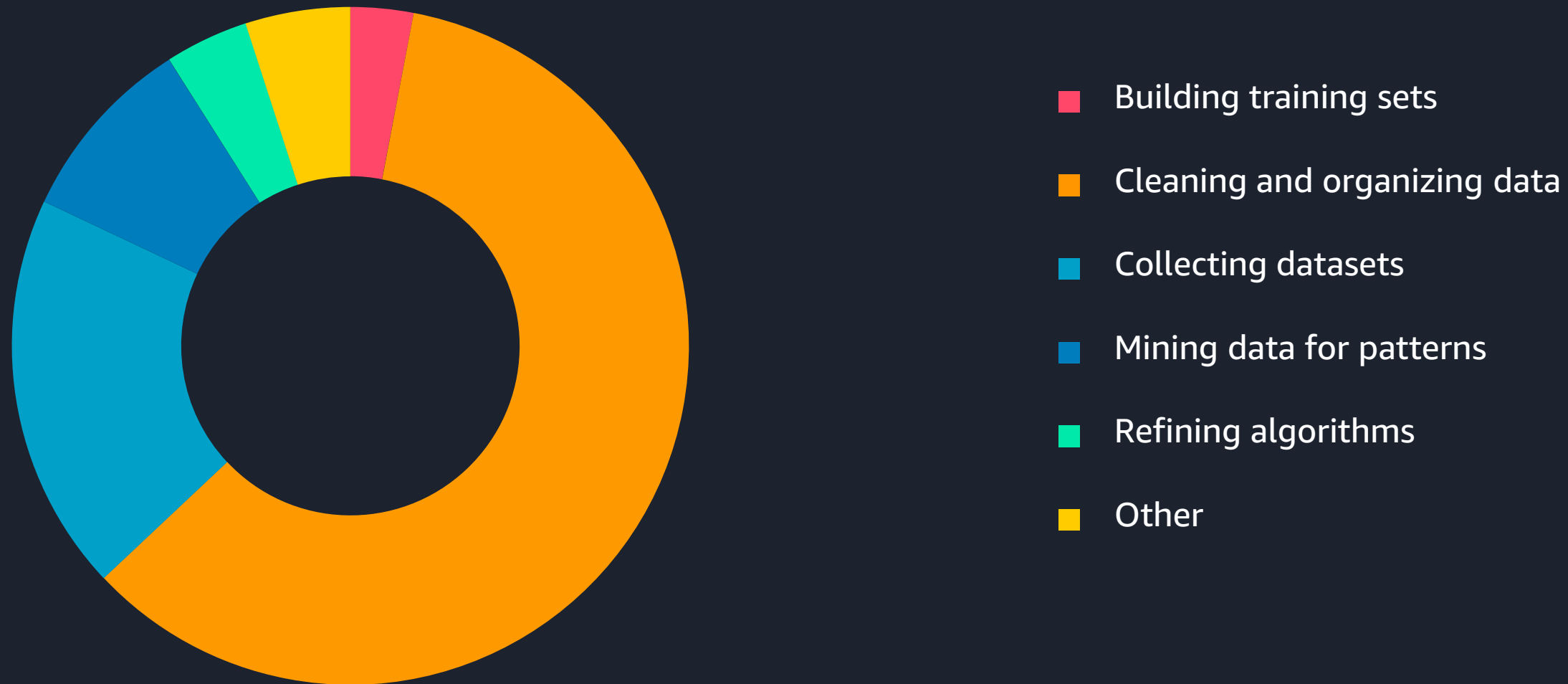


I want to get the most value out of my data.....fast!



Data preparation accounts for ~80% of the work ...





“Without AWS Lake Formation, it would have been impossible to achieve the goal of a scalable, easy to use security layer for all data on Amazon S3. It was easy to set up and apply fine-grained access controls based on user personas”



Build Your Data Lake on Amazon S3 in Days

Kumar Nachiketa

Senior Partner Solutions Architect, AWS

Agenda

Trends driving the revolution

What are data lakes?

What's hard today?

AWS Lake Formation makes data lakes easy

Demo



In the past, decision-making ...

...revolved around the **enterprise data warehouse**



Data no longer fits



There is **more data** than people think

Data is **more diverse**

Data	Data platforms need to	
grows >10x every 5 years	live for 15 years	scale 1,000x

* IDC, Data Age 2025: The Evolution of Data to Life-Critical: Don't Focus on Big Data, Focus on the Data That's Big, April 2017.



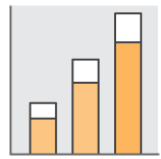
Broader workloads



Data Scientists



Business Users



Analysts



Applications

**Machine
learning**

Scientific

SQL analytics

**Real-time,
streaming**

There are **more people**
accessing data

who want to **analyze it**
in different ways

Amazon S3 as the foundation for data lakes



Durable, available, exabyte-scalable

Secure, compliant, auditable

High performance

Low-cost storage and analytics

Broad network integration



What are data lakes?

Data lake: The new information hub

A **centralized, secure repository** enables you to **govern, discover, share, and analyze structured and unstructured data** at any scale

