grants permission to members of the predefined ***AllUsers*** or ***AuthenticatedUsers*** groups
- **Public bucket policy** → **wild cards in Principal** and **Condition** elements

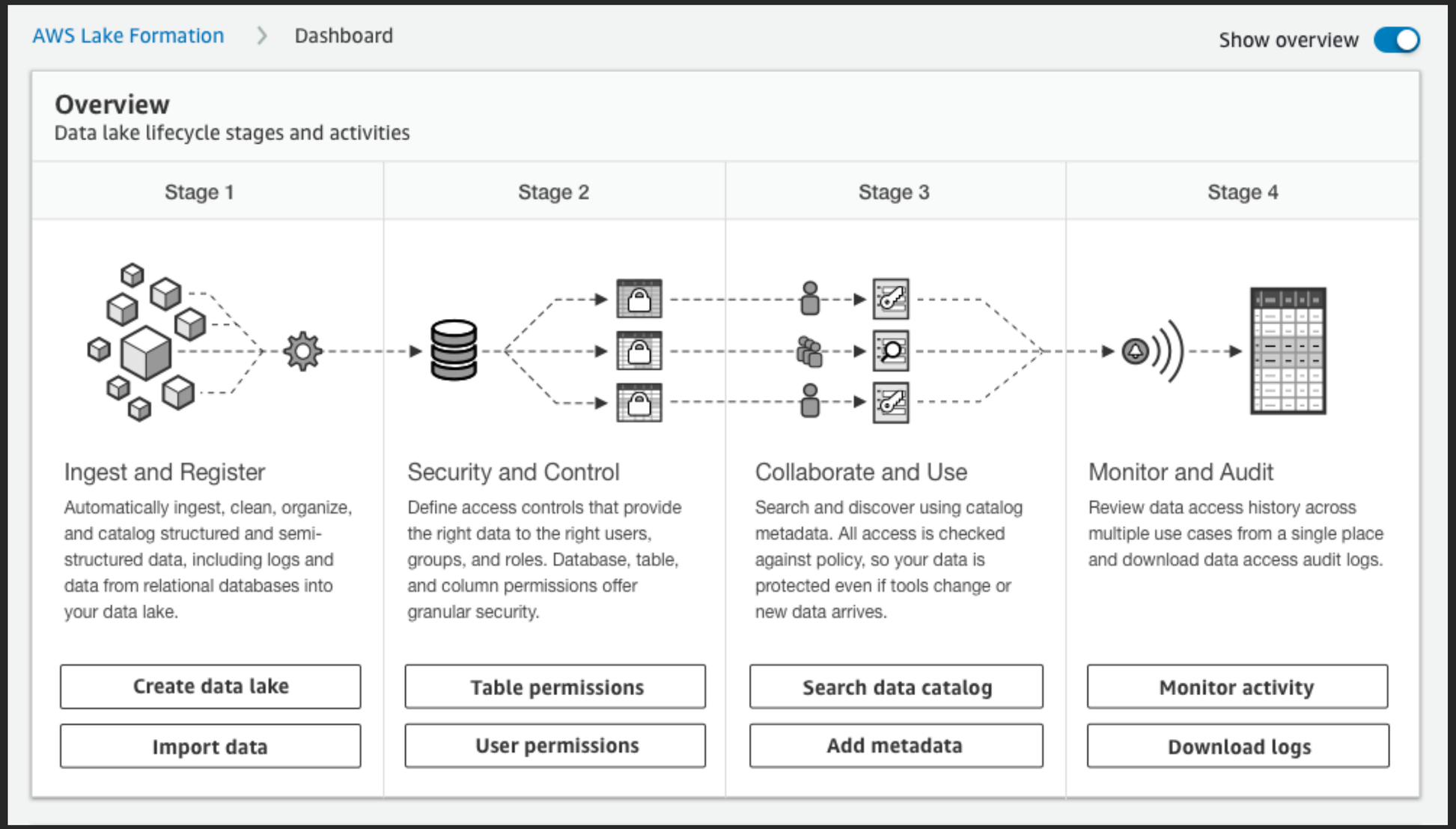
Encryption - data-at-rest and in-motion



- 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 KMS service

AWS Lake Formation

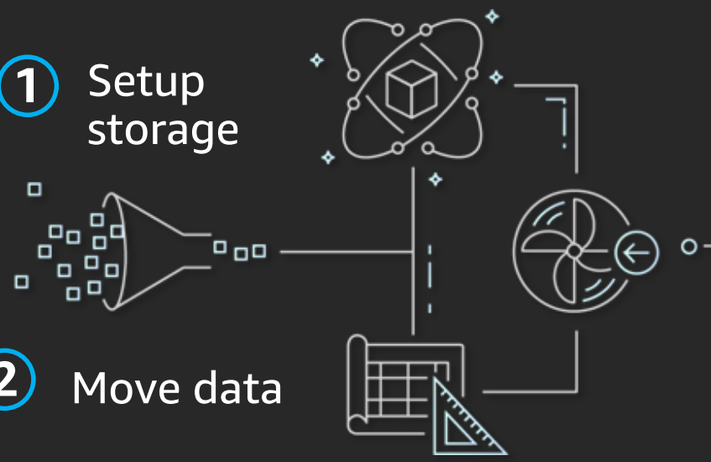
AWS Lake Formation



Typical steps of building a data lake

