AWS re:Invent

STG359-R

Best practices for implementing a Data Lake on Amazon S3

Amy Che
Principal Technical Program Manager, Amazon S3
Amazon Web Services

John Mallory
Storage Business Development Manager
Amazon Web Services

Gerard Bartolome
Data Platform Architect
Sweetgreen

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Data at scale

Growing exponentially

From new sources

Increasingly diverse

Used by many people

Analyzed by many applications

Agenda

Data at scale and Data Lakes

Sweetgreen's Data Lake best practices

Data Lake foundation best practices

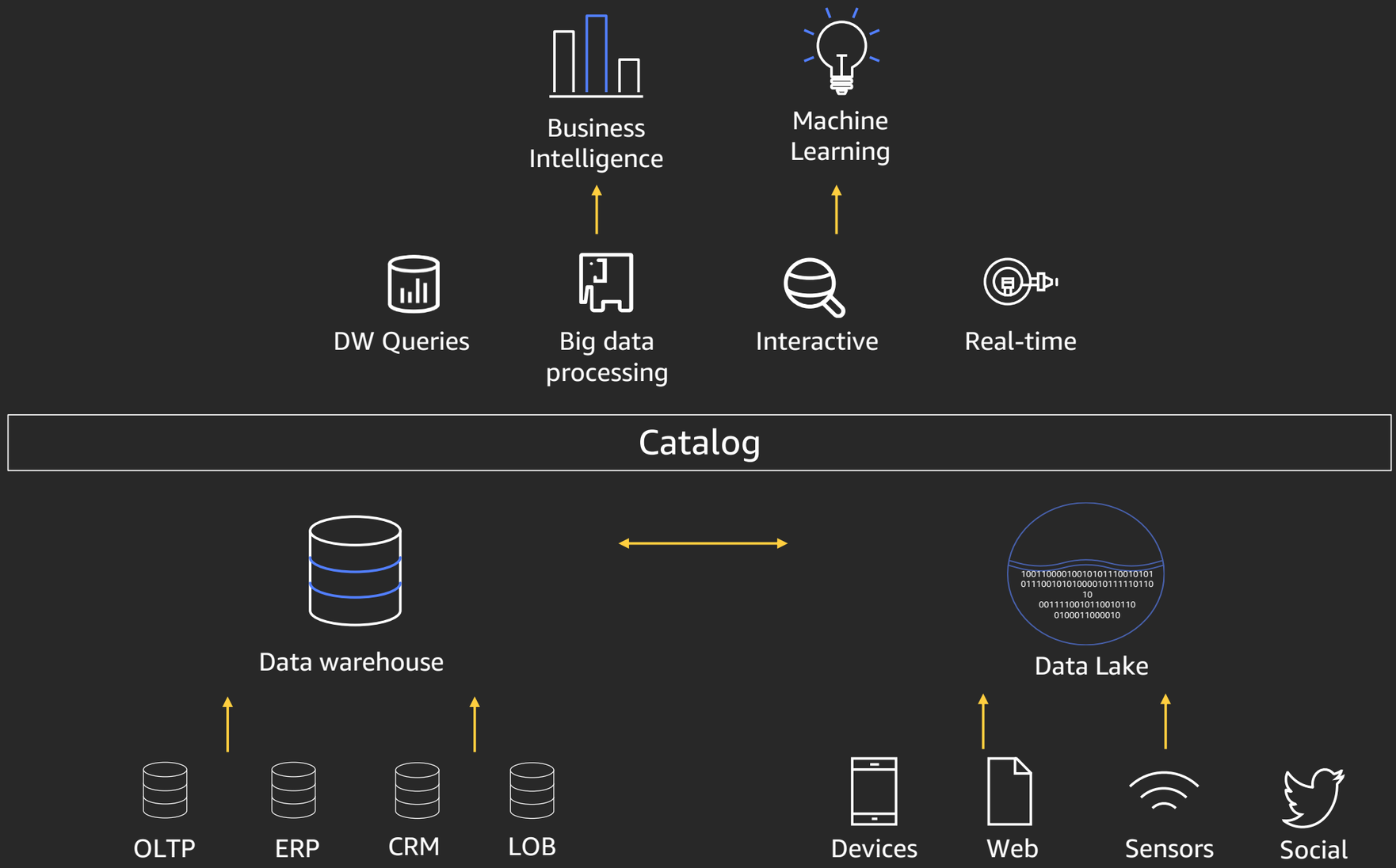
[Data Lake] performance best practices

[Data Lake] security best practices

Data at Scale

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Defining the Data Lake



Defining the Data Lake

Amazon S3 as the foundation for Data Lakes

Durable, available, exabyte scalable

Secure, compliant, auditable

High performance

Low cost storage and analytics

Broad ecosystem integration

Best practices for implementing a Data Lake on Amazon S3

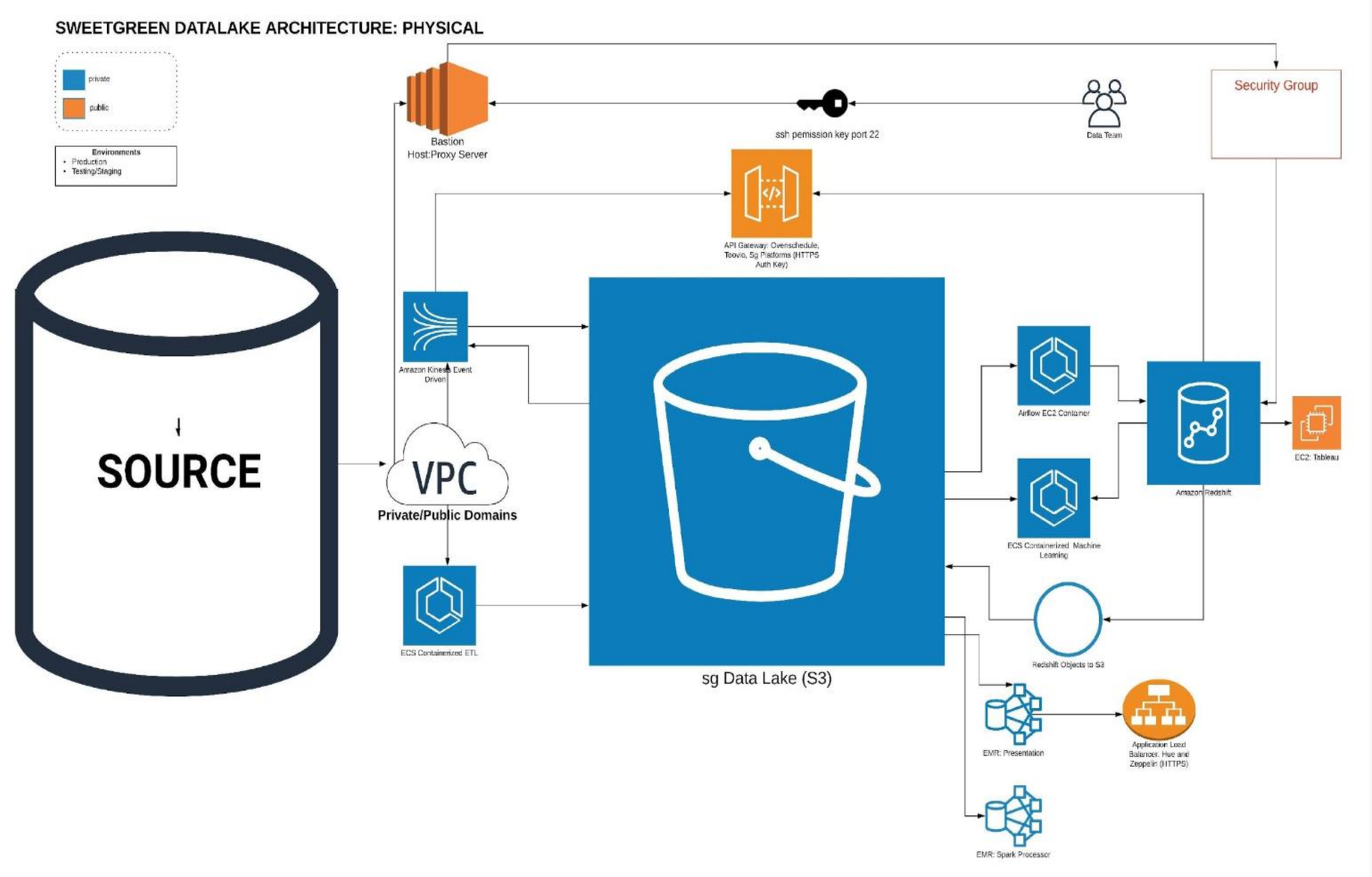
Gerard Bartolome

OUR FOOD ETHOS

We believe the choices we make about what we eat, where it comes from and how it's prepared have a direct and powerful impact on the health of individuals, communities and the environment.

scratch cooking | transparency | sustainability | local sourcing | animal welfare

ECOSYSTEM


SWEETGREEN DATALAKE ARCHITECTURE: PHYSICAL
private
public
Environments
• Production
• Testing/Staging
Bastion
Host:Proxy Server
ssh pemission key port 22
Data Team
Security Group
API Gateway: Ovenschedule, Toovio, Sg Platforms (HTTPS Auth Key)
SOURCE
Amazon Kinesis Event Driven
VPC
Private/Public Domains
ECS Containerized ETL
sg Data Lake (S3)
Airflow EC2 Container
ECS Containerized Machine Learning
Amazon Redshift
EC2: Tableau
Redshift Objects to S3
EMR: Presentation
Application Load Balancer: Hue and Zeppelin (HTTPS)
EMR: Spark Processor

"Adapt the language to the data;
DON'T adapt the data to the language”