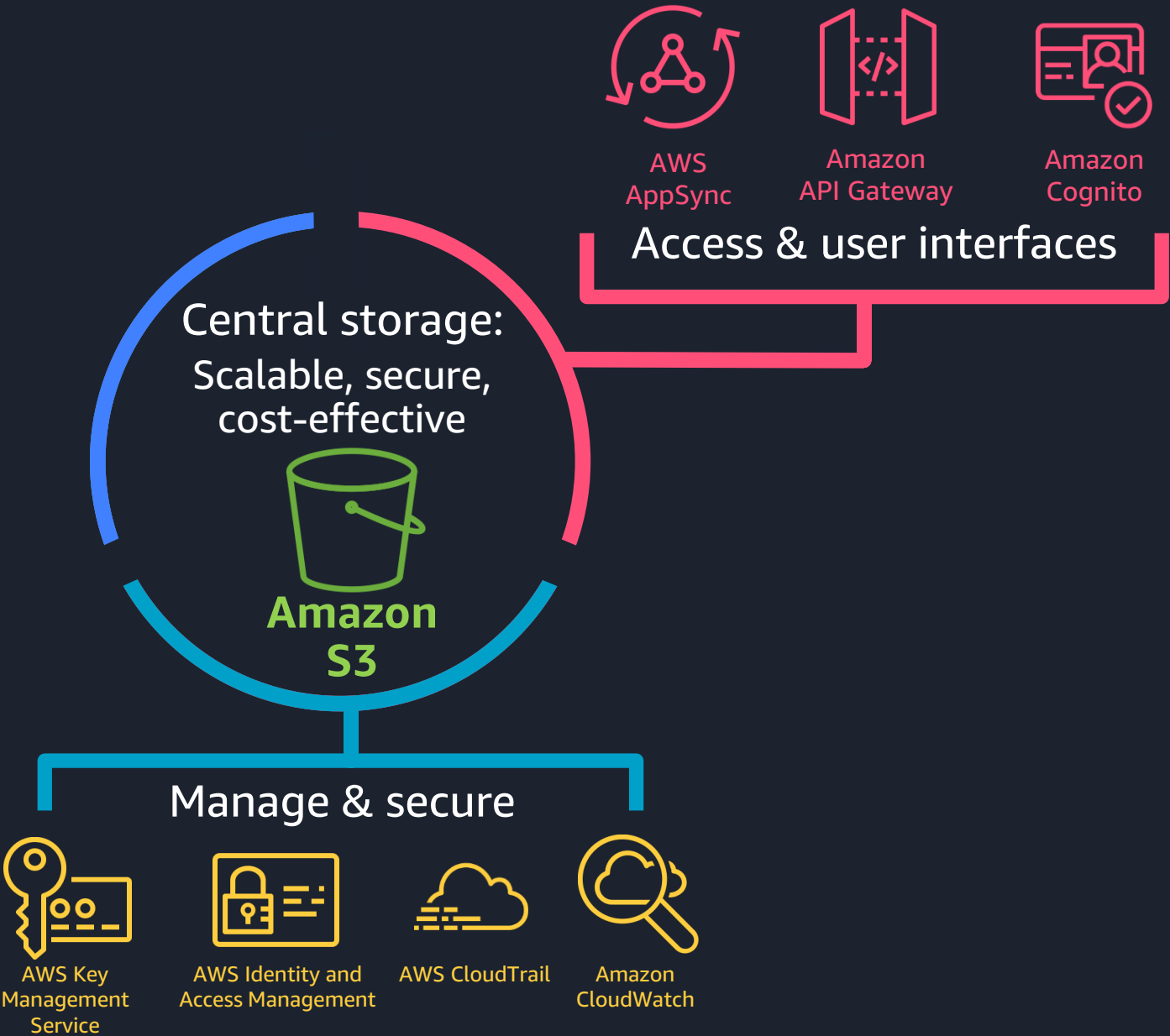
Data lake on AWS – S3 at the core



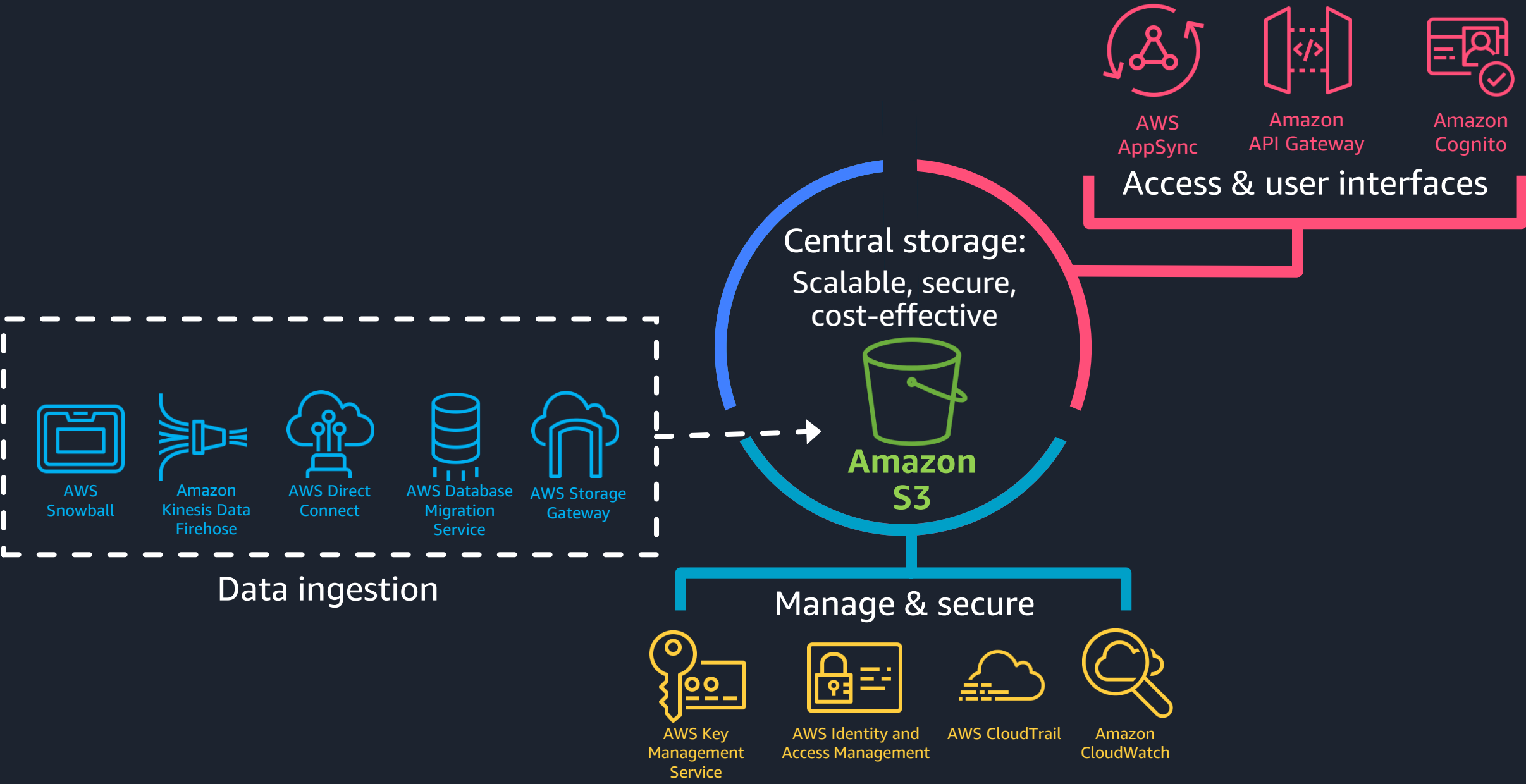
Data lake on AWS – access and user interface