Ingestion and cleaning

1. Setup storage
2. Move data
3. Cleanse, prep, catalog data



Data engineer

Security

4. Configure and enforce security and compliance policies

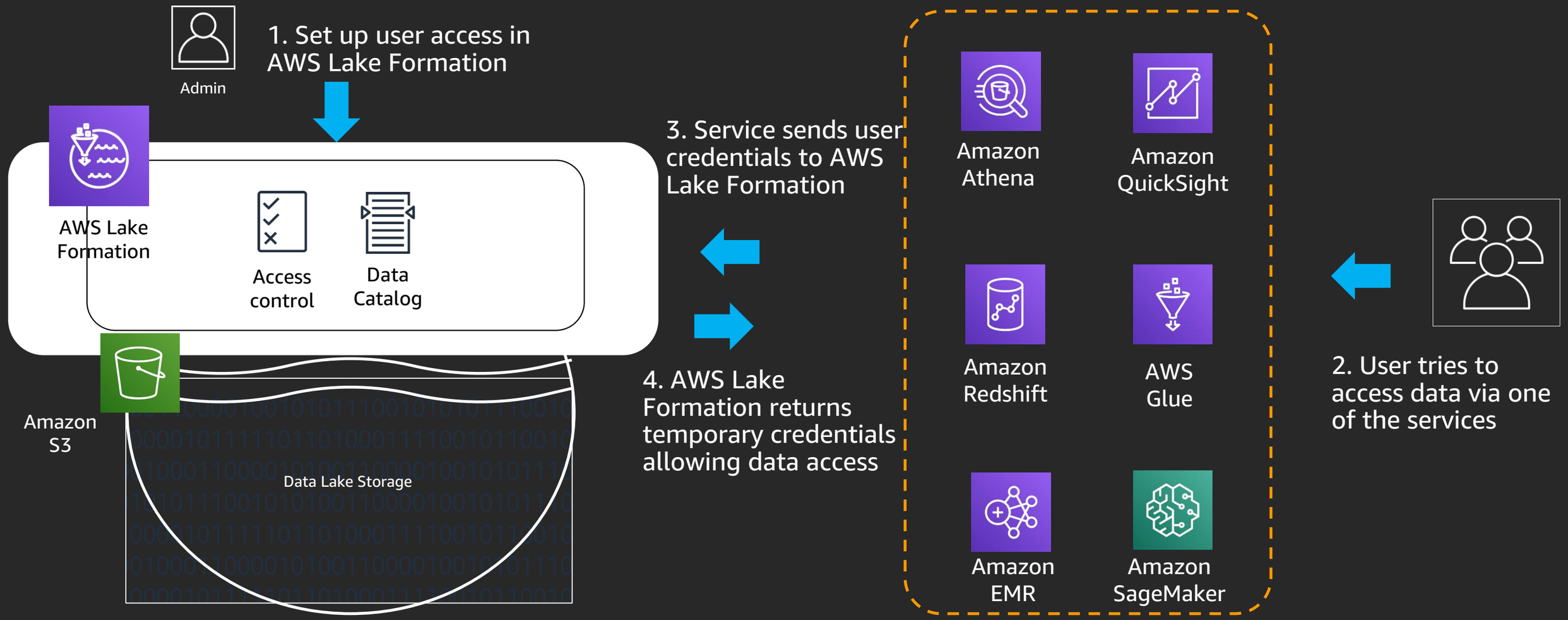
Data security officer

Analytics and ML

5. Make data available for analytics

Data analyst

Secure once - access in multiple ways



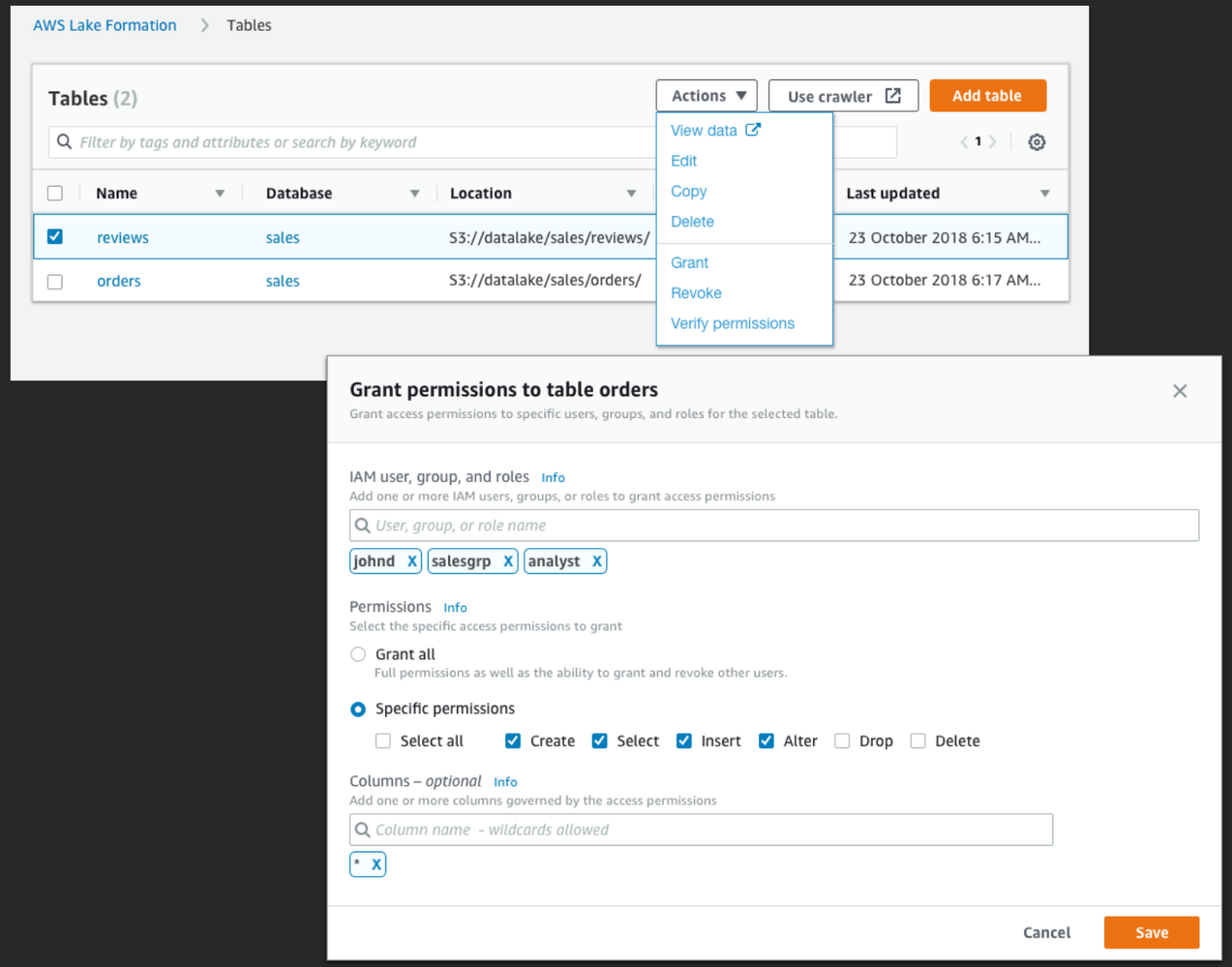
Security permissions in AWS Lake Formation