SOURCE TYPES

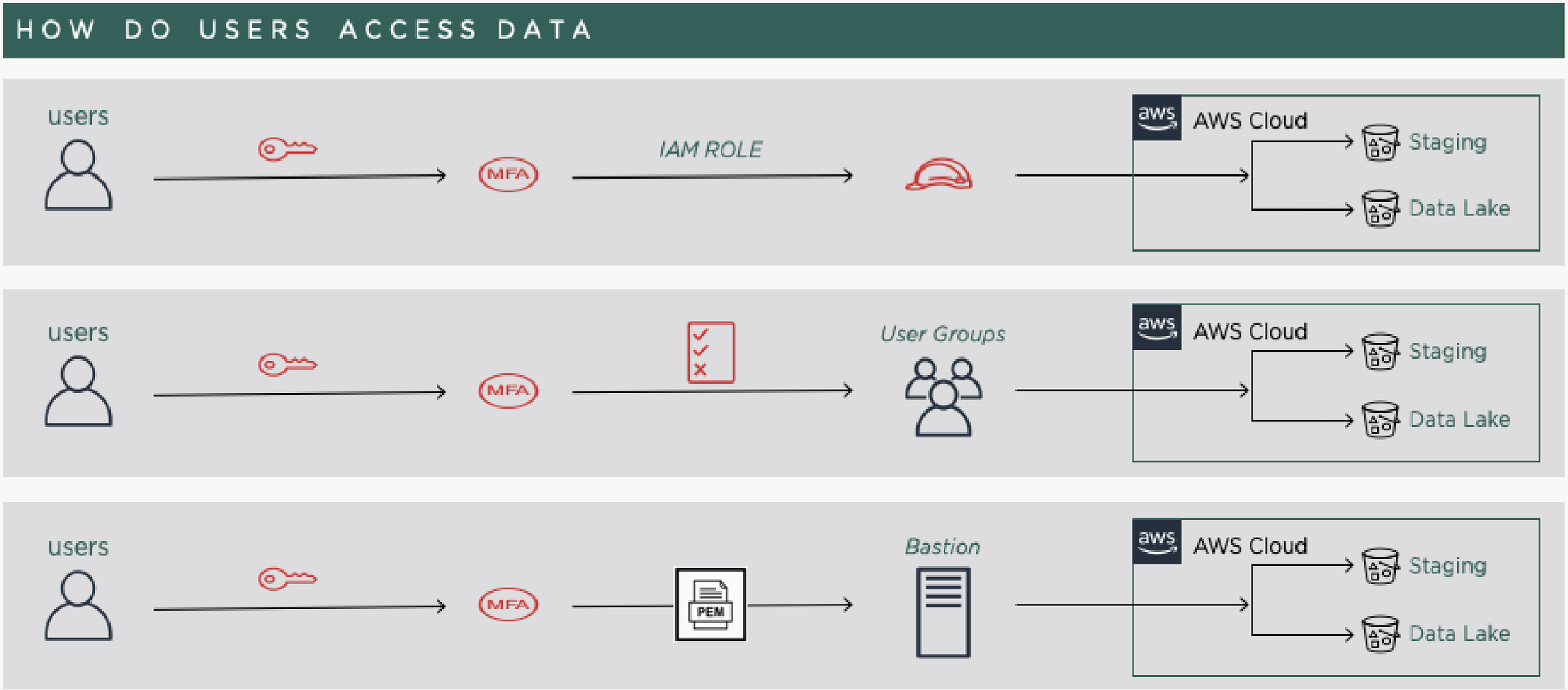
STRUCTURED

static format with structure (databases, file uploads, vendor objects pushed)

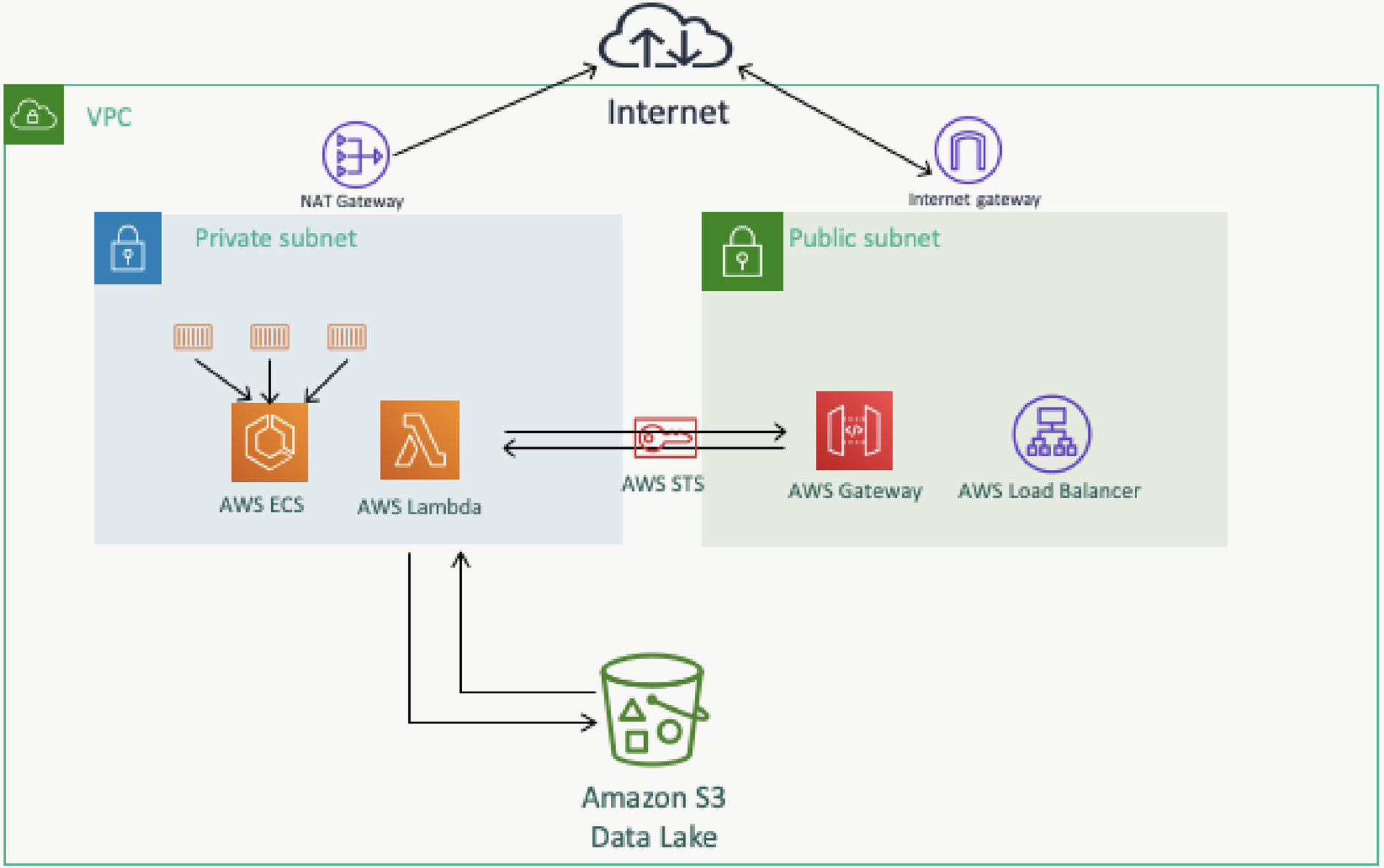
SEMI/UNSTRUCTURED

dynamic and no predefined structure (Kinesis/ Kafka streams, JSON)