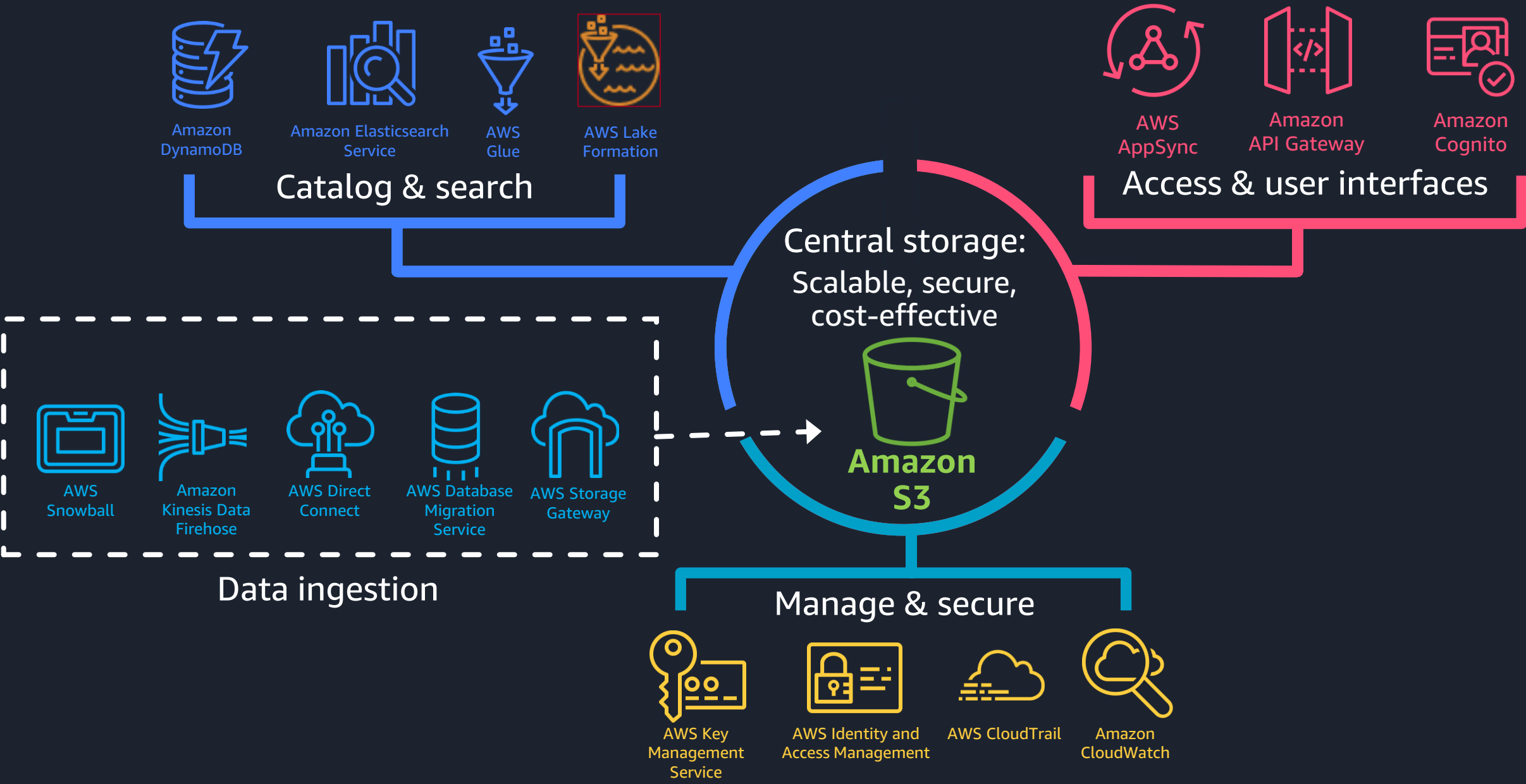
Data lake on AWS – manage and secure



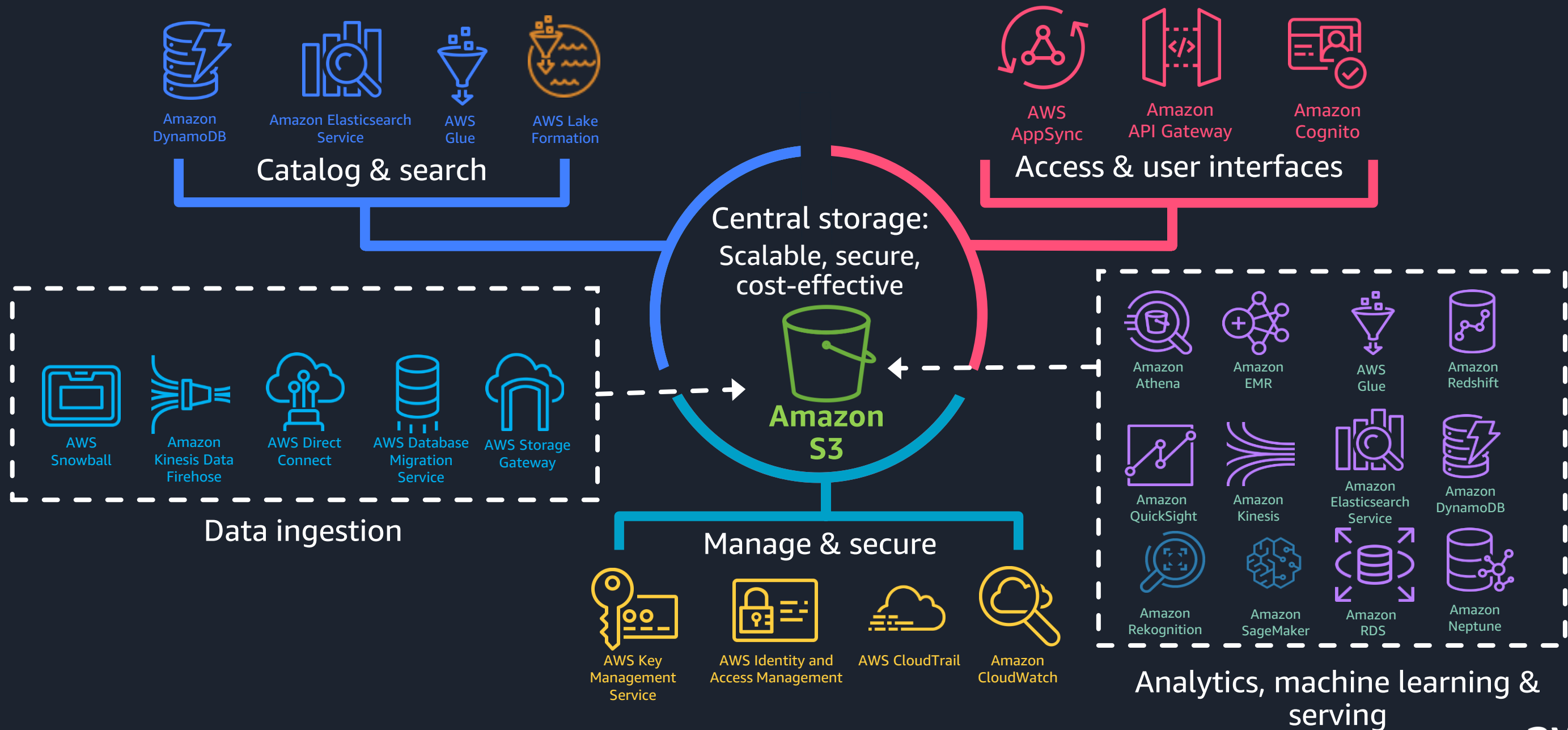
Data lake on AWS – data ingestion



Data lake on AWS – catalog & search



Data lake on AWS – analytics, ML, and serving



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 S3 Glacier Deep Archive

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

Optimize costs for all stages of data lake workflows

More data lakes & analytics on AWS than anywhere else



Using AWS Lake Formation...



What customers are saying about AWS Lake Formation



*AWS Lake Formation is a **one stop shop** for setting up granular user and service access to IATA's AWS based Data Lake. It allows us to simplify our IAM access policies as well as **easily manage and audit access rights and permissions**.*

Bogdan Pasol
Manager BI
IATA



Onefootball

*Blueprints helped eliminate complexity and created better maintainability by **simplifying data ingestions into our centralized data lake and eliminating ETL workloads**.*

Stephan Durry
Head of Data & Insight
Onefootball



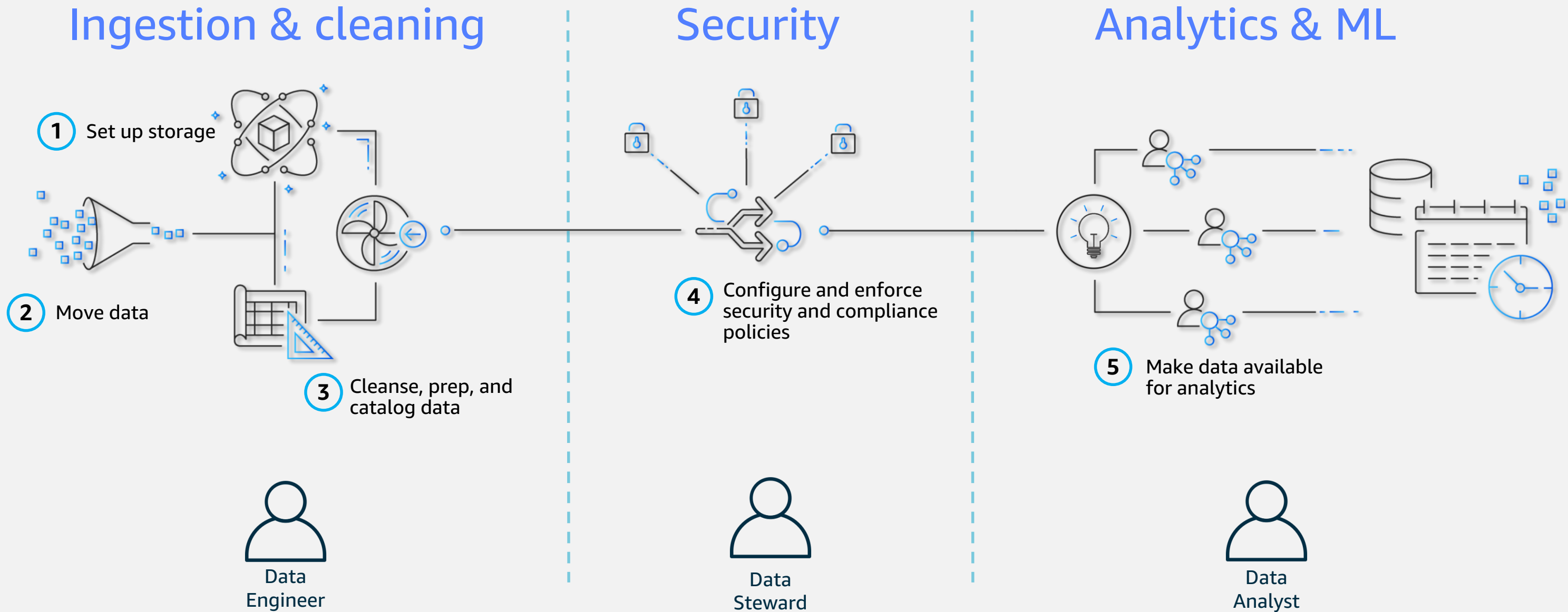
*Data normalization is critical in providing better patient outcomes by bringing transparency to benchmark pricing data for clinical and medical products. **Using ML Transforms, we process data sets in four hours, down from one week, and our accuracy improved to near 100%**.*

Nic Sagez
CTO
Curvo Labs

What's hard today?