Control data access with simple grant and revoke permissions

Specify permissions on tables and columns rather than on buckets and objects

Easily view policies granted to a particular user

Audit all data access at one place



Security permissions in Lake Formation

Control data access with simple grant and revoke permissions

Specify permissions on **tables** and **columns** rather than on buckets and objects

Easily view permissions granted to a particular user

Audit all data access in one place

Column name	Data type
marketplace	string
customer_id	bigint
review_id	string
product_id	string
product_parent	bigint
product_title	string
star_rating	string
helpful_votes	bigint
total_votes	bigint
vine	string
verified_purchase	string
review_headline	string
review_body	string
review_date	string
product_category	string

User 1

User 2

Security deep dive

User

1 Query T

Athena

EMR

Redshift

Glue

2 Request access T

3 Short-term creds. for T

AWS Lake Formation

4 Request objs comprising T

5 Return objs of T

Amazon S3

Principals can be
IAM users, roles, and
users via federation
e.g., Active Directory

Granular permissions control for users

Grant permissions

Grant access permissions to specific users and roles.

IAM users and roles

Add one or more IAM users or roles.

Choose IAM principals to add

datalake_user_redshift
User

Database

Add one or more databases.

Choose databases

amazoncloudtrail

Table - *optional*

Add one or more tables.

Choose tables

amazoncloudtrail_cloudtrail

Column - *optional*

Grant permissions to:

The include columns

Include columns

Add one or more columns to include.

Choose columns

useridentity string	eventsource string	eventname string
sourceipaddress string		

Table permissions

Choose the specific access permissions to grant.

- [x] Select all
- [x] Alter
- [x] Insert
- [x] Drop
- [x] Delete
- [x] Select
- [ ] Grant all

Enabling this permission grants full access to the specified resources while still logging access requests based on the individual permission settings above. It is typically used for debugging. Specific individual permissions set above will take effect when Grant all is later removed.

Grantable permissions

Choose the specific permissions that may be granted to others.

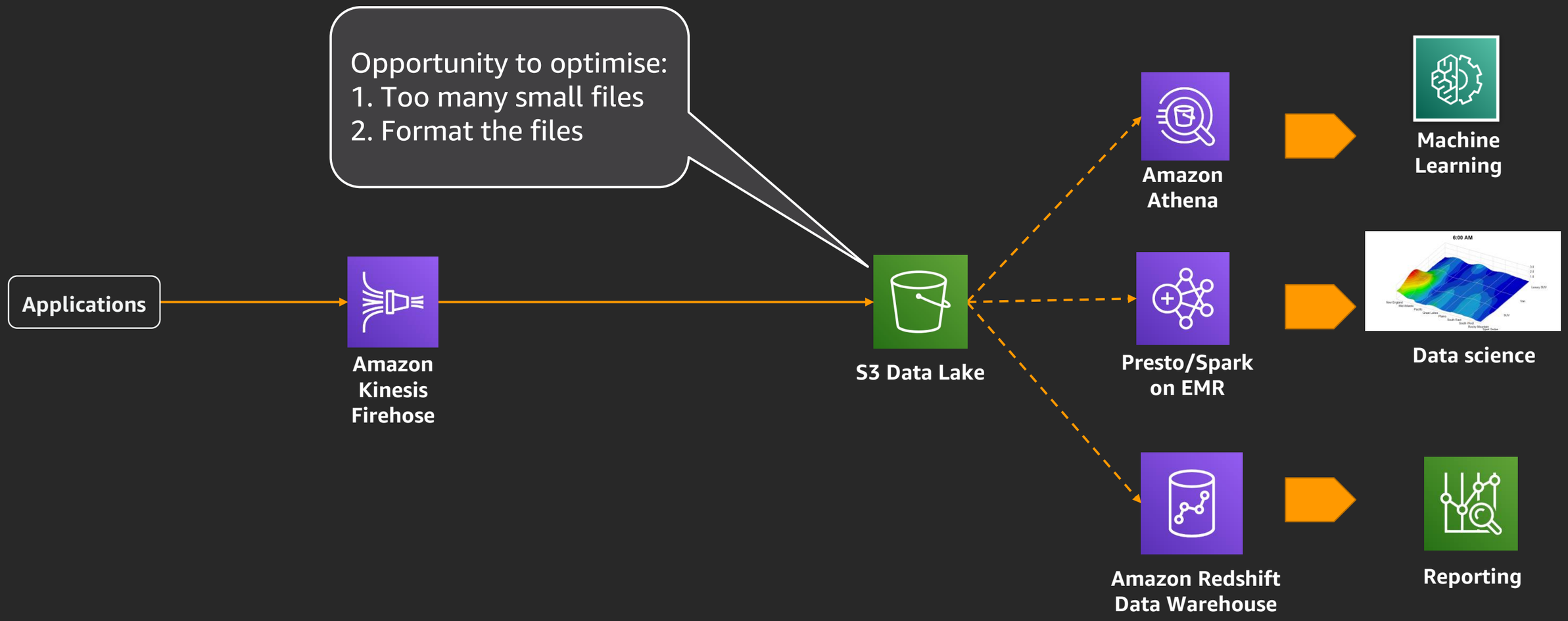
- [ ] Select all
- [ ] Create table
- [ ] Alter
- [ ] Drop
- [ ] Grant all