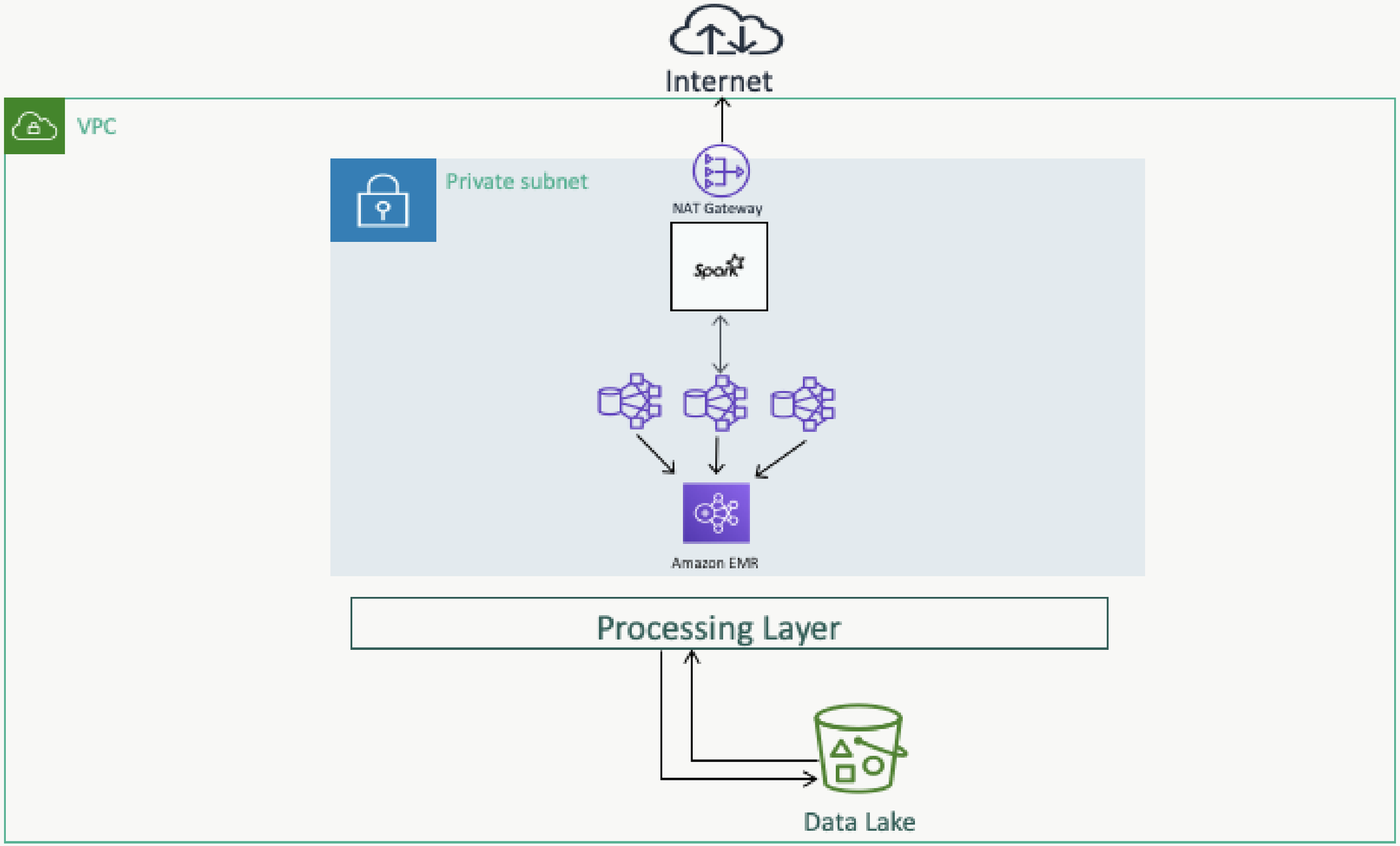
TRANSFORM: S3 SECURITY



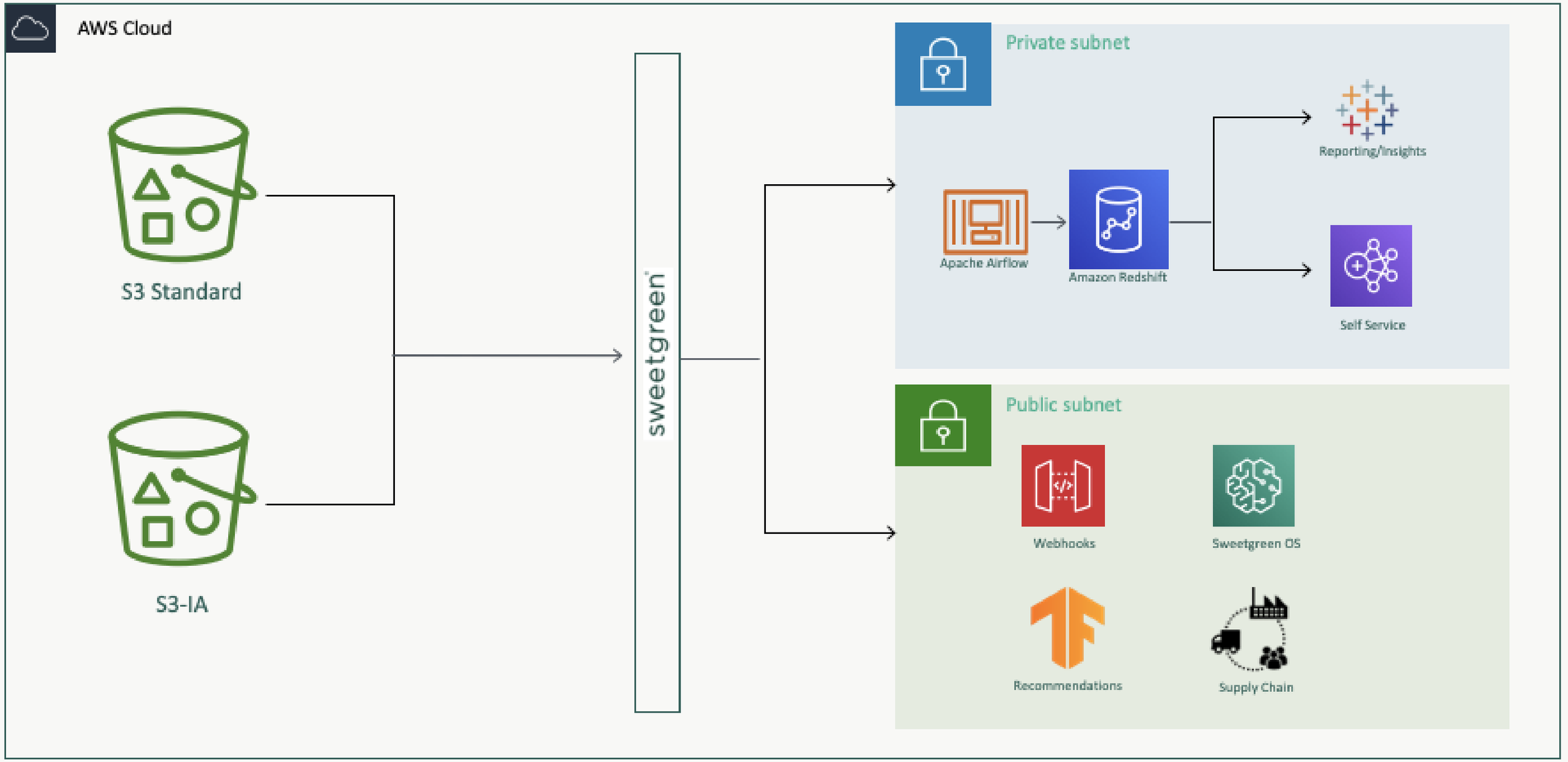
TRANSFORM: ECS / SERVERLESS



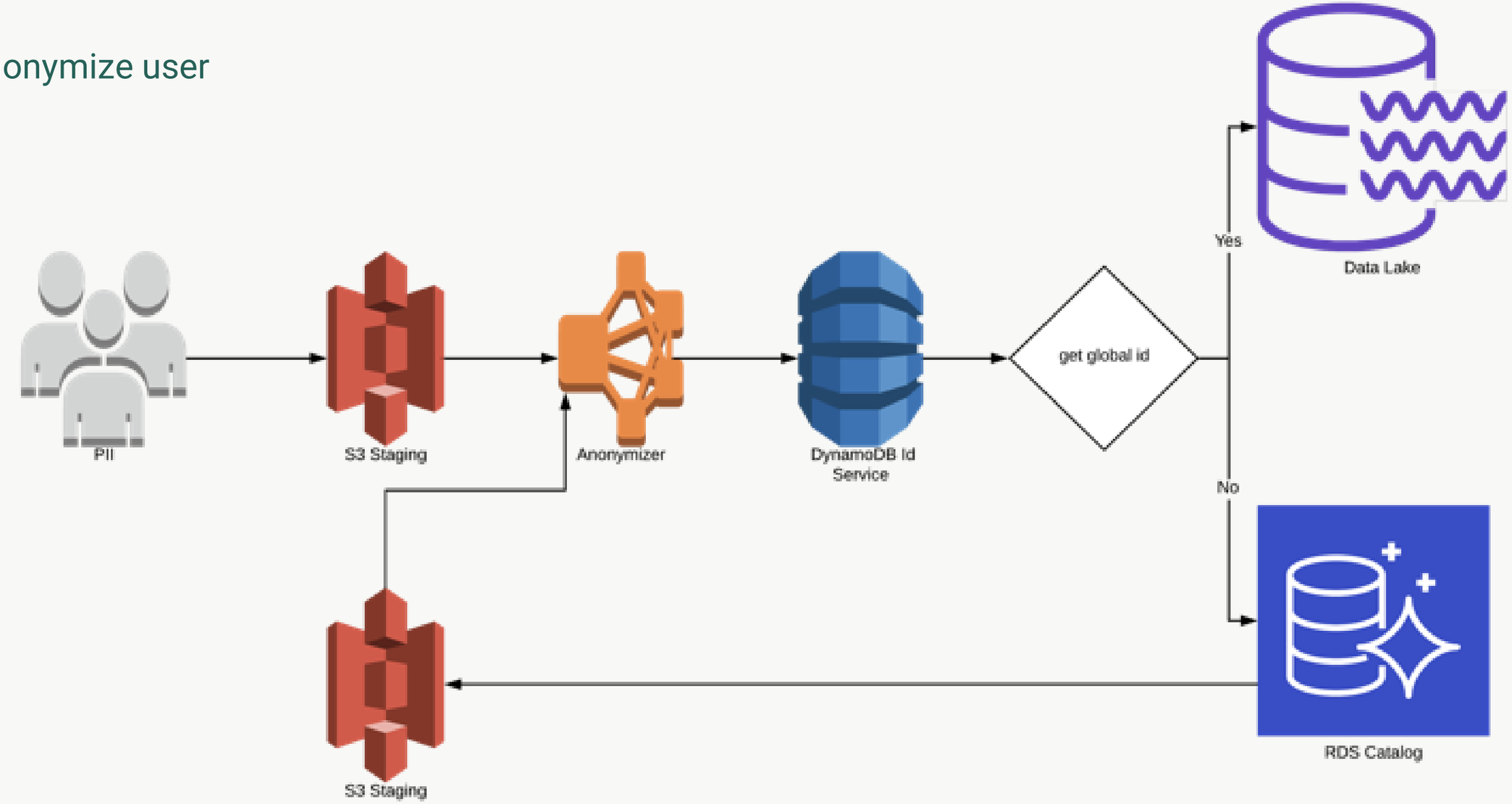
TRANSFORM: EMR



USAGE


AWS Cloud
S3 Standard
S3-IA
sweetgreen
Private subnet
Apache Airflow
Amazon Redshift
Reporting/Insights
Self Service
Public subnet
Webhooks
Sweetgreen OS
Recommendations
Supply Chain

Anonymize user



SCAN ME

We are hiring!!

https://grnh.se/b4116be81

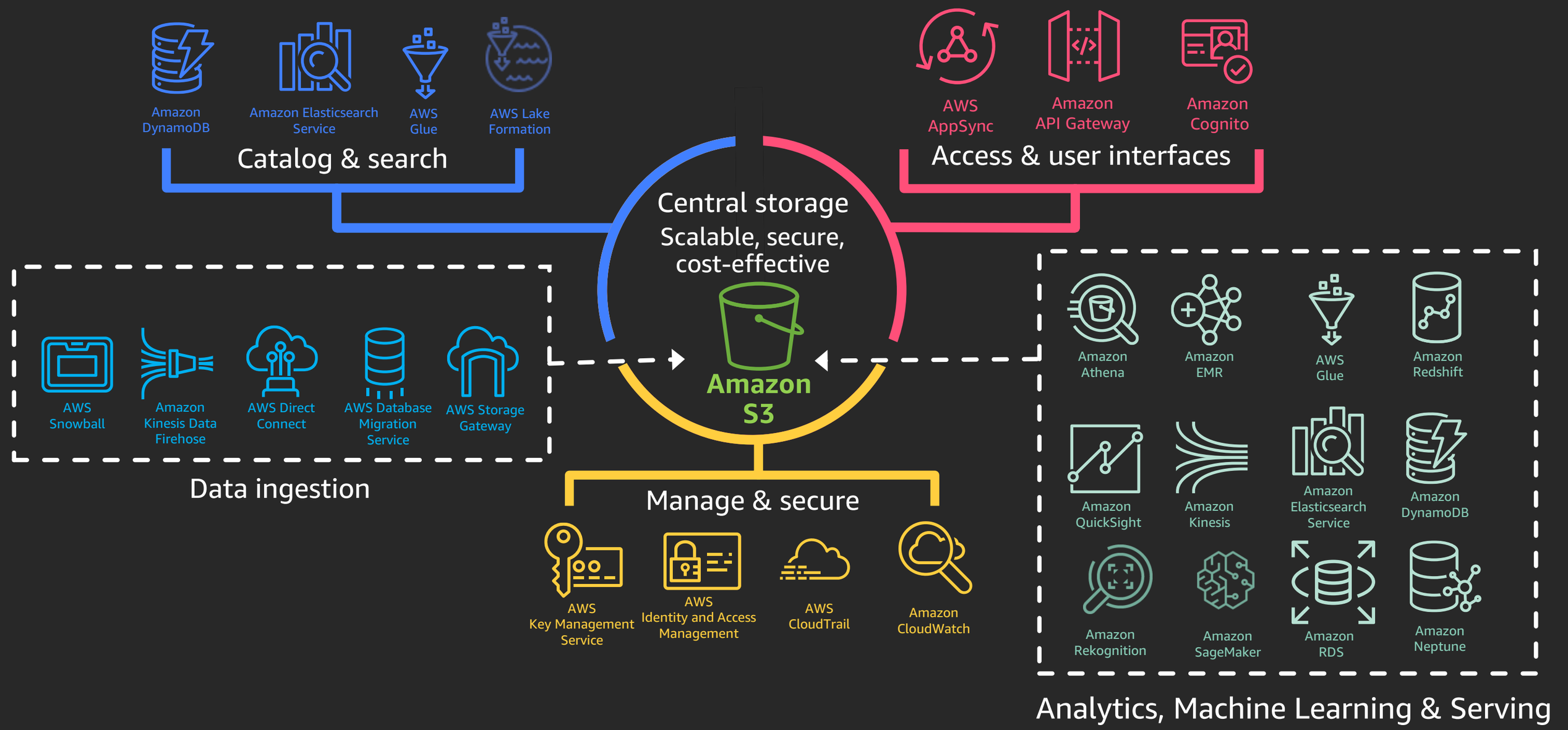
Data Lake foundation best practices

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Data Lake on AWS



Data Lake ingest and transform patterns

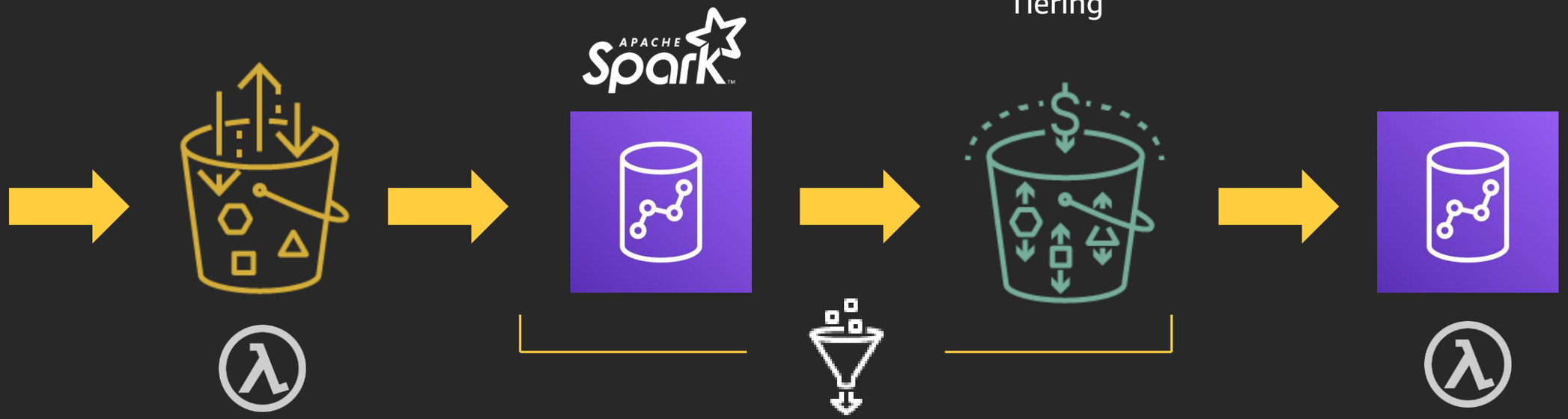
Pipelined architectures improve governance, data management and efficiency