Building **clean** and **secure** data lakes
can take months

Typical steps of building a data lake



Sample of steps required

Find sources

The screenshot shows the AWS Management Console interface for Amazon RDS. The left sidebar contains navigation options: Dashboard, Instances (selected), Clusters, Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, and Recommendations (14). The main content area is titled 'RDS > Instances' and shows 'Instances (7)'. At the top of the main area are buttons for 'Instance actions', 'Restore from S3', and 'Create database'. Below these is a search bar labeled 'Filter instances'. The main area contains a table with the following data:

DB instance	Engine	Status	CPU
blueprint-source-db-instance	MySQL	available	0.29%
jdbc-mariadb	MariaDB	available	1.00%
mysql-test	MySQL	available	1.36%
oracle-test	Oracle Enterprise Edition	available	1.33%
oracle-test2	Oracle Enterprise Edition	available	1.48%
postgres-test	PostgreSQL	available	1.56%
sqlserver-test	SQL Server Express Edition	available	35.93%

Sample of steps required

Create Amazon Simple Storage Service (Amazon S3) locations

The screenshot shows the AWS Management Console interface for Amazon S3. The left sidebar contains navigation options like Dashboard, Instalar, Clusters, Query, Performance, Snapshots, Automation, Reservations, Subnets, Parameters, Options, Events, and Reconstructions. The main content area is titled 'Amazon S3' and shows 'Buckets' with a search bar and filters. Below the search bar are buttons for '+ Create bucket', 'Edit public access settings', 'Empty', and 'Delete'. Summary statistics show 67 Buckets and 14 Regions. A table lists several buckets with their names, public access status, regions, and creation times.

Bucket Name	Public Access Status	Region	Creation Time
awsglue-datasets-ap-northeast-1	Objects can be public	Asia Pacific (Tokyo)	Aug 11, 2017 6:10:37 PM GMT-0700
awsglue-datasets-ap-northeast-2	Objects can be public	Asia Pacific (Seoul)	Aug 11, 2017 6:07:26 PM GMT-0700
awsglue-datasets-ap-south-1	Objects can be public	Asia Pacific (Mumbai)	Aug 11, 2017 6:05:49 PM GMT-0700
awsglue-datasets-ap-southeast-1	Objects can be public	Asia Pacific (Singapore)	Aug 11, 2017 6:07:54 PM GMT-0700
awsglue-datasets-ap-southeast-2	Objects can be public	Asia Pacific (Sydney)	Aug 11, 2017 6:10:08 PM GMT-0700
awsglue-datasets-ca-central-1	Objects can be public	Canada (Central)	Aug 11, 2017 6:11:09 PM GMT-0700
awsglue-datasets-eu-central-1	Objects can be public	EU (Frankfurt)	Aug 11, 2017 6:11:37 PM GMT-0700
awsglue-datasets-eu-west-1	Objects can be public	EU (Ireland)	Aug 11, 2017 6:12:00 PM GMT-0700

Sample of steps required

Configure access policies

Public access settings | Access Control List | **Bucket Policy** | CORS configuration

Bucket policy editor ARN: arn:aws:s3:::awsglue-datasets-us-east-1

Type to add a new policy or edit an existing policy in the text area below.

```
1 {
2   "Id": "Policy1543402505352",
3   "Version": "2018-11-28",
4   "Statement": [
5     {
6       "Sid": "Stmt1543402503273",
7       "Action": [
8         "s3:GetObject",
9         "s3:ListBucket",
10        "s3:ListBucketByTags",
11        "s3:PutObject"
12      ],
13       "Effect": "Allow",
14       "Resource": "arn:aws:s3:::awsglue-datasets-us-east-1",
15       "Principal": {
16         "AWS": [
17           "arn:aws:iam::<account#>:user/test-user"
18         ]
19       }
20     }
21   ]
22 }
```

[Documentation](#) | [Policy generator](#)

Sample of steps required

Map tables to Amazon S3 locations

The screenshot shows the AWS Glue console interface. A modal window titled 'Add table' is open, displaying the 'Define a schema' step. On the left, a sidebar lists navigation options like 'Dashboards', 'Instants', 'Clusters', 'Queries', etc. The main content area shows the following configuration:

- Table properties:** Name: githubarchive_demo, Database: githubarchive
- Data store:** s3://awsglue-datasets-us-east-1/
- Data format:** JSON
- Schema:** (Selected step)
- Review:** (Not selected)

The 'Define a schema' section includes an 'Add column' button and a table with the following columns: Column name, Data type, Key, and Comment. The table contains three rows:

	Column name	Data type	Key	Comment
1	user_id	string		
2	event_type	string		
3	payload	STRUCT		

At the bottom of the modal, there are 'Back' and 'Next' buttons. The 'Next' button is highlighted in blue.

Sample of steps required

ETL jobs to load and clean data

The screenshot displays the AWS Glue console interface for configuring an ETL job named 'github_2_csv'. The job is set to run on an Amazon S3 bucket. The configuration includes a source table '2015' from the 'gitarchive' database. The job is composed of four sequential transformations: 'ApplyMapping', 'ResolveChoice', 'DropNullFields', and 'DataSink'. The 'DataSink' transformation is configured to write the output to an S3 path 's3://glue-sample-target/output-dir' in Parquet format. The console also shows a preview of the PySpark code used for the job, which includes imports for various Glue and Spark classes, and logic for reading data from the source table, applying mappings, resolving choices, dropping null fields, and writing the final output to the target S3 location.

Job: github_2_csv [Action] [Save] [Run job] [Generate diagram] [Insert template at cursor] [Source] [Target] [Target Location] [Transform] [Spigot] [?] [X]

Database Name gitarchive
Table Name 2015

Transform Name ApplyMapping

Transform Name ResolveChoice

Transform Name DropNullFields

Path s3://glue-sample-target/output-dir

```
1 |import sys
2 |from awsglue.transforms import *
3 |from awsglue.utils import getResolvedOptions
4 |from pyspark.context import SparkContext
5 |from awsglue.context import GlueContext
6 |from awsglue.job import Job
7
8 |## @params: [JOB_NAME]
9 |args = getResolvedOptions(sys.argv, ['JOB_NAME'])
10
11 |sc = SparkContext()
12 |glueContext = GlueContext(sc)
13 |spark = glueContext.spark_session
14 |job = Job(glueContext)
15 |job.init(args['JOB_NAME'], args)
16 |## @type: DataSource
17 |## @args: [database = "gitarchive", table_name = "2015", transformation_ctx = "datasource0"]
18 |## @return: datasource0
19 |## @inputs: []
20 |datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "gitarchive", table_name = "2015", transformation_ctx = "datasource0")
21 |## @type: ApplyMapping
22 |## @args: [mapping = [{"id", "string", "id", "string"}, {"type", "string", "type", "string"}, {"actor.login", "string", "actor", "string"}, {"repo.name", "string", "repo", "string"}]]
23 |## @return: applymapping1
24 |## @inputs: [frame = datasource0]
25 |applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [{"id", "string", "id", "string"}, {"type", "string", "type", "string"}, {"actor.login", "string", "actor", "string"}, {"repo.name", "string", "repo", "string"}])
26 |## @type: ResolveChoice
27 |## @args: [choice = "make_struct", transformation_ctx = "resolvechoice2"]
28 |## @return: resolvechoice2
29 |## @inputs: [frame = applymapping1]
30 |resolvechoice2 = ResolveChoice.apply(frame = applymapping1, choice = "make_struct", transformation_ctx = "resolvechoice2")
31 |## @type: DropNullFields
32 |## @args: [transformation_ctx = "dropnullfields3"]
33 |## @return: dropnullfields3
34 |## @inputs: [frame = resolvechoice2]
35 |dropnullfields3 = DropNullFields.apply(frame = resolvechoice2, transformation_ctx = "dropnullfields3")
36 |## @type: DataSink
37 |## @args: [connection_type = "s3", connection_options = {"path": "s3://glue-sample-target/output-dir"}, format = "parquet", transformation_ctx = "datasink4"]
38 |## @return: datasink4
39 |## @inputs: [frame = dropnullfields3]
40 |datasink4 = glueContext.write_dynamic_frame.from_options(frame = dropnullfields3, connection_type = "s3", connection_options = {"path": "s3://glue-sample-target/output-dir"}, format = "parquet", transformation_ctx = "datasink4")
41 |job.commit()
```

Logs Schema

Documentation Policy generator

Sample of steps required

The screenshot shows the AWS IAM console 'Create policy' wizard. The 'JSON' tab is selected, and the following policy is being defined:

```
1 | import sys
2 | from aws glue.transforms import
3 |
4 |
5 |
6 | "Effect": "Allow",
7 | "Action": [
8 |     "glue:GetTables"
9 | ],
10 | "Resource": [
11 |     "arn:aws:glue:us-west-2:123456789012:catalog",
12 |     "arn:aws:glue:us-west-2:123456789012:database/db1",
13 |     "arn:aws:glue:us-west-2:123456789012:table/db1/store_sales",
14 |     "arn:aws:glue:us-west-2:123456789012:table/db1/stores"
15 | ]
```

A blue callout box with the text "Create metadata access policies" is overlaid on the top right of the wizard. The wizard includes buttons for "Cancel" and "Review policy".

