



Build a Secure Data Lake without Sacrificing Accessibility using AWS Lake Formation

Raghu Prabhu, Principal Lake House Specialist
2021

What problems are we solving today?

Secure
Data

Share
Data

Scale
Data

Lake House architecture on AWS



SCALABLE DATA LAKES

PURPOSE-BUILT DATA SERVICES

AUTOMATED DATA MOVEMENT

CENTRAL GOVERNANCE

PERFORMANT AND COST-EFFECTIVE



Typical Data Lake Architecture

Your data

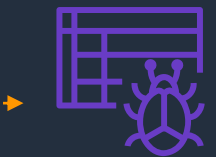
- Web app data (LOG icon)
- Amazon RDS (Database icon)
- Other databases (Database icon)
- On premises data (Grid icon)
- Streaming data (Streaming icon)

Challenges #3: Help build easy data ingestion pipelines



Amazon S3

Challenges #2: Manage S3 permissions for Analytics

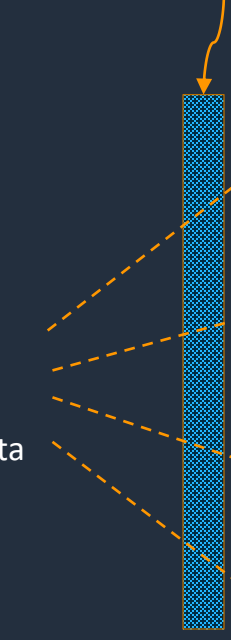


AWS Glue Crawler



AWS Glue Data Catalog

AWS Lake Formation



Challenges #1: Security and governance layer

Challenges #4: Share databases and tables across AWS Accounts

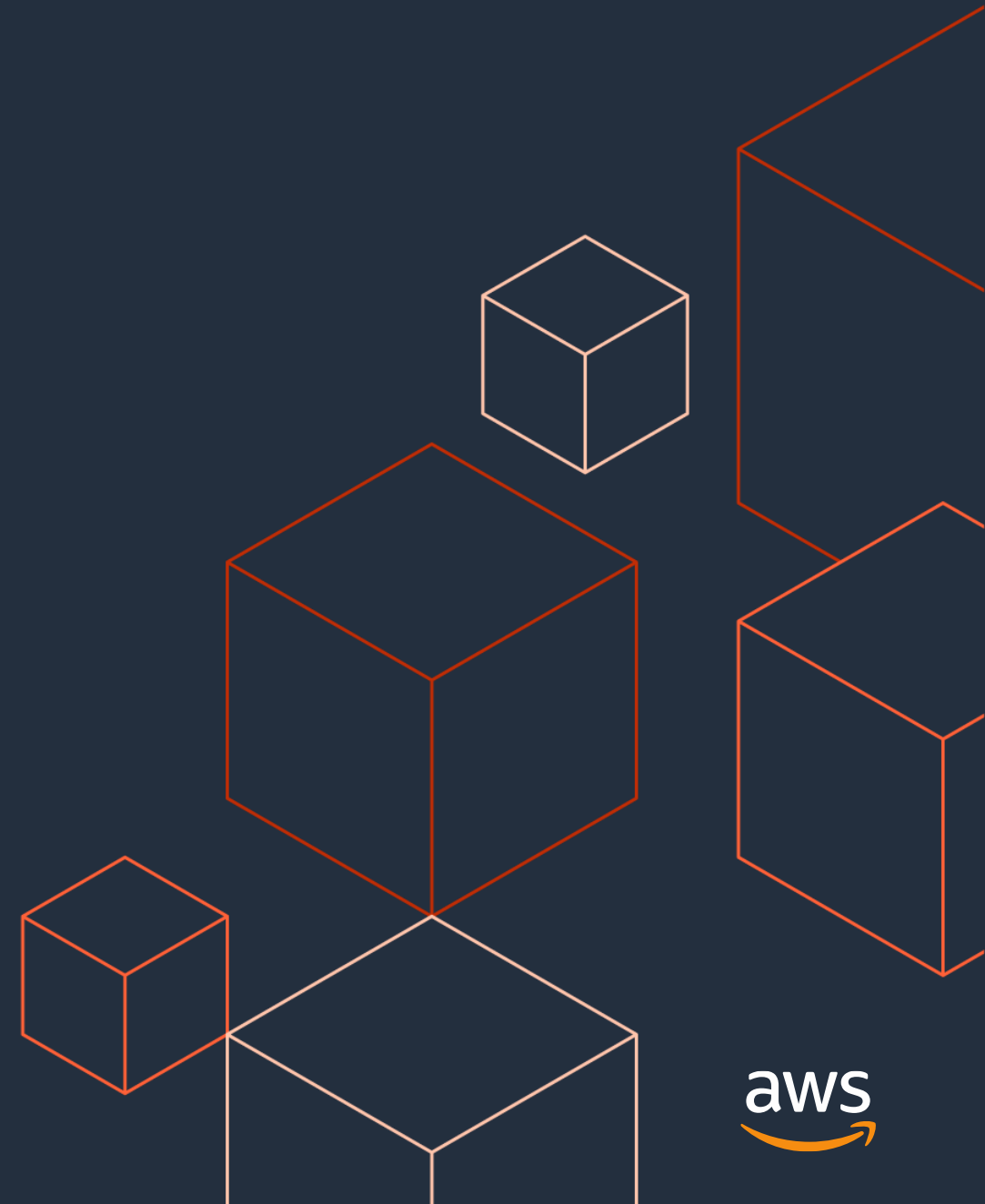
Challenges #5: Complete Data Management solutions

Analytics and ML

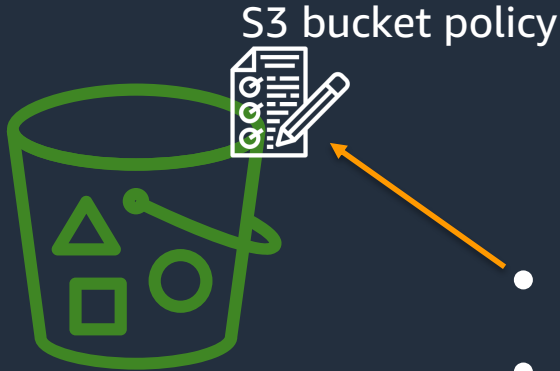
- AWS Glue ETL
- Amazon Athena
- Amazon EMR
- Amazon Redshift Spectrum
- Amazon QuickSight
- Amazon SageMaker



Secure Data