Enabling this permission grants full access to the specified resources while still logging access requests based on the individual permission settings above. It is typically used for debugging. Specific individual permissions set above will take effect when Grant all is later removed.

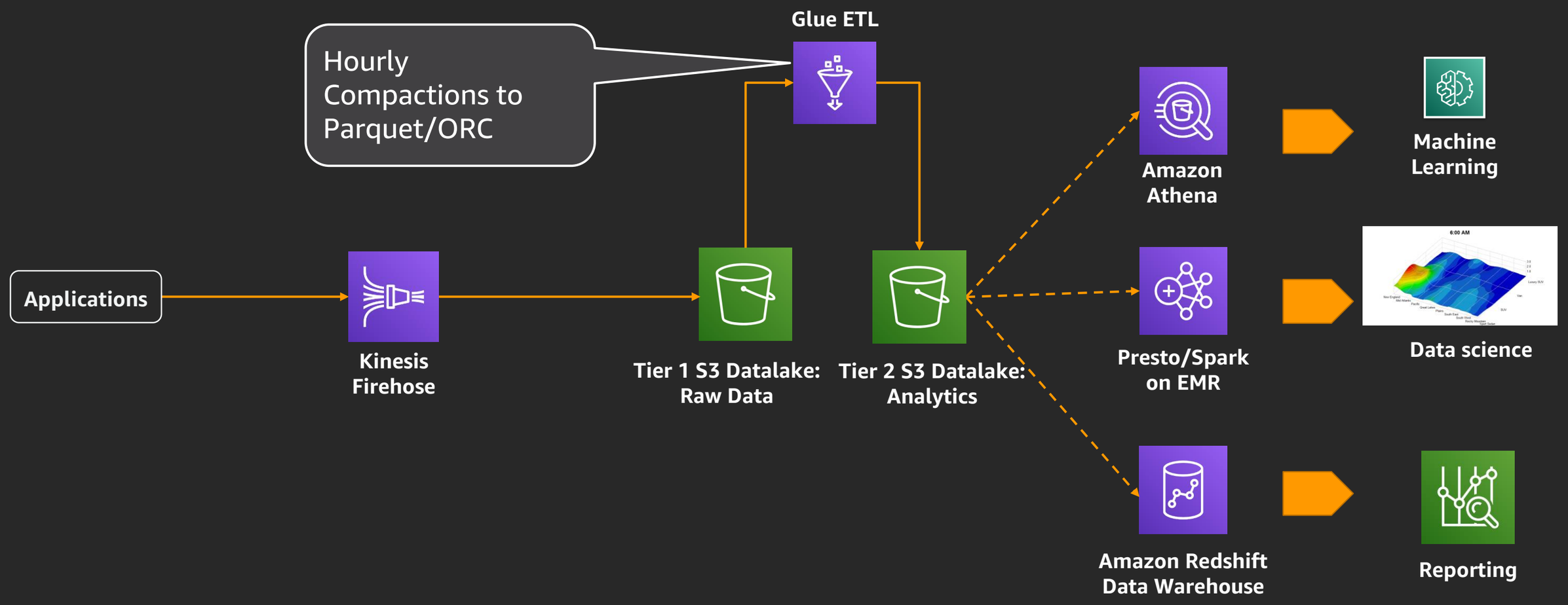
Cancel | Grant

Data lake design patterns

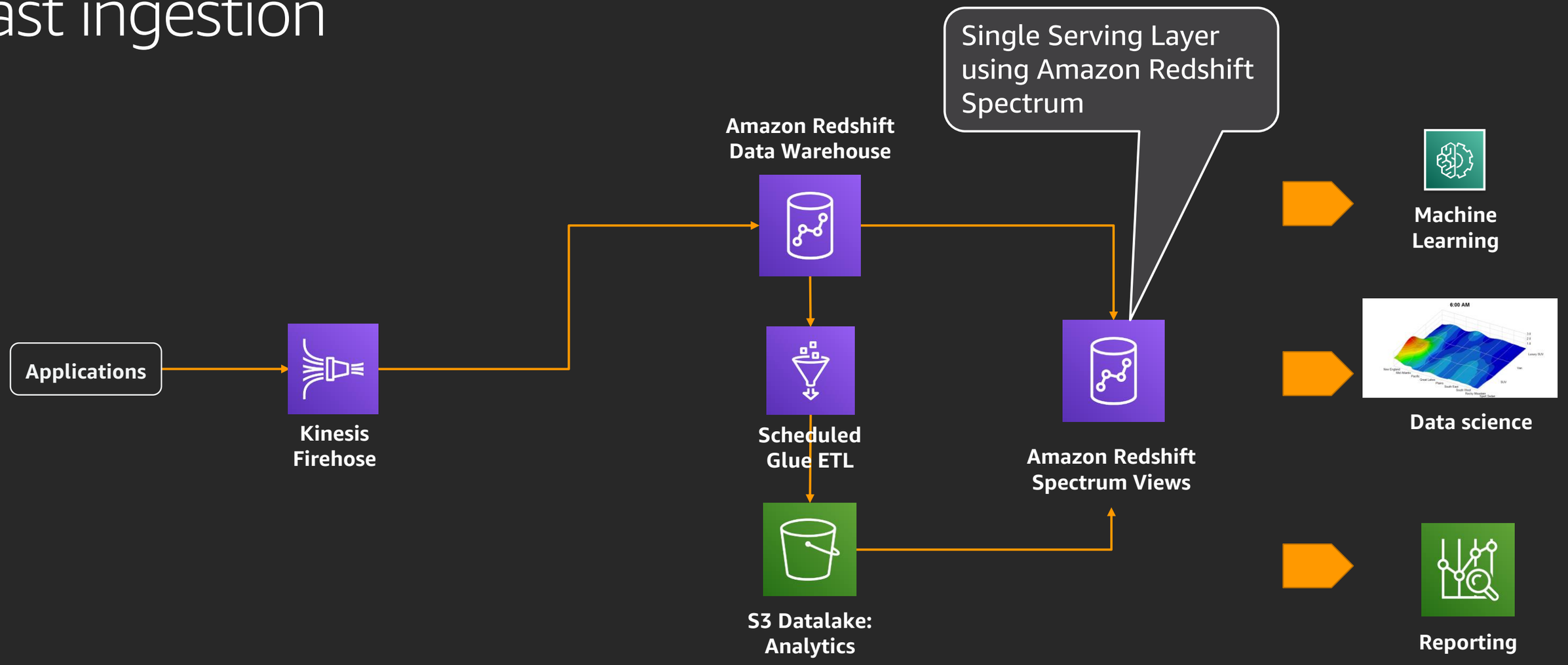