Raw data
Amazon S3-Standard

ETL
Amazon Glue or EMR

Production data (Data Lake)
Amazon S3-Intelligent Tiering

Data warehouse
Amazon Redshift



Triggered code
Amazon Lambda

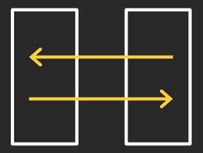
ETL & catalog management
AWS Glue and AWS Lake Formation

Triggered code
Amazon Lambda

Data management at scale best practices

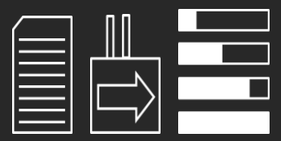
Utilize S3 object tagging

Granularly control access, analyze usage,
manage lifecycle policies, and replicate objects



Implement lifecycle policies

Automated policy-driven archive and data expiration



Utilize batch operations

Manage millions to billions of objects with a single request

Plan for rapid growth and automating management at any scale

Choosing the right Data Lake storage class

Select storage class by data pipeline stage

Raw data

Amazon
S3-Standard

Small log files
Overwrites if synced
Short lived
Moved & deleted
Batched & archived

ETL

Amazon
S3-Standard

Data churn
Small intermediates
Multiple transforms
Deletes < 30 days
Output to Data Lake

Production Data Lake

Amazon
S3-Intelligent-Tiering

Optimized sizes (MBs)
Many users
Unpredictable access
Long lived assets
Hot to cool

Online cool data

Amazon
S3-Standard
Infrequent Access (S3-IA/ZIA)

Replicated DR data
Infrequently accessed
Infrequent queries
ML model training

Historical data

Amazon
S3-Glacier or Deep-Archive

Historical assets
ML model training
Compliance/Audit
Data protection
Planned restores

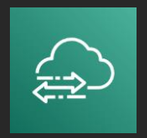
Optimize costs for all stages of Data Lake workflows

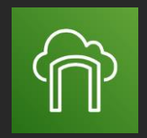
Efficiently ingest data from all sources

Ingest methods

AWS Direct Connect

Amazon Kinesis


AWS DataSync


AWS Storage Gateway

AWS Database Migration Service

AWS Snowball Edge

IoT, Sensor Data, Clickstream Data, Social Media Feeds, Streaming Logs

On-premise Data Lakes, EDW, Large Scale Data Collection

On-premise ERP, Mainframes, Lab Equipment, NAS Storage

Oracle, MySQL, MongoDB, DB2, SQL Server, Amazon RDS

Offline Sensor Data, NAS, On-premise Hadoop

Real time

Predictive analytics, IoT, sentiment analysis, recommendation engines

Batch

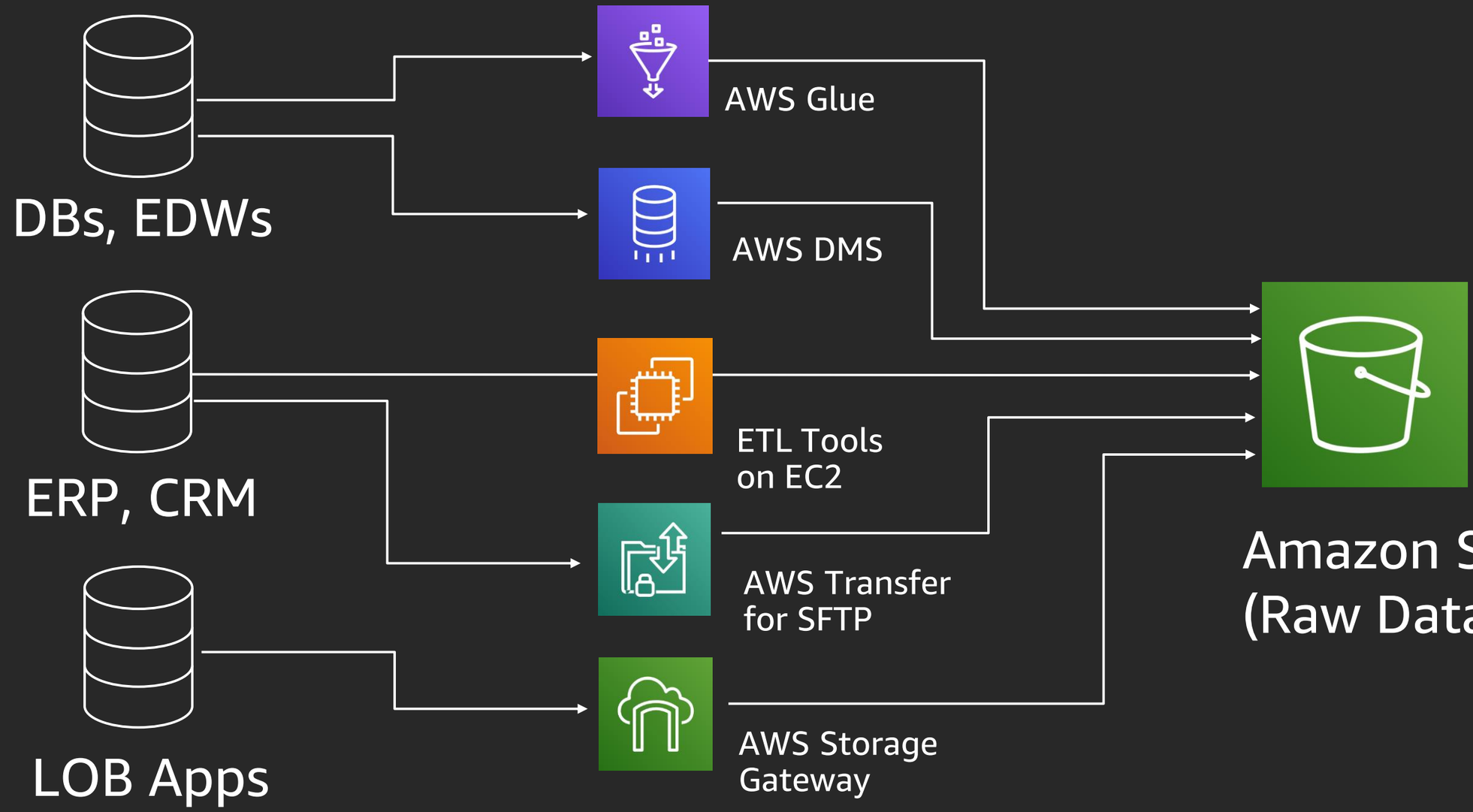
BI reporting, log analysis, data warehousing, usage optimization

Bulk

Machine learning model training, Ad hoc data discovery, data annotation

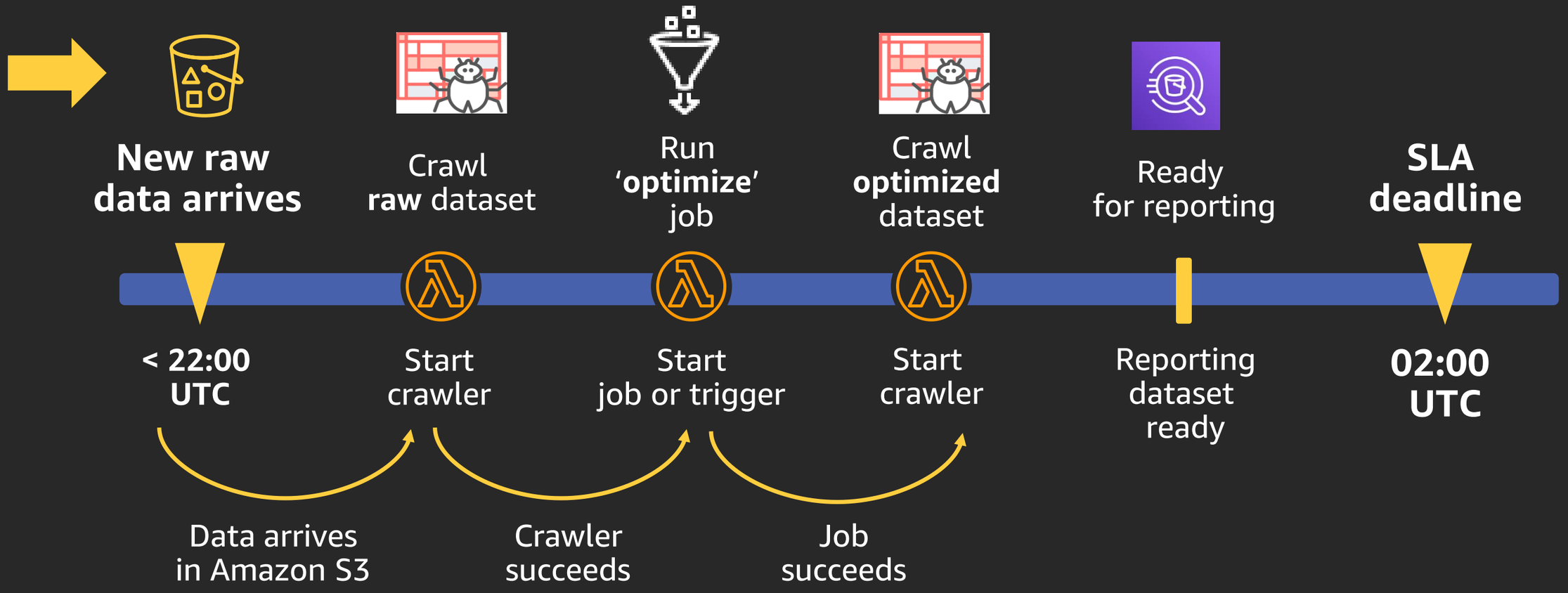
An S3 Data Lake accommodates a wide variety of concurrent data sources