Sample of steps required

Configure access from analytics services

The screenshot shows the AWS IAM console interface for configuring a policy. The main content area displays SQL GRANT statements and a table of access privileges.

```
[finals=#
[finals=#
[finals=#
[finals=#
[finals=#
[finals=# grant select(sid,name), update, insert ON grades to test_user;
GRANT
[finals=# grant select on enrolled to test_user;
GRANT
[finals=# \dp+
```

Schema	Name	Type	Access privileges	Column privileges	Policies
public	enrolled	table	mashah=awdDxt/mashah+ test_user=r/mashah		
public	enrolled_scores	table			
public	grades	table	mashah=awdDxt/mashah+ test_user=aw/mashah	sid: + test_user=r/mashah+ name: + test_user=r/mashah	
public	overall_pct	table			
public	scores	table			

finals=# █

Buttons: Cancel, Review policy

Footer: Documentation, Policy generator

Sample of steps required

Rinse and repeat for other:
Datasets, users, and end-services

And more:
manage and monitor ETL jobs
update metadata catalog as data changes
update policies across services as users and permissions change
manually maintain cleansing scripts
create audit processes for compliance

...

Manual | Error-prone | Time consuming

[Documentation](#) [Policy generator](#)

AWS Lake Formation makes data lakes easy

Fully managed service that enables



data engineers



data stewards



data analysts

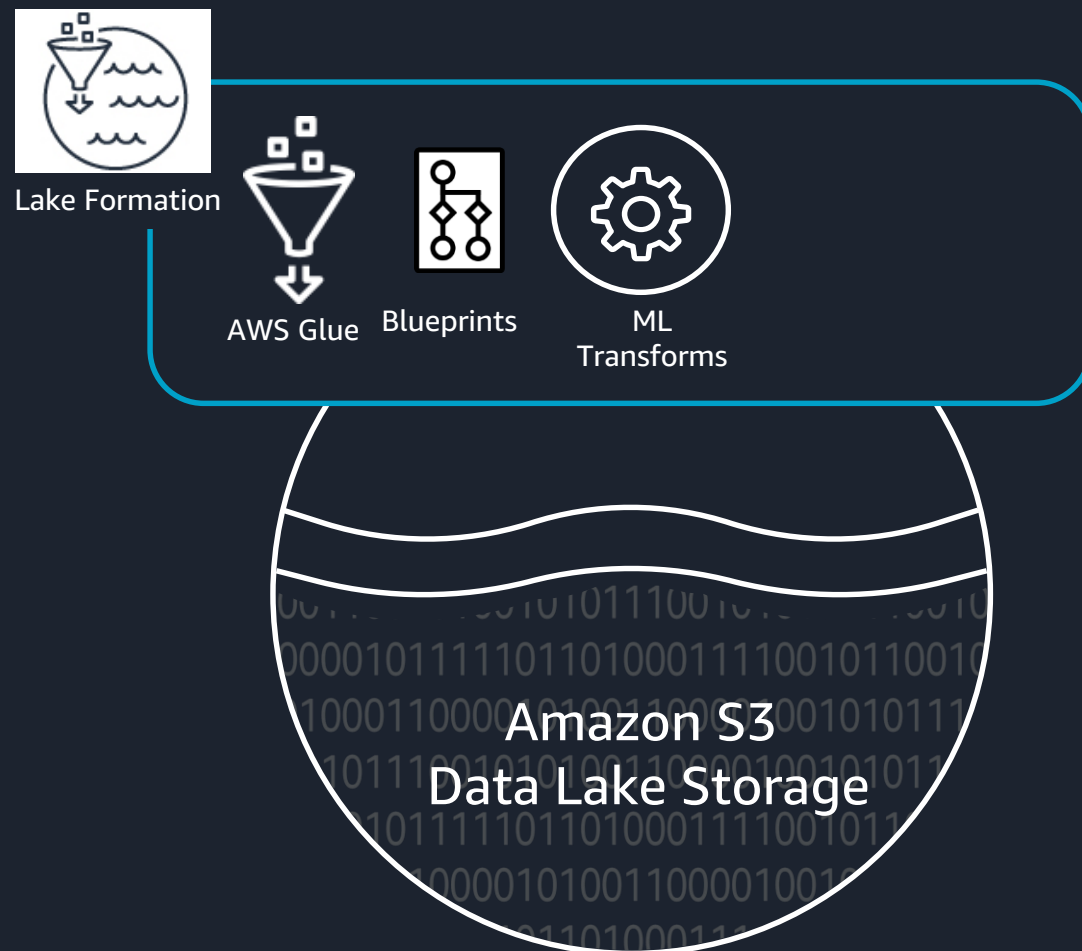
to build clean and secure data lakes in days