Log analytics, IoT sensor data



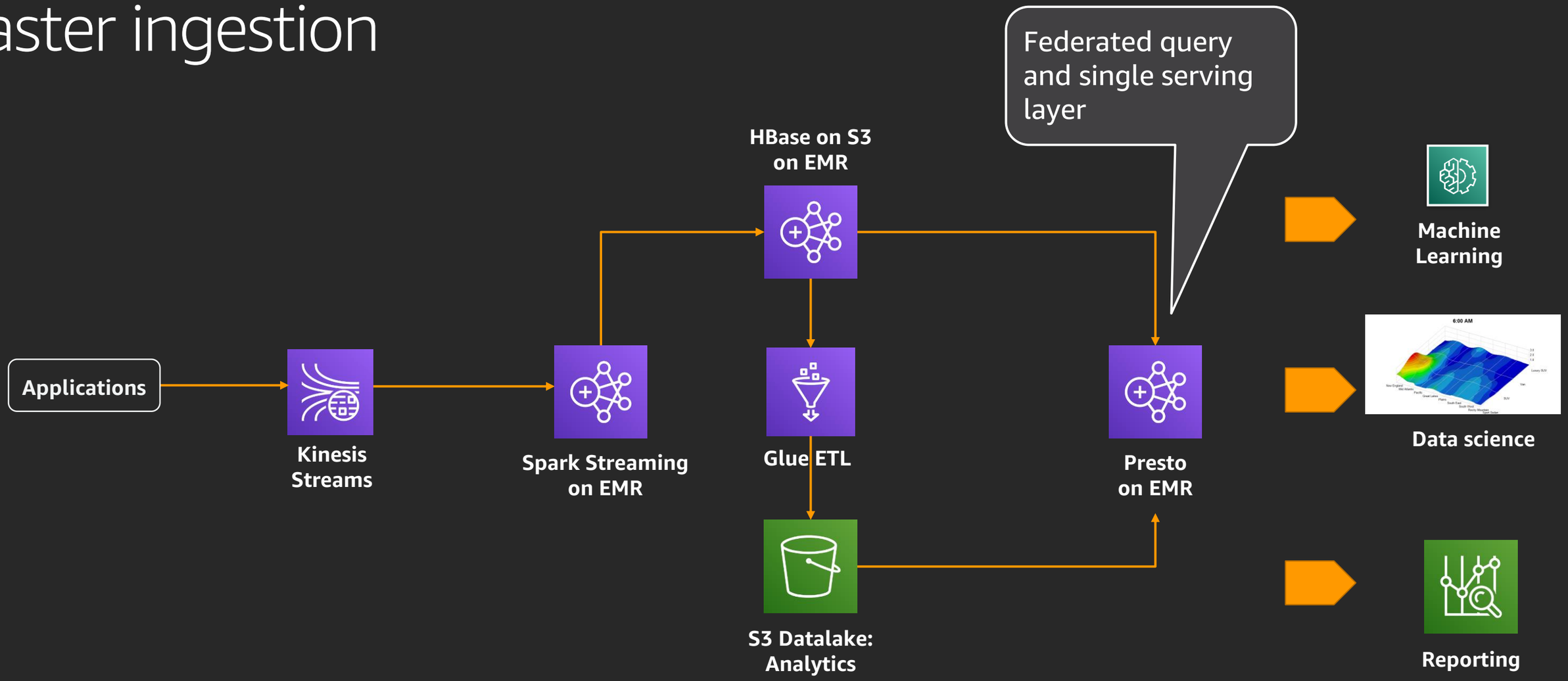
Log analytics, IoT sensor data



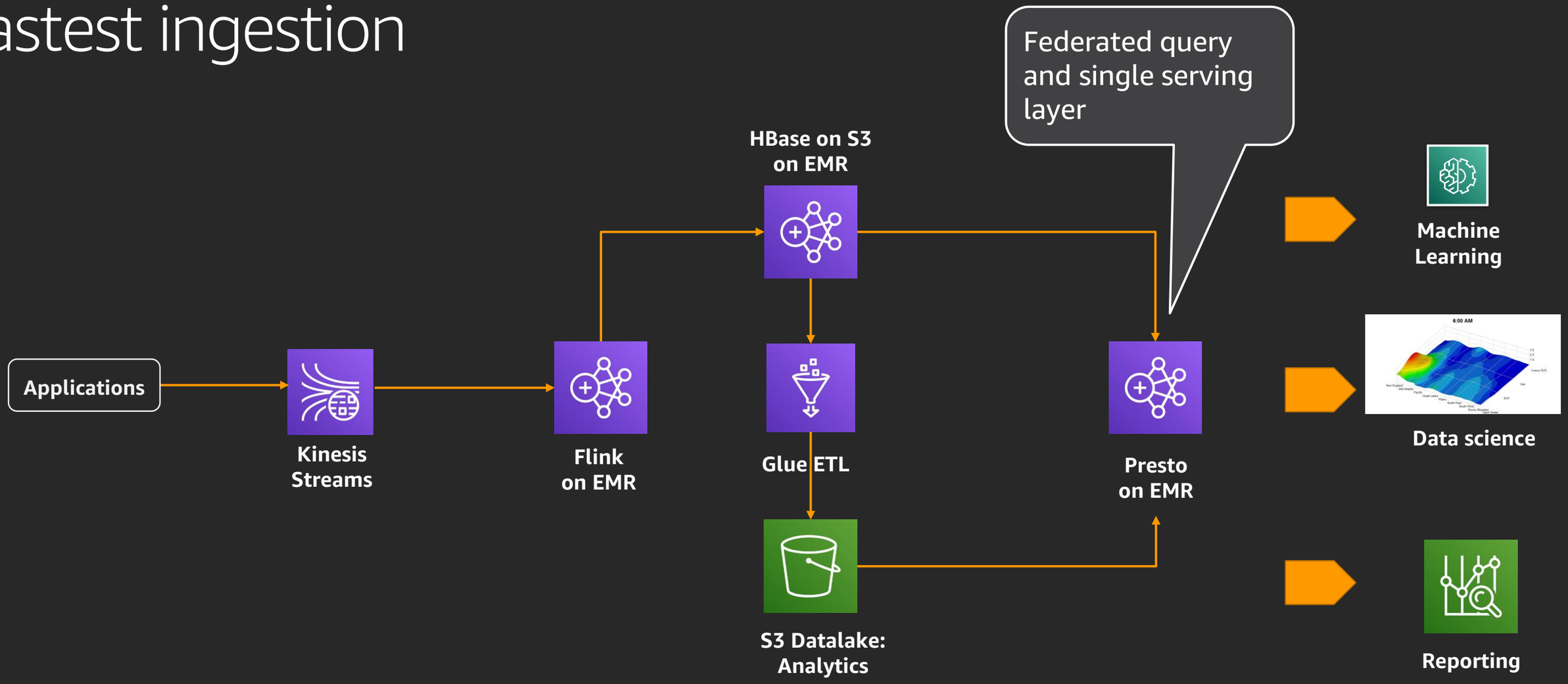
Fast ingestion