Batch relational data ingestion



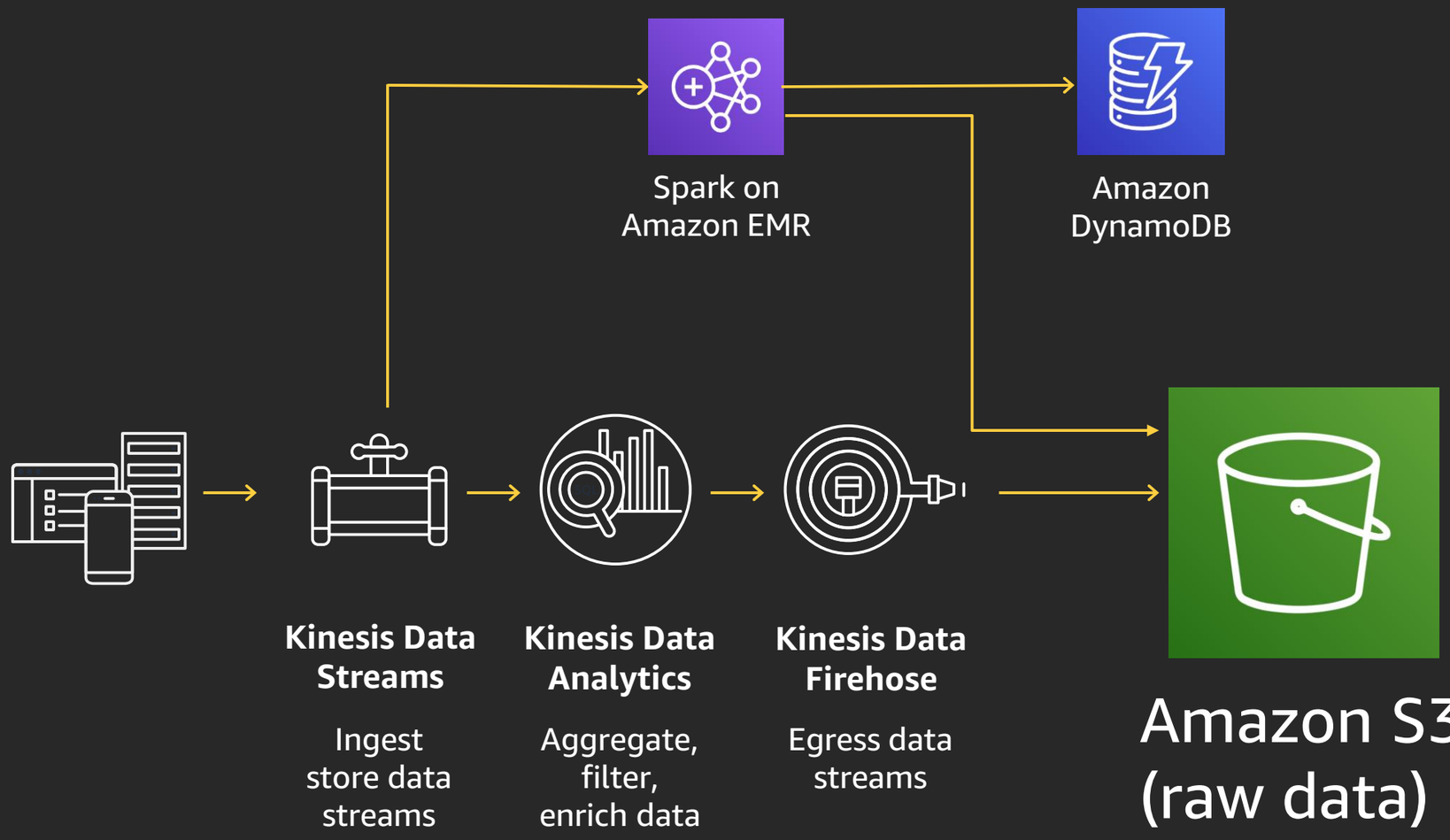
Event-driven batch ingest pipeline

Let Amazon CloudWatch Events and AWS Lambda drive the pipeline



Real-time data ingestion

Collect, process, analyze and aggregate data streams in real time



Streaming data collected and processed in fast layer

Aggregated and batched before ingesting in S3

Provides real-time insights and query

Aggregated raw data stored for further analysis

Running analytics on AWS Data Lakes

Lift & Shift

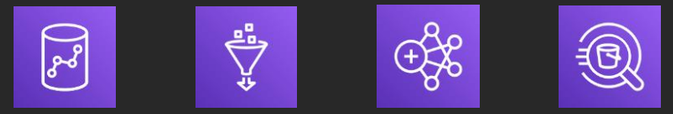
cloudera

databricks

Qubole

Informatica

AWS Managed Services



	Lift & Shift	AWS Managed Services
What	Run third-party analytic tools on EC2 Use EBS and S3 as data stores Self-managed environments	AWS Managed & Serverless Platforms Glue, Athena, EMR, Redshift More options to process data in place
Why	Simplify on-premises migrations Use existing tools, code and customizations Minimize application changes	Focus on data outcomes, not infrastructure Speed adoption of new capabilities More tightly integrated with AWS security
Consider	You provision, manage and scale You monitor and manage availability You own upgrades and versioning	Utilizing AWS Lake Formation Flexibility and choice with open data formats Leverage AWS pace of innovation

Amazon S3 is the storage foundation for both approaches

AWS Lake Formation

Build a secure Data Lake in days

Build Data Lakes quickly

Move, store, catalog, and clean your data faster

Transform to open formats like Parquet and ORC

ML-based de-duplication and record matching

Simplify security management

Centrally define security, governance, and auditing policies

Enforce policies consistently across multiple services

Integrates with IAM and KMS

Provide self-service access to data

Build a data catalog that describes your data

Enable analyst and data scientist to easily find relevant data

Analyze with multiple analytics services without moving data

Performance best practices

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Optimizing Data Lake performance

Scaling request rates on S3

S3 automatically scales to thousands of transactions per second in request performance

At least 3500 PUT/COPY/POST/DELETE and 5500 GET/HEAD request per second **per prefix** in a bucket

Horizontally scale parallel requests to S3 endpoints to distribute load over multiple paths through the network

10 prefixes in an S3 bucket will scale read performance to 55,000 read requests per second

Use the AWS SDK Transfer Manager to automate horizontally scaling connections

No limits to the number of prefixes in a bucket!

Vast majority of analytic use cases don’t require prefix customization

Optimizing Data Lake performance

Scaling request rates on S3

Using the AWS SDK Transfer Manager, the vast majority of applications can use any prefix naming scheme and get thousands of RPS on ingest and egress.

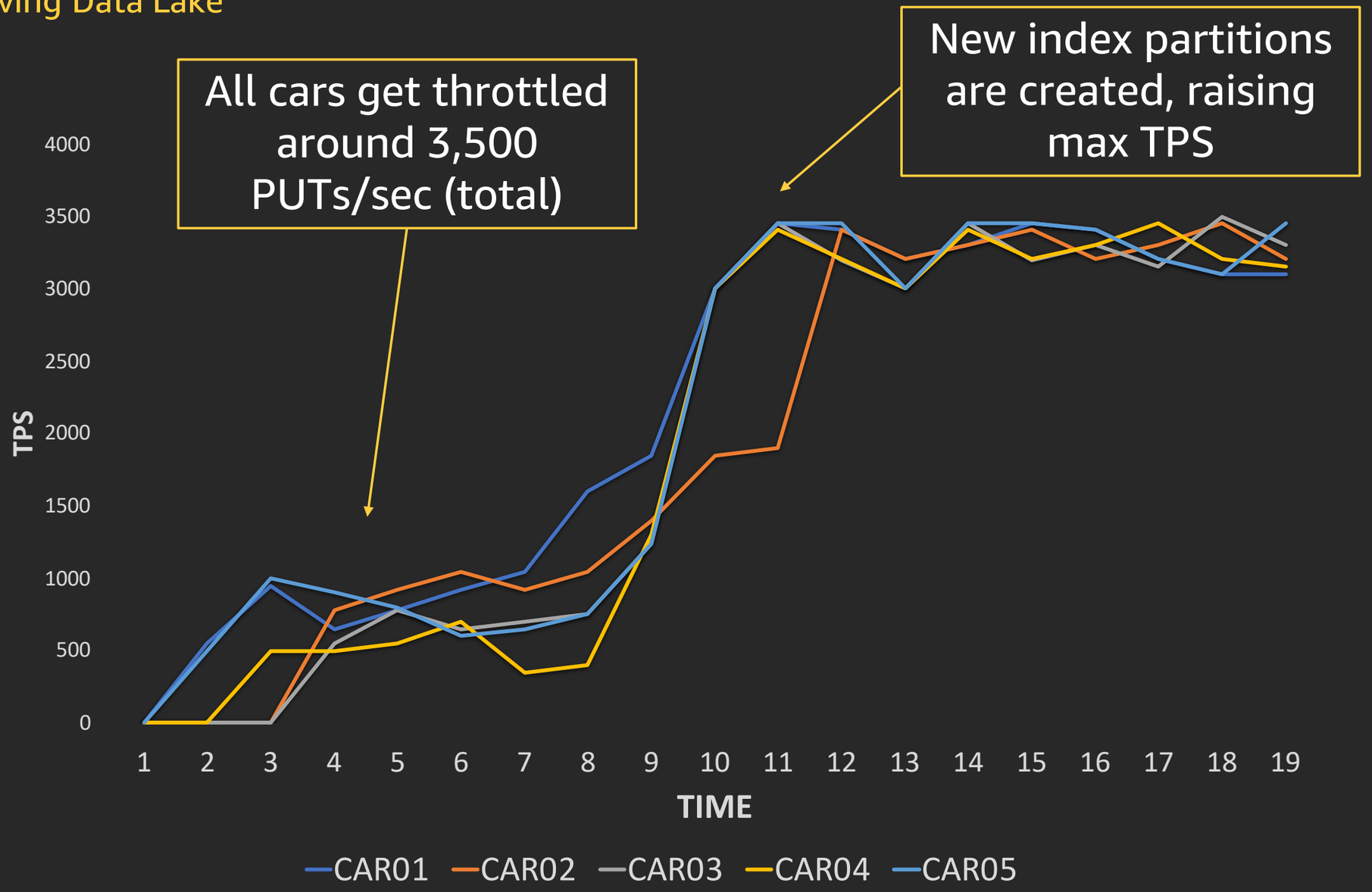
AWS SDK retry logic helps occasional 503 errors while S3 automatically scales for sustained high load.

Only consider prefix customization if:

- Your application exponentially increases RPS in seconds or a few minutes (e.g., 0 RPS to 600K RPS for GET in 5 minutes).
- Your application requires a high RPS on another S3 API like LIST.