AWS Lake Formation solution stack



Cost-effective, durable storage with global replication capabilities

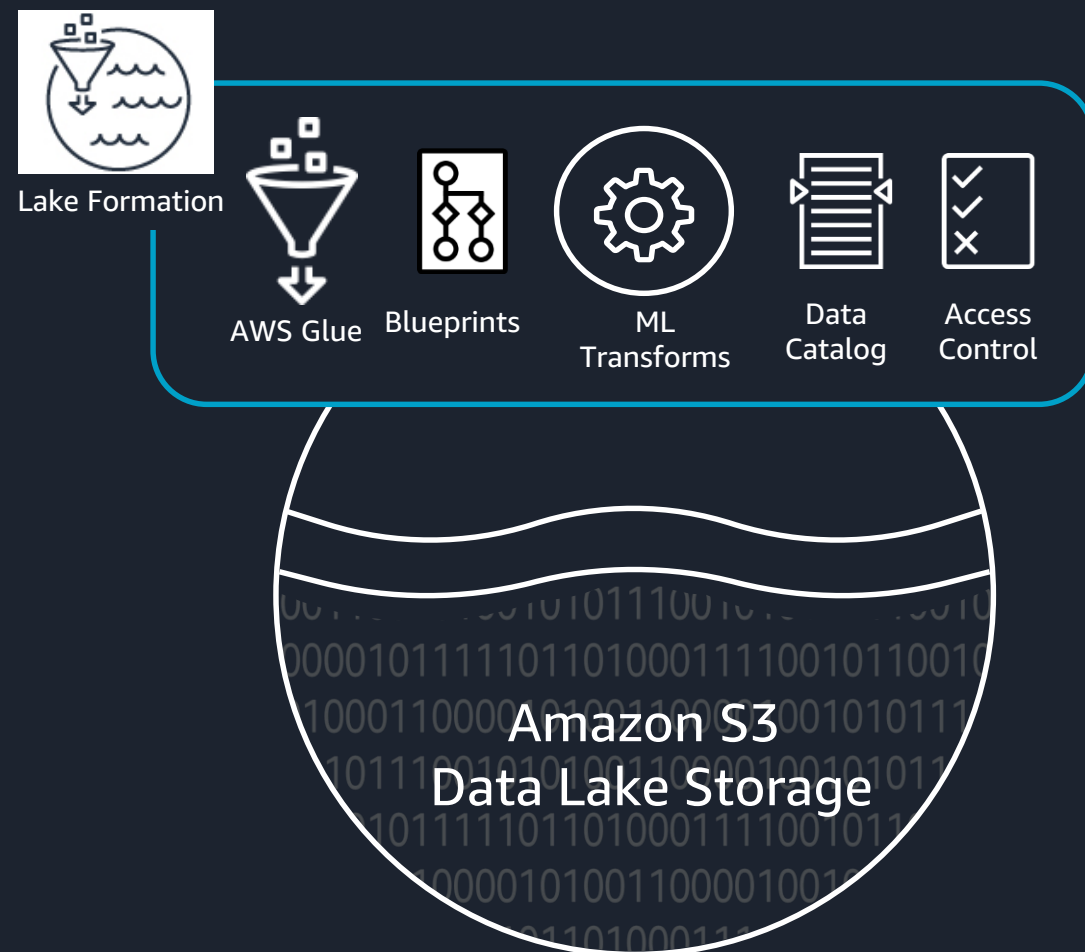
AWS Lake Formation solution stack



Simplified **ingest & cleaning** enables data engineers to build faster

Cost-effective, durable storage with global replication capabilities

AWS Lake Formation solution stack



Centralized management of **fine-grained permissions** empower security officers

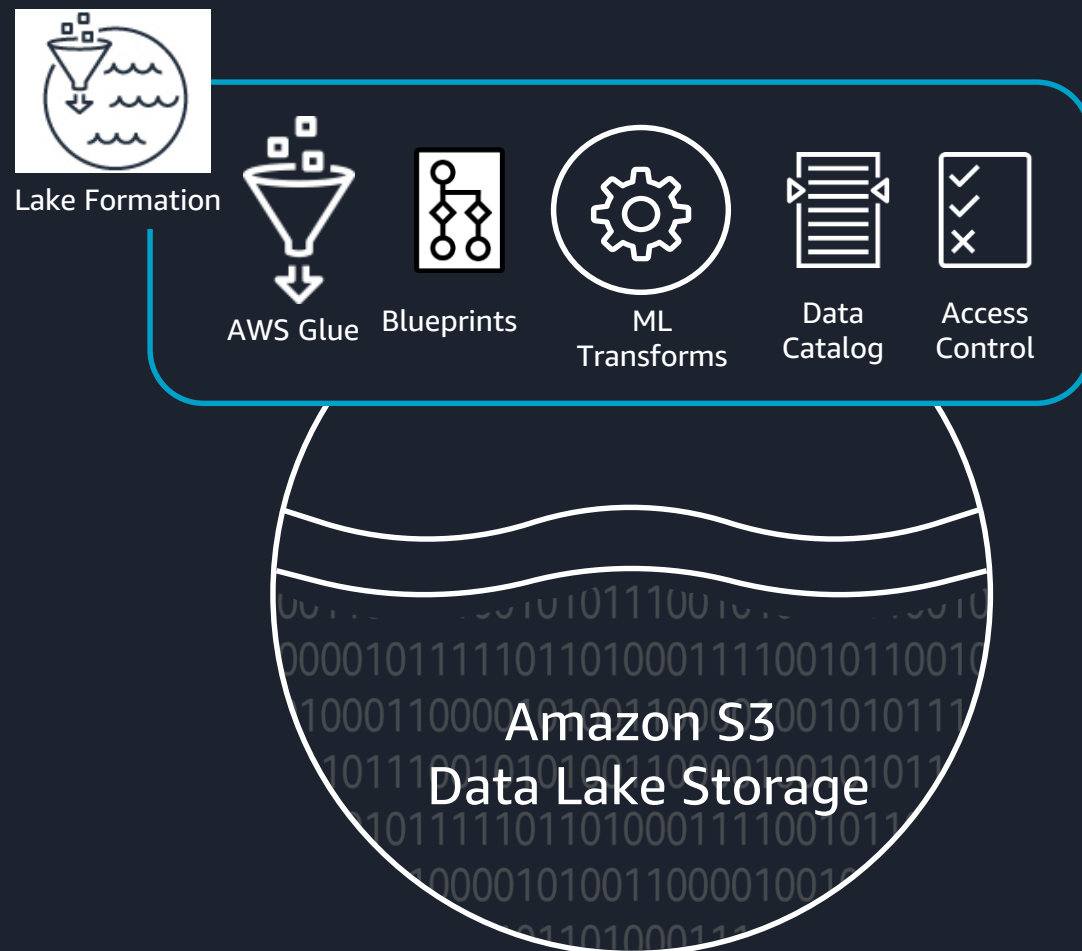
Simplified **ingest & cleaning** enables data engineers to build faster