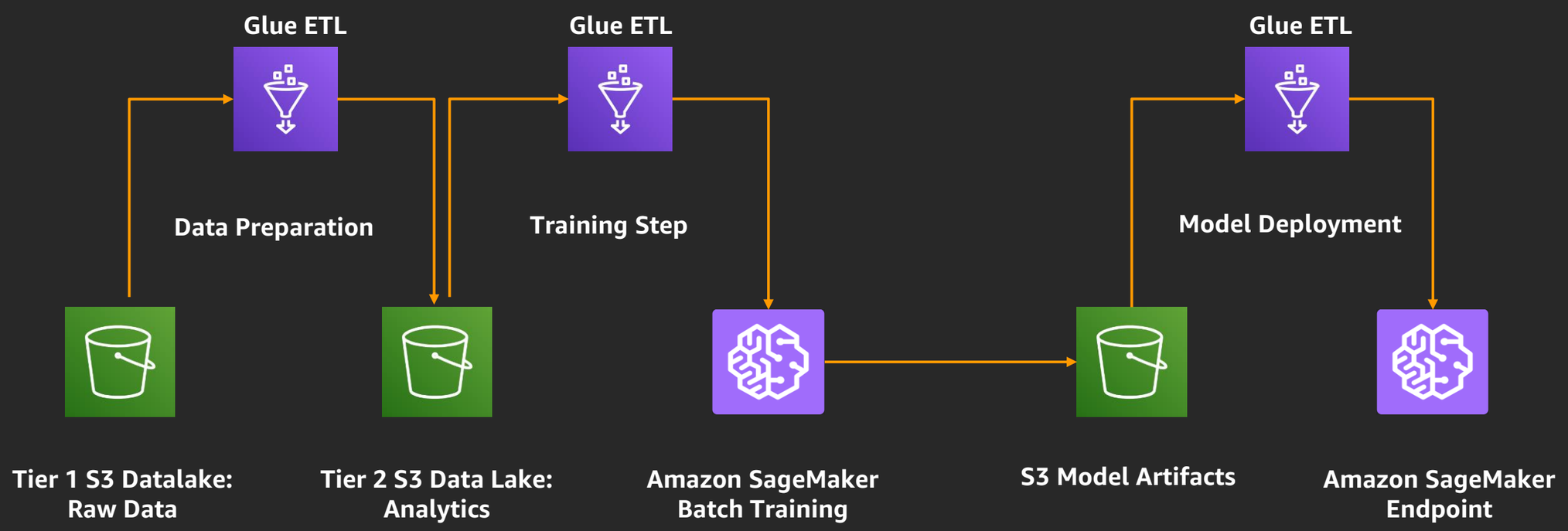
Faster ingestion



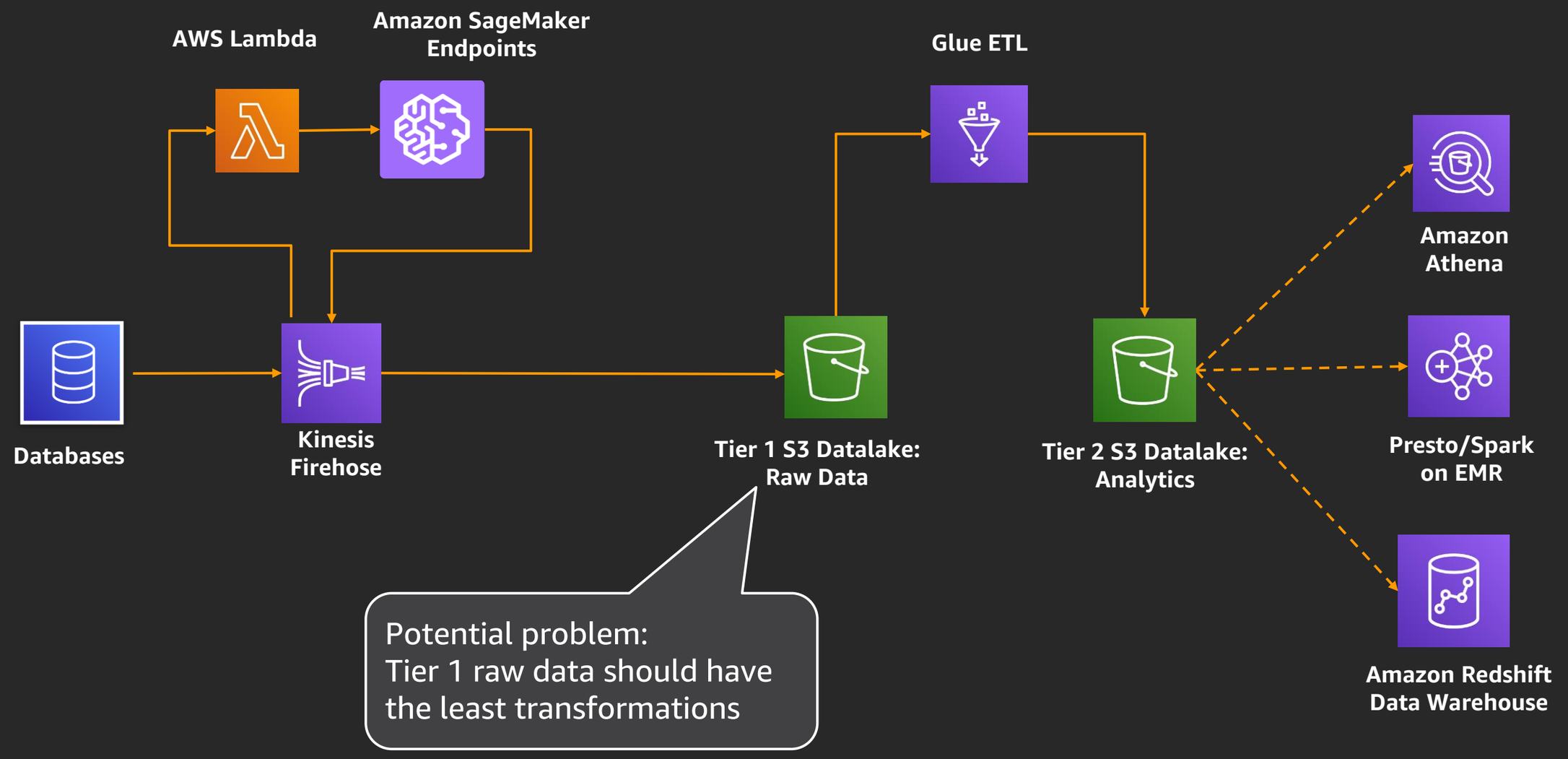
Fastest ingestion



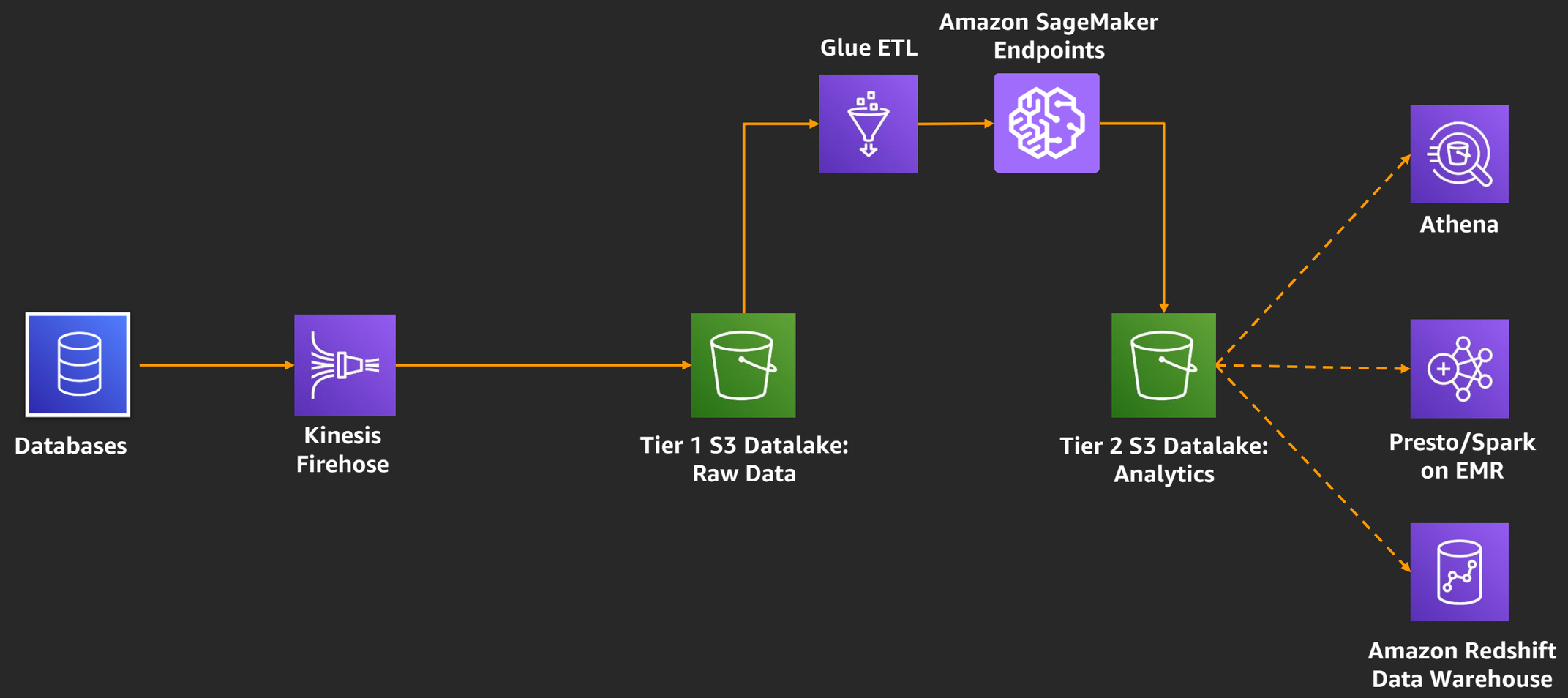
Machine Learning - batch training pipeline