Automatic request rate scaling on Amazon S3

Autonomous driving Data Lake



Optimizing Data Lake performance

Use optimized object sizes and data formats

Aim for 16–256 MB object sizes to optimize throughput and cost

- Also reduces LISTs, metadata operations, and job setup time

Aggregate during ingest with Kinesis or during ETL with Glue or EMR+Spark

Utilize Parquet or ORC formats

- Compress by default and are splittable
- Parquet enables parallel queries

Utilize caching and tiering where appropriate

Utilize EMR HDFS namespace for small file Spark workloads

Consider Amazon DynamoDB and ElastiCache for low latency data presentation

Utilize Amazon CloudFront for distributing frequently accessed end user content

Amazon S3 Select

SELECT s.country, s.city from S3Object s where s.city = 'Seattle'

Operates within the Amazon S3 system

SQL statement operates on a per-object basis

Works like GET request, but only returns SQL query filtered results

Supports CSV, JSON, and Parquet formats

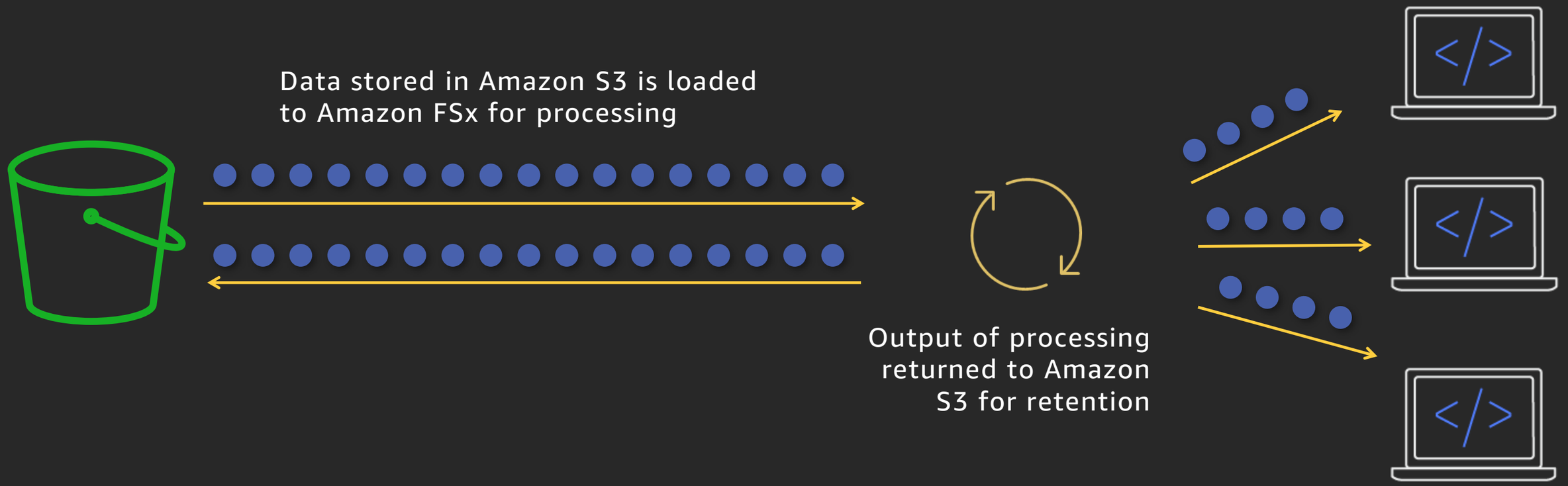
EMR 5.18 and above supports S3 Select for Hive, Presto, and Spark

Scan range queries — up to 10x performance boost for large objects **NEW!**

SELECT a filtered set of data from within an object using standard SQL statements

FSx Lustre for HPC Data Lake workloads

Link your Amazon S3 data set to your Amazon FSx for Lustre file system, then…



When your workload finishes, simply delete your file system.

Security best practices

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Make a secure Data Lake

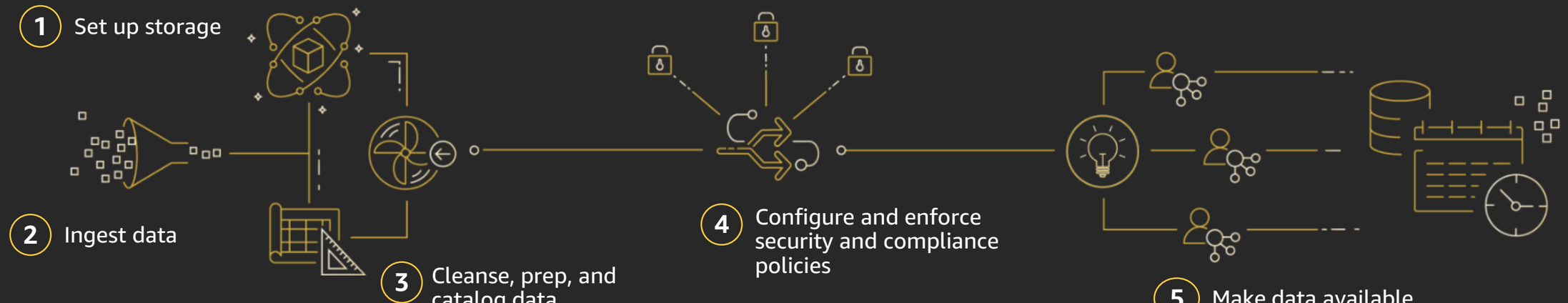
Secure multiple data input sources

Support many unique users and teams

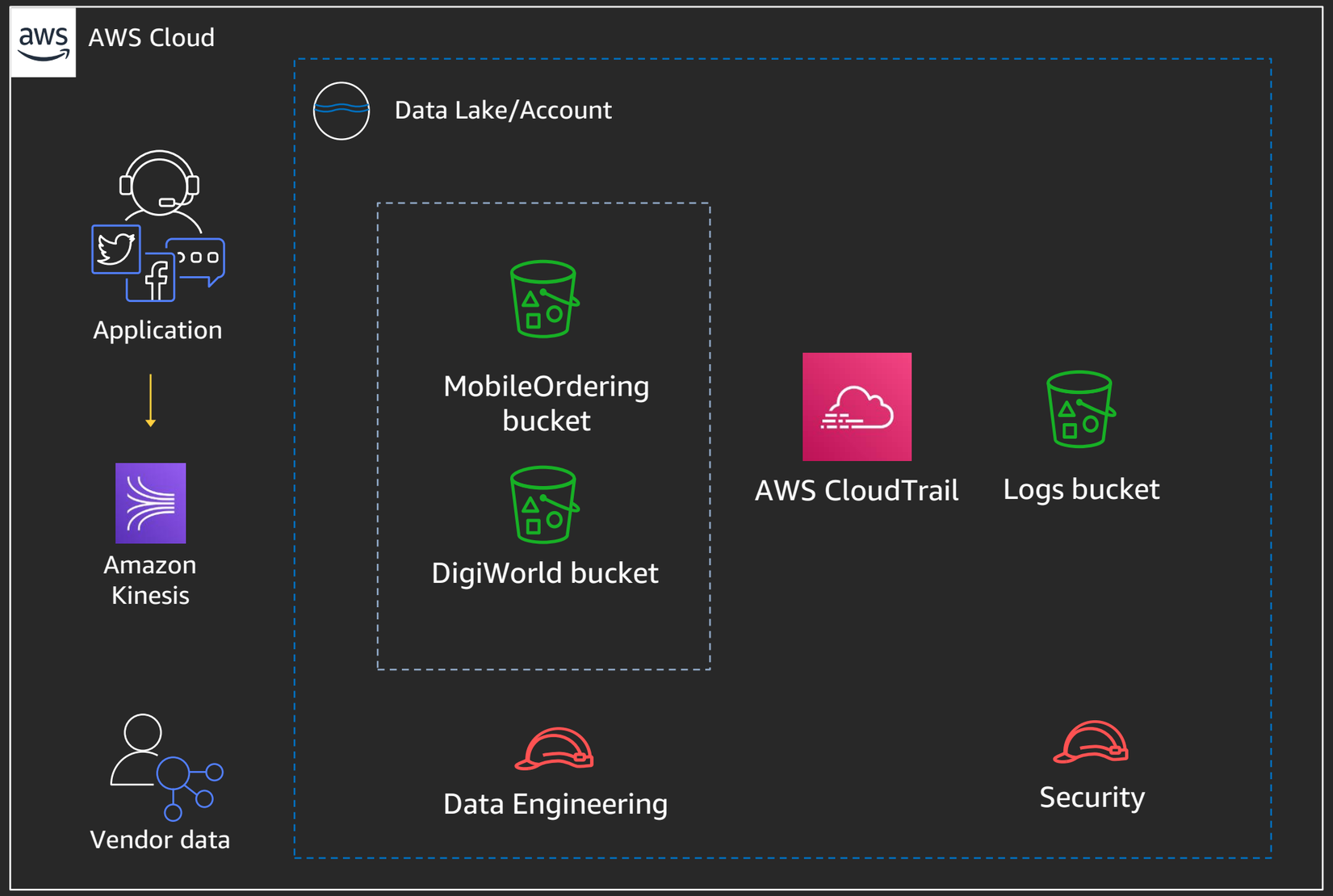
Provide specific access where appropriate