Cost-effective, durable storage with global replication capabilities

AWS Lake Formation solution stack



Discovery, sharing, and integrated tools to enable every user



Centralized management of fine-grained permissions empower security officers

Simplified ingest & cleaning enables data engineers to build faster

Cost-effective, durable storage with global replication capabilities

AWS Lake Formation builds on AWS Glue

AWS Lake Formation

Monitoring

Blueprints

Security, search,
collaboration

Workflow

Glue Data Catalog

Glue ETL Jobs

Glue Crawlers

Connections,
Databases, Tables

AWS Glue

AWS Glue provides scalable **serverless** components



Data catalog

Apache Hive Metastore
compatible

Integrated with AWS
analytic services

AWS Glue provides scalable **serverless** components



Data catalog

Apache Hive Metastore compatible

Integrated with AWS analytic services



Crawlers

Automatically infer schemas

Populate data catalog

AWS Glue provides scalable **serverless** components



Data catalog

Apache Hive Metastore compatible

Integrated with AWS analytic services



Crawlers

Automatically infer schemas

Populate data catalog



Serverless ETL

Interactive development

Apache Spark / Python shell jobs

Serverless execution

AWS Glue provides scalable **serverless** components



Data catalog

Apache Hive Metastore compatible

Integrated with AWS analytic services



Crawlers

Automatically infer schemas

Populate data catalog

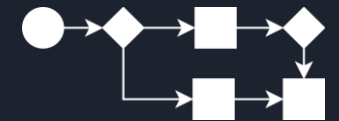


Serverless ETL

Interactive development

Apache Spark / Python shell jobs

Serverless execution



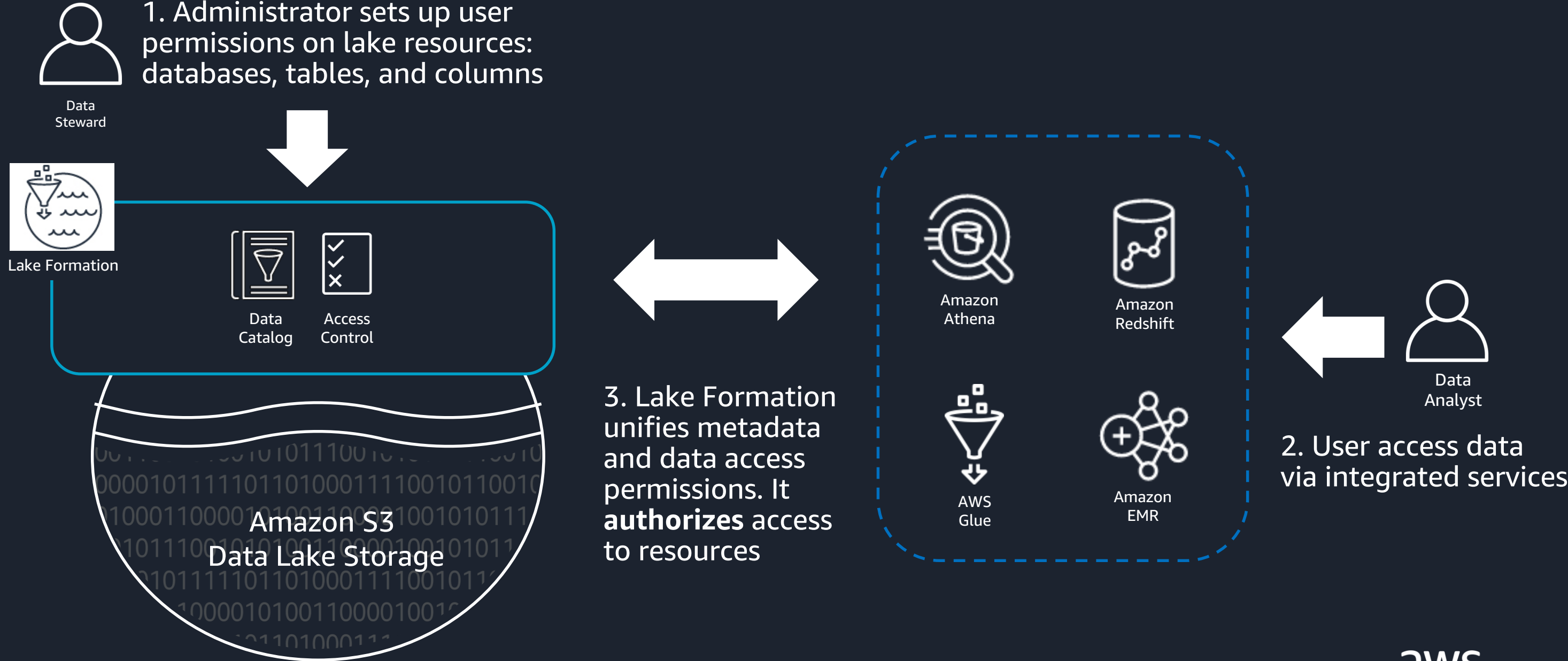
Flexible workflows

Orchestrate triggers, crawlers & jobs

Author & monitor entire flows

Integrated alerting

Centralized permissions



Security permissions in AWS Lake Formation

Control data access with simple grant and revoke permissions

Specify permissions on **DBs, 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

Security permissions in AWS Lake Formation

Control data access with simple grant and revoke permissions

Specify permissions on **DBs, 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

Security permissions in AWS Lake Formation


Control data access with simple grant and revoke permissions

Specify permissions on **DBs, 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