Machine Learning - predictions on streaming data

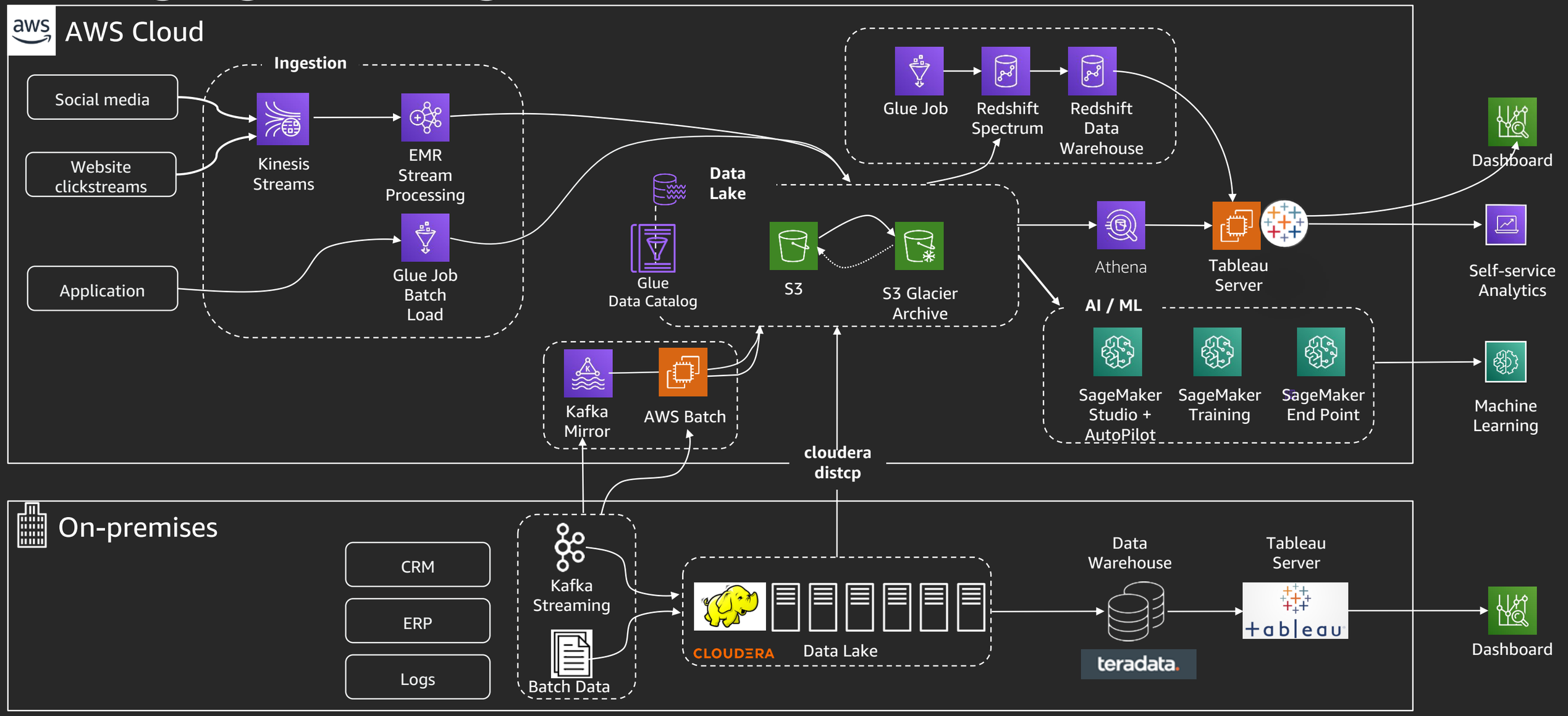


Machine Learning - predictions on streaming data



Phased architecture build

Bringing it all together …



Conclusion

Building modern data architecture is a phased journey

Data lake should be designed as a self-service platform

Leverage tools to automate (AWS Lake Formation)

Leverage design patterns to deliver key use cases

Use security and governance controls

Thank you!

aws SUMMIT ONLINE