Deny access as default

Typical steps in building a Data Lake

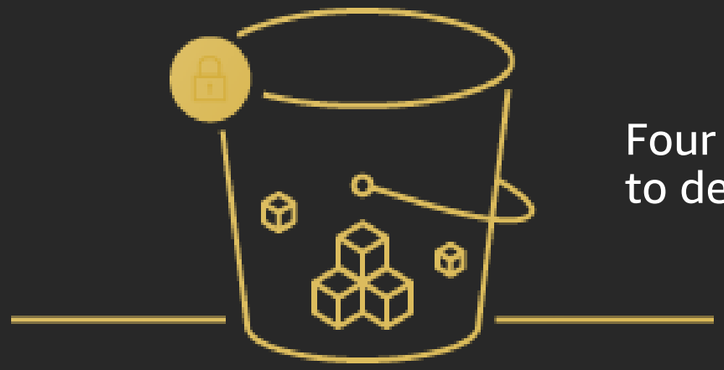


Securing your Data Lake



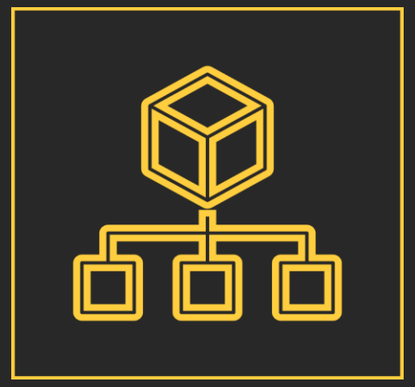
Deny access as default

Amazon S3 Block Public Access



Four security settings
to deny public access

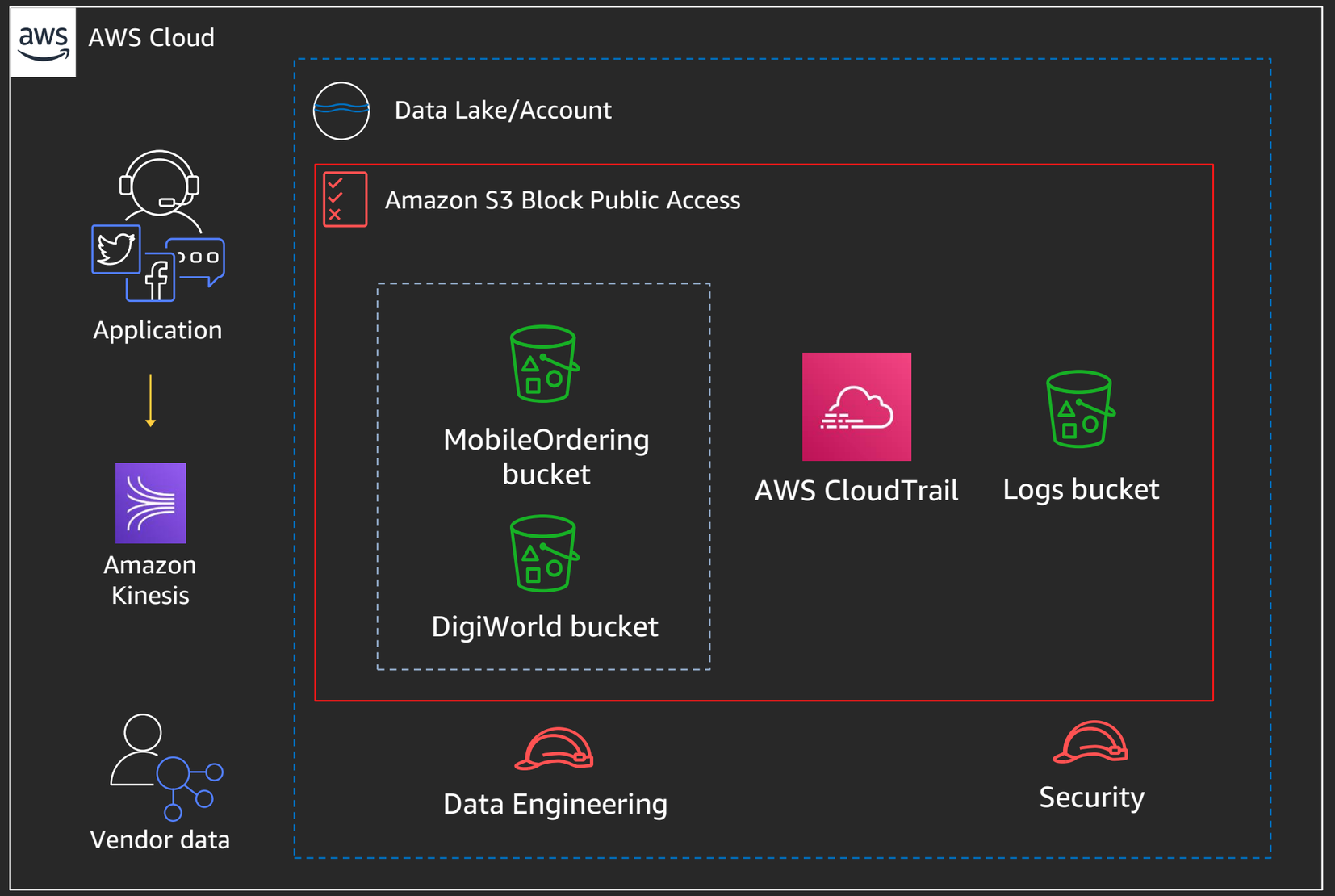
Account-level or
Bucket-level



Use AWS Organizations Service Control Policies (SCP)
to prevent setting changes

https://tinyurl.com/S3BPAdoc

Securing your Data Lake



Deny access as default

Encrypt your data

Amazon S3 encryption support

Encryption in transit — HTTPS/TLS

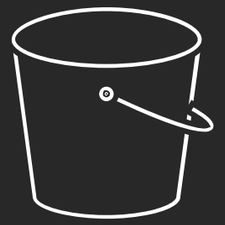
Encryption at rest

Server side
- SSE-S3 (Amazon S3 Managed Keys)
- SSE-KMS (AWS Key Management Service)
- SSE-C (customer-provided keys)

Client side
- Encrypt with the AWS Encryption SDK

https://tinyurl.com/S3EncryptDoc

Amazon S3 default encryption for S3 buckets



One-time bucket level set up

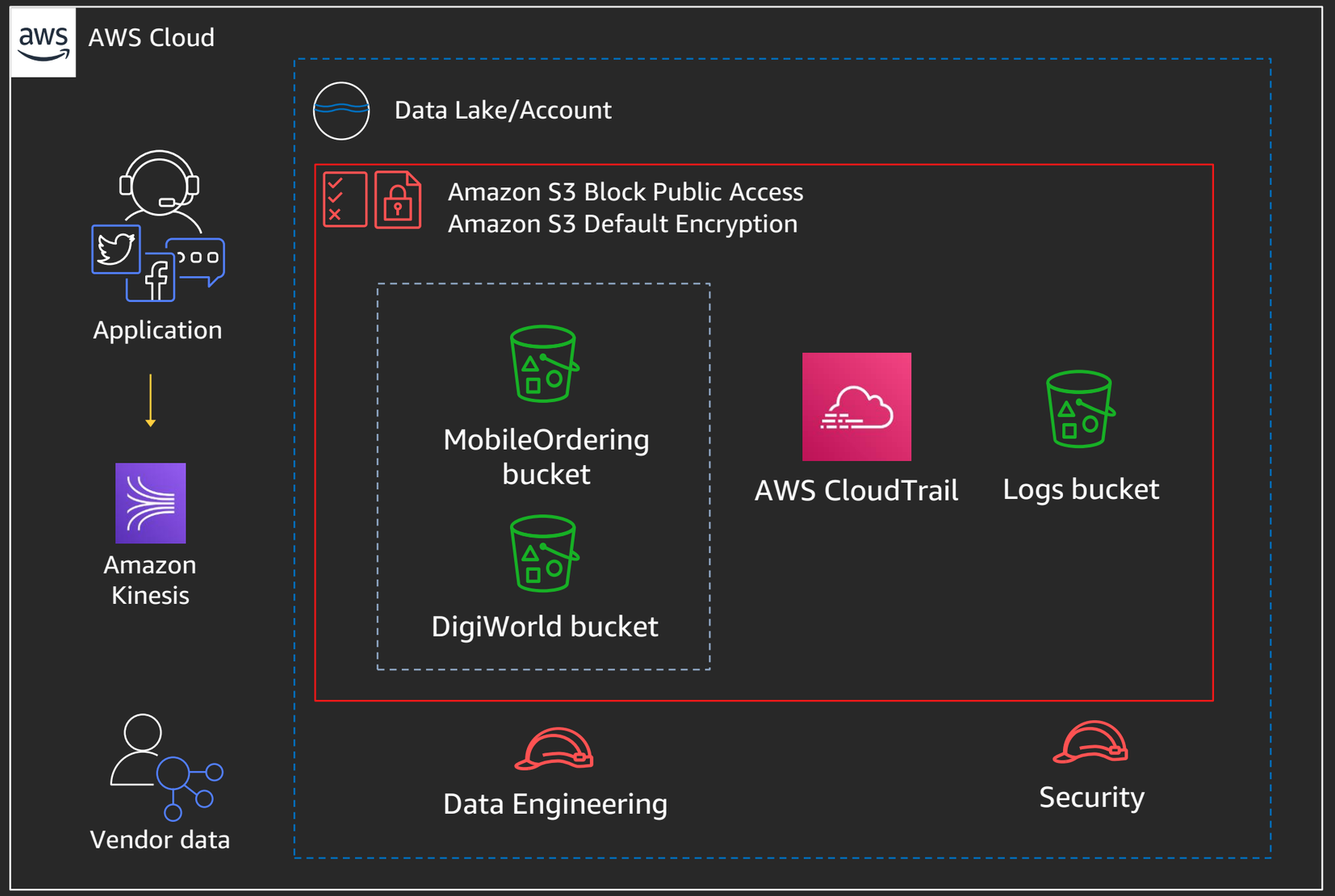
Automatically encrypts all new objects

Simplified compliance

Supports SSE-S3 and SSE-KMS

Defaults S3 encryption-at-rest support for buckets

Securing your Data Lake



Deny access as default

Encrypt your data

Secure multiple data input sources

A couple AWS Identity and Access Management (IAM) terms

Principals

Users
(and groups)

Roles

Applications

Resources

S3 bucket

EMR

IAM user policy and S3 bucket policies

AWS Identity and Access Management (IAM) user policy

What can this user do in AWS?

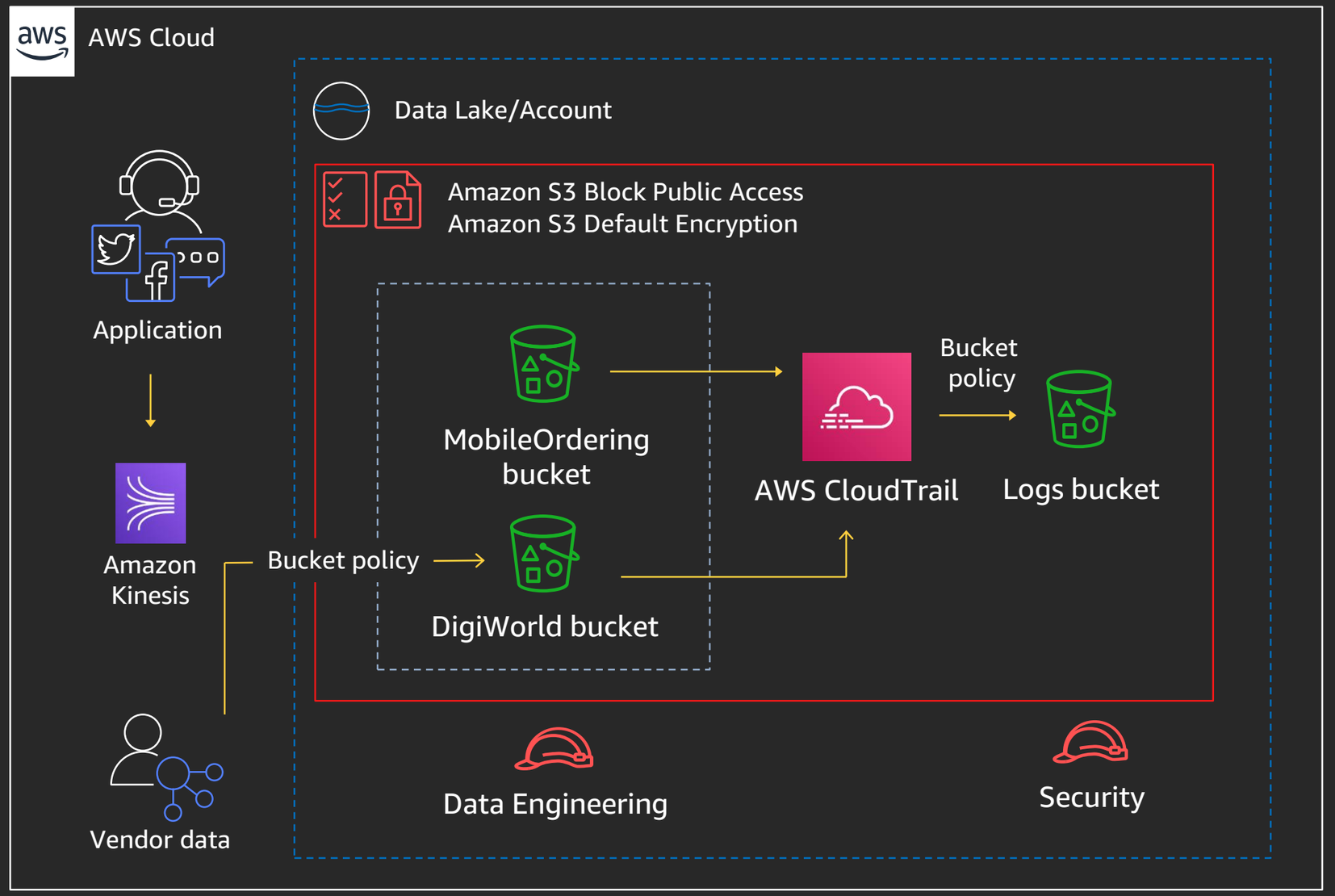
- You prefer to keep access control policies in IAM environment
- No permissions by default
- Controls all AWS Services

Amazon S3 bucket policy

Who can access this S3 resource?