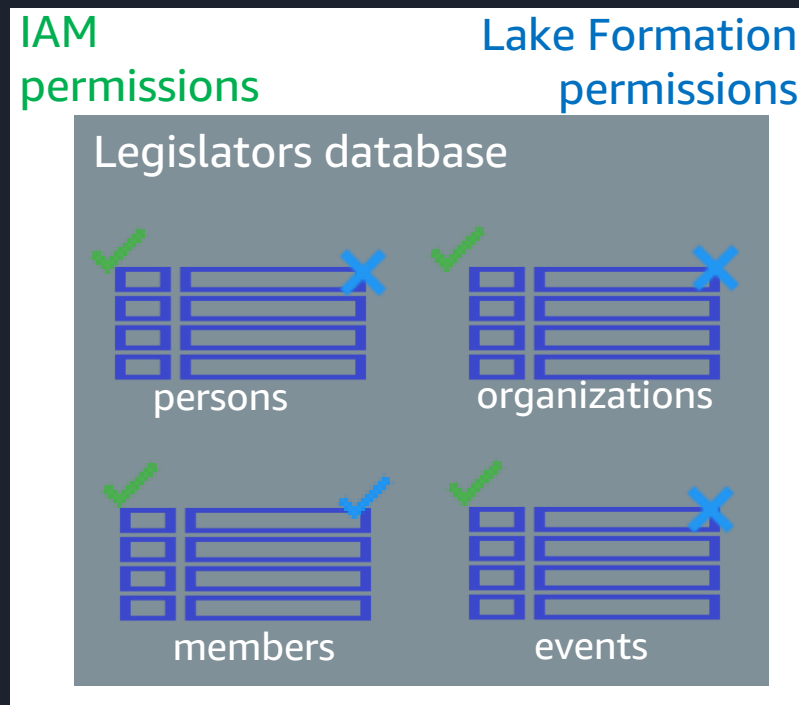


User 2

AWS Lake Formation security model

Works **in conjunction** with IAM

New permissions



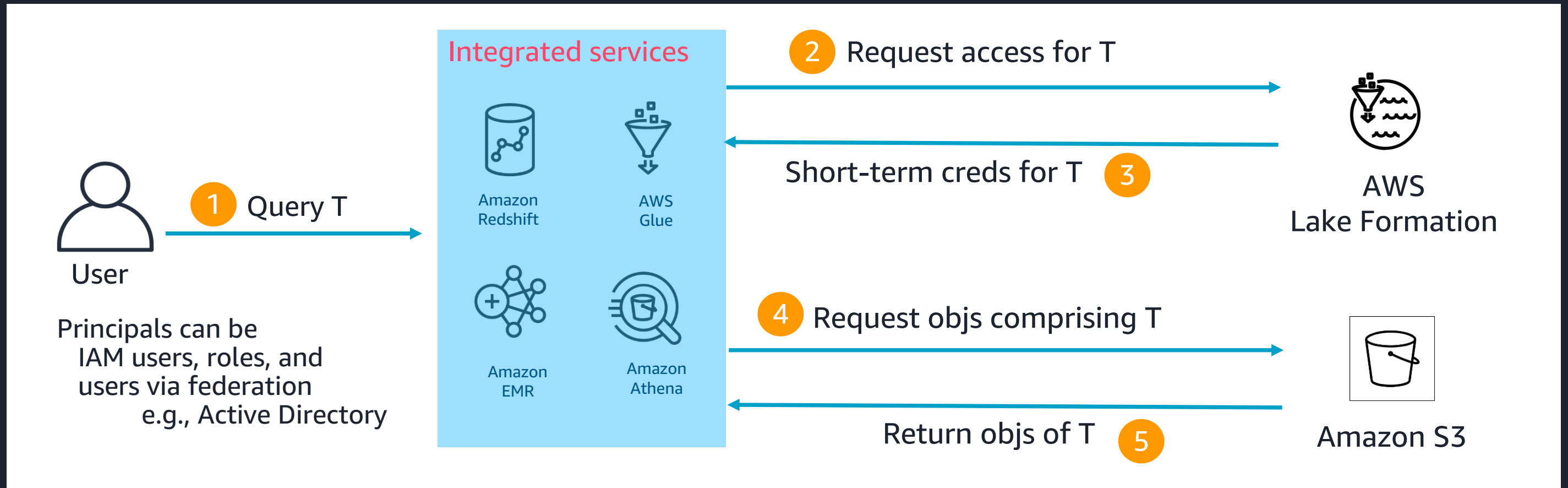
Temp credential vending



AWS Lake Formation security – request flow

AWS Lake Formation manages access to **registered locations**

No intermediary in data path

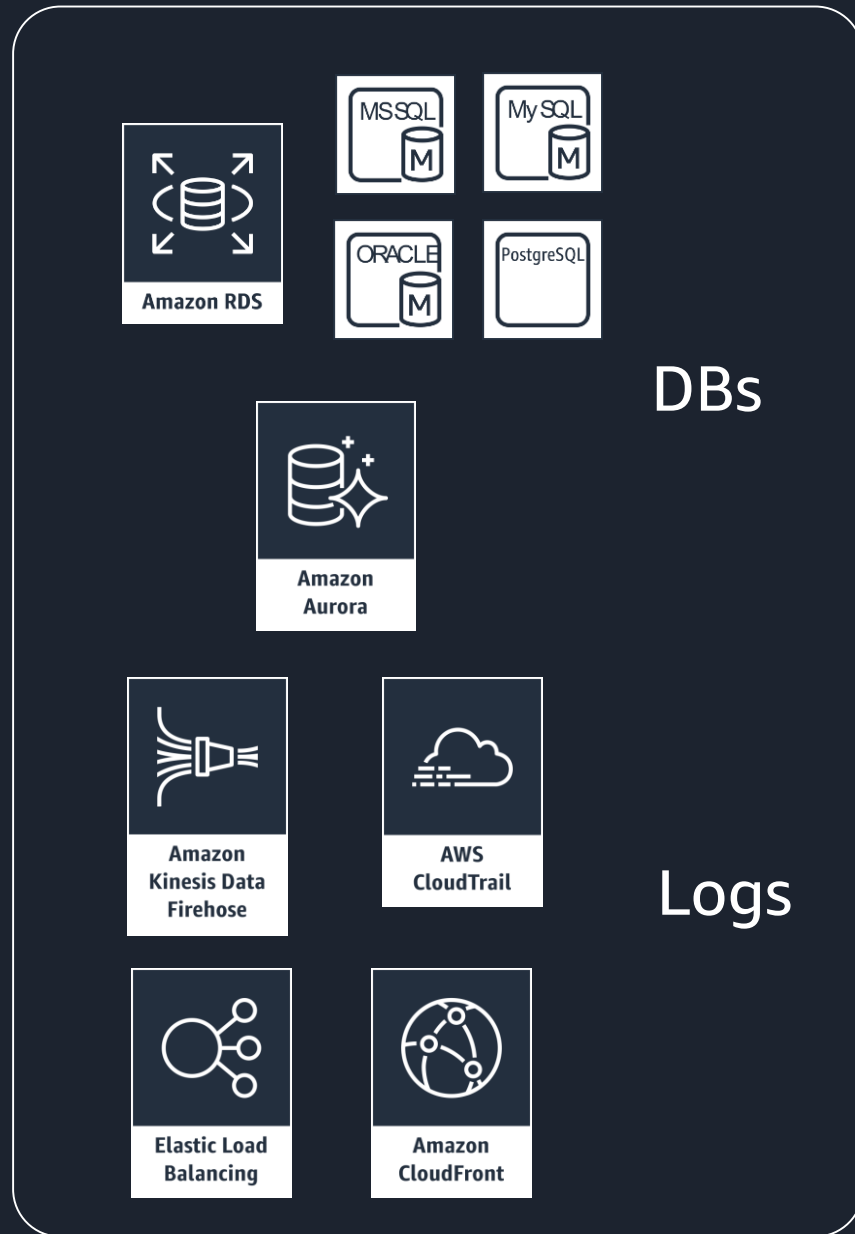


Easily load data into your data lake with blueprints

Prebuilt **templates** for common ingestion use cases

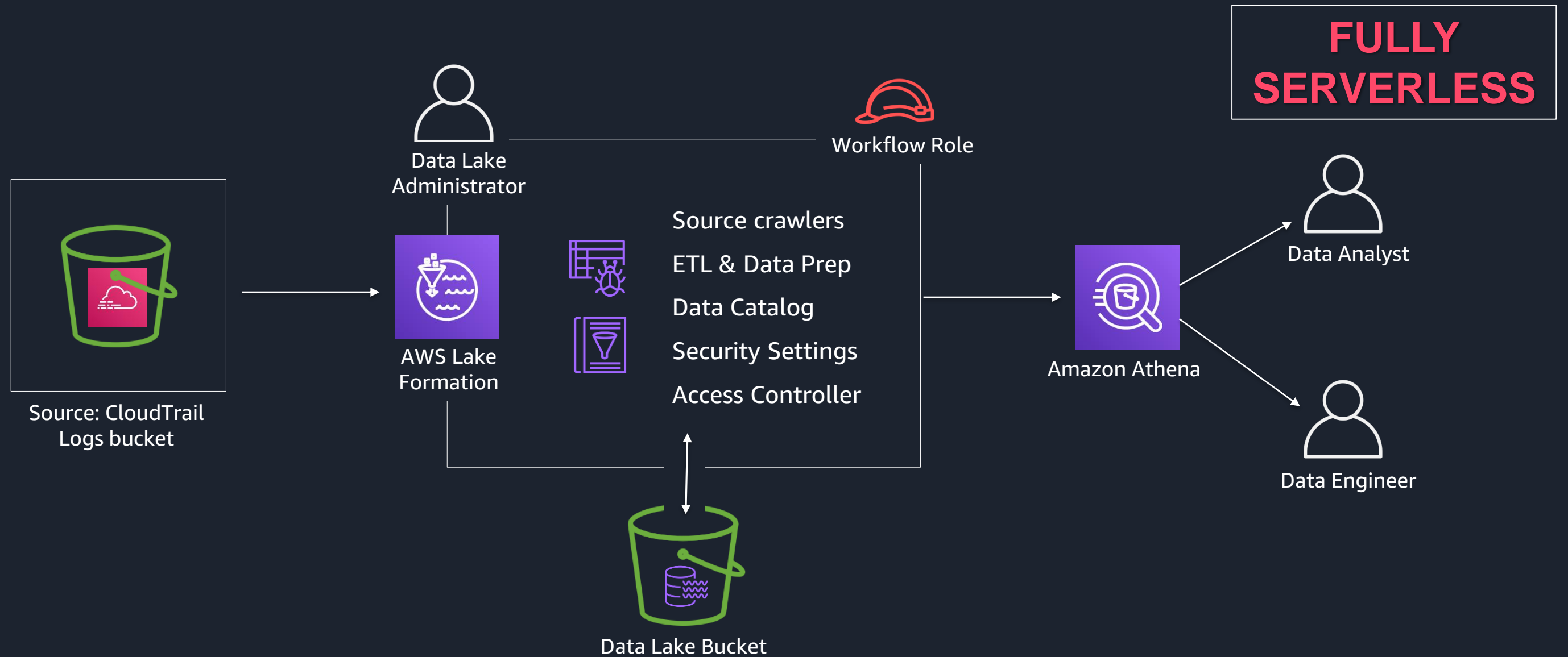
Generates **AWS Glue Workflows**
configures triggers, crawlers , jobs, data formats
generates code and sets up partitions
populates Data Catalog
snapshot or continuous

Customize for your needs



DEMO

Creating a Data Lake from an AWS CloudTrail Source



Creating a Data Lake from an AWS CloudTrail Source

Demo Step 1: Register S3 path, data lake location

Demo Step 2 & 3: Load data with blueprint

Demo step 4: Grant permissions to users

Demo Step 5: Run query in Amazon Athena

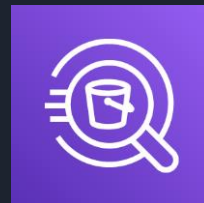
Conclusion

Amazon S3 is the center of your data lake on AWS

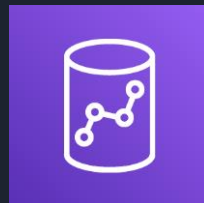
Data lakes are the **evolution of warehousing**

AWS Lake Formation makes setting up, securing, and using data lakes simple

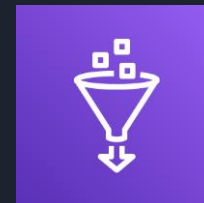
Integrated services:



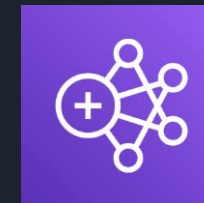
Amazon
Athena



Amazon
Redshift



AWS
Glue



Amazon EMR
(Public Beta)

Try this out...

Overview of a Data Lake on AWS

<https://aws-reference-architectures.gitbook.io/datalake/>

AWS Lake Formation Workshop

<https://lakeformation.aworkshop.io/>

Tutorial: Creating a Data Lake from an AWS CloudTrail Source

<https://docs.aws.amazon.com/lake-formation/latest/dg/getting-started-cloudtrail-tutorial.html>

AWS Training and Certification



Training for the Whole Team

Explore tailored Data or Database learning paths for customers and partners



Flexibility to Learn Your Way

Build cloud skills with free digital Data training courses such as "The elements of Data Science", or dive deep with classroom training



Validate Skills with AWS Certification

Demonstrate expertise with a Data industry-recognized credential (Data analytics and Database Specialty AWS Certifications)

aws.amazon.com/training/

Visit the Data, Databases, and Analytics Resource Hub for more resources

Dive deeper with these newly created whitepapers and e-books to help you uncover new insights and value from your data

- An introduction to cloud databases
- Enter the purpose-built database era
- Harness the power of data
- Creating a modern analytics architecture
- The data-driven enterprise
- ... and more!









[https://tinyurl.com/
aws-data-databases-analytics](https://tinyurl.com/aws-data-databases-analytics)

[Visit resource hub »](#)

Thank you for attending AWS Data, Databases, and Analytics Online Series

We hope you found it interesting! A kind reminder to **complete the survey**.
Let us know what you thought of today's event and how we can improve the event
experience for you in the future.

-  aws-apac-marketing@amazon.com
-  twitter.com/AWSCloud
-  facebook.com/AmazonWebServices
-  youtube.com/user/AmazonWebServices
-  slideshare.net/AmazonWebServices
-  twitch.tv/aws