- You prefer to keep access control policies in S3 environment
- Buckets are private by default
- Grant cross-account access to your S3 bucket without using IAM roles

Bucket policy example for your Data Lake



Enable third-party vendor access to input objects into DigiWorld bucket

Principal: AWS account ID for third-party vendor

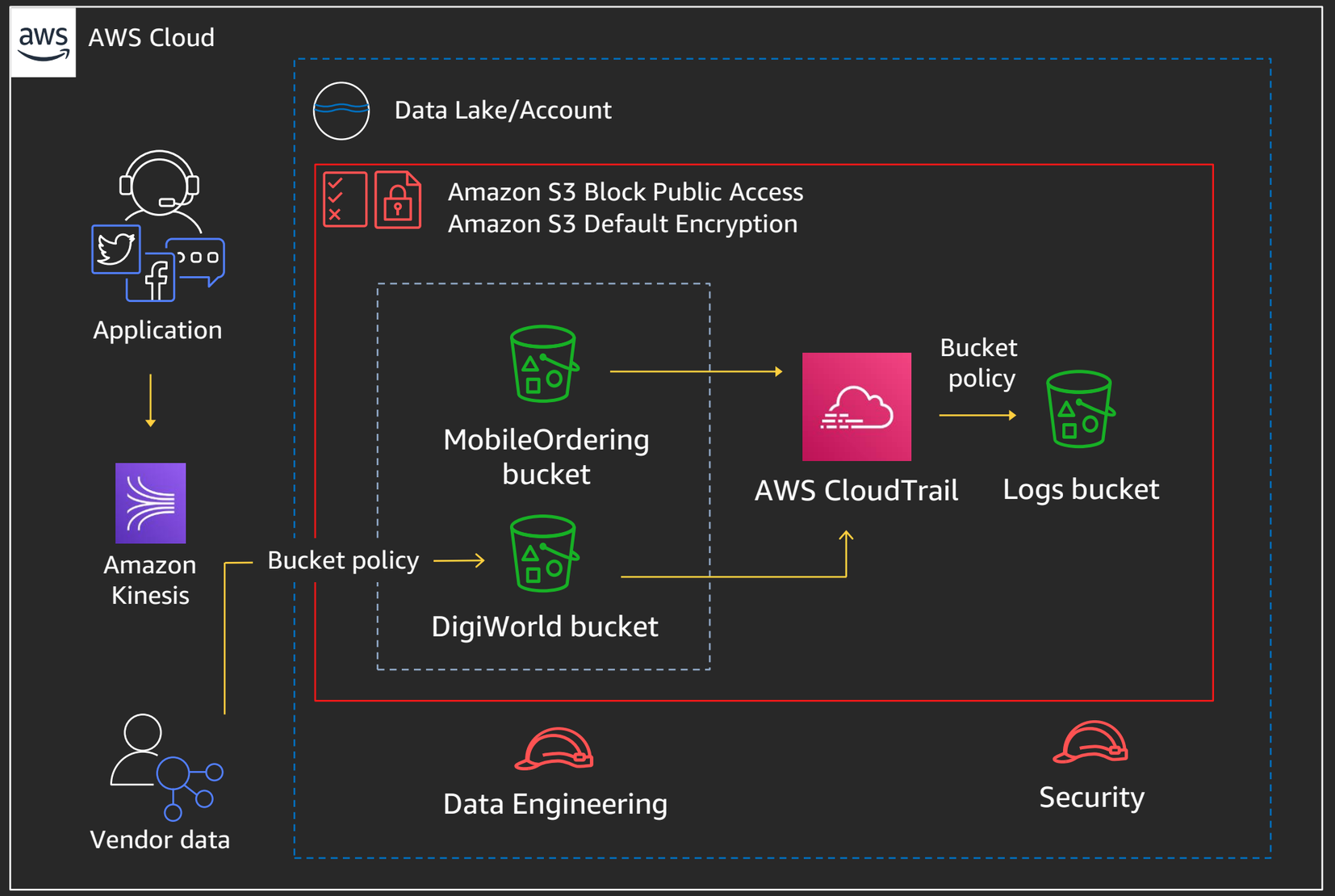
Effect: Allow

Action: PUT object

Resource: DigiWorld bucket (prefix/*)

Condition: s3:x-amz-acl → bucket-owner-full-control

Securing your Data Lake

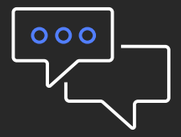


- Deny access as default
- Encrypt your data
- Secure multiple data input sources
- Provide specific access where appropriate
- Support multiple unique users and teams

AWS organizations

Control access and permissions

Audit, monitor, and secure your environment for compliance



Share resources across accounts

Centrally manage costs and billing

Manage and define your organization and accounts

IAM policy examples for your Data Lake

Enable Amazon Kinesis to PUT objects into MobileOrdering bucket

Effect: Allow
Action: PUT object
Resource: MobileOrdering bucket (prefix/*)

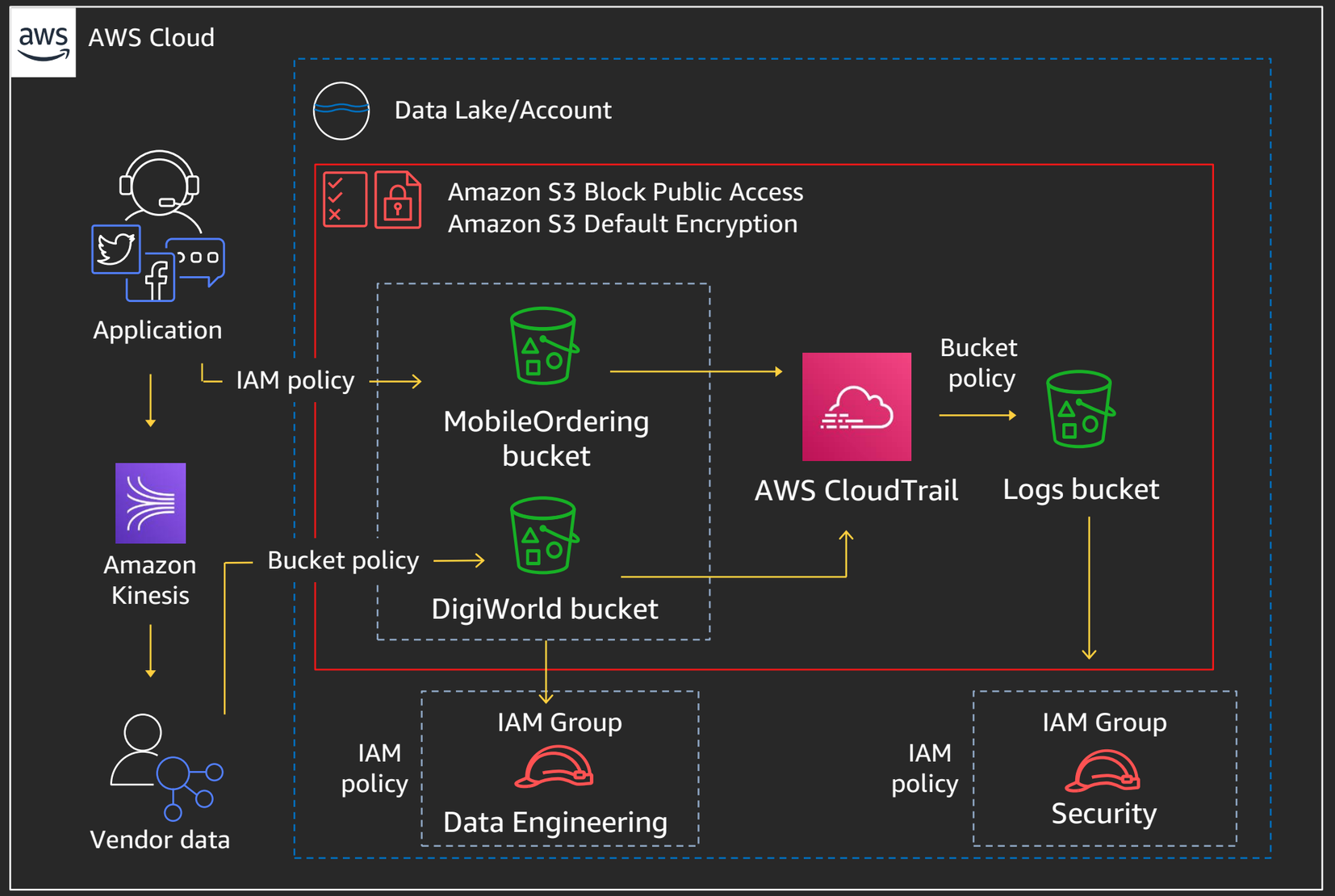
Enable Data Engineering access to MobileOrdering and DigiWorld buckets

Effect: Allow
Action: GET object
Resource: MobileOrdering and DigiWorld buckets (prefix/*)

Enable Security access to Logs bucket

Effect: Allow
Action: GET object
Resource: Project Log bucket (prefix/*)

Securing your Data Lake



- Deny access as default
- Encrypt your data
- Secure multiple data input sources
- Provide specific access where appropriate
- Support multiple unique users and teams

Amazon S3 (Data Lake) security best practices

- (Account) block public access: Enable
- (Bucket) default encryption: SSE or SSE-KMS
- By bucket policy, require TLS
- CloudTrail and S3 Server Access Logs enable security and access audits
- VPC endpoint: Enable and require, with bucket policies limiting access
- MFA delete and object lock governance mode for permanence

STG301 – [Breakout] Deep dive on S3 security and management

Overarching takeaways

- S3 is the foundation for Data Lakes
- Leverage pipelined architectures improve governance, data management, and efficiency
- Improve performance by parallelizing access and scaling horizontally
- Privatize your Data Lake, encrypt everything, and secure specific access to and from your Data Lake

Related breakouts

[STG314] [Workshop] Building a Data Lake on Amazon S3

[STG340] [Chalk talk] What to consider when building a Data Lake on Amazon S3

[ARC345] [Chalk talk] Architecting Data Lakes with AWS data and analytics services

[STG301] [Breakout] Deep dive on Amazon S3 security and management

[STG308] [Chalk talk] Deep dive on security in Amazon S3 and Amazon S3 Glacier

[STG356] [Chalk talk] Managing access to Amazon S3 buckets

[STG363] [Builders session] Managing access to Amazon S3 buckets at scale

[STG334] [Chalk talk] Optimizing performance on Amazon S3

Fastest way to complete your alert

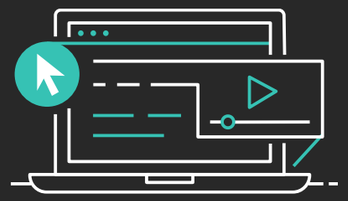
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Please complete the session survey in the mobile app.

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Learn storage with AWS Training and Certification

Resources created by the experts at AWS to help you build cloud storage skills



45+ free digital courses cover topics related to cloud storage, including:

- Amazon S3
- AWS Storage Gateway
- Amazon S3 Glacier
- Amazon Elastic File System (Amazon EFS)
- Amazon Elastic Block Store (Amazon EBS)

Classroom offerings, like Architecting on AWS, feature AWS expert instructors and hands-on activities

Visit aws.amazon.com/training/path-storage/

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws training and certification

Thank you!

Amy Che
amyche@amazon.com

John Mallory
johmallo@amazon.com

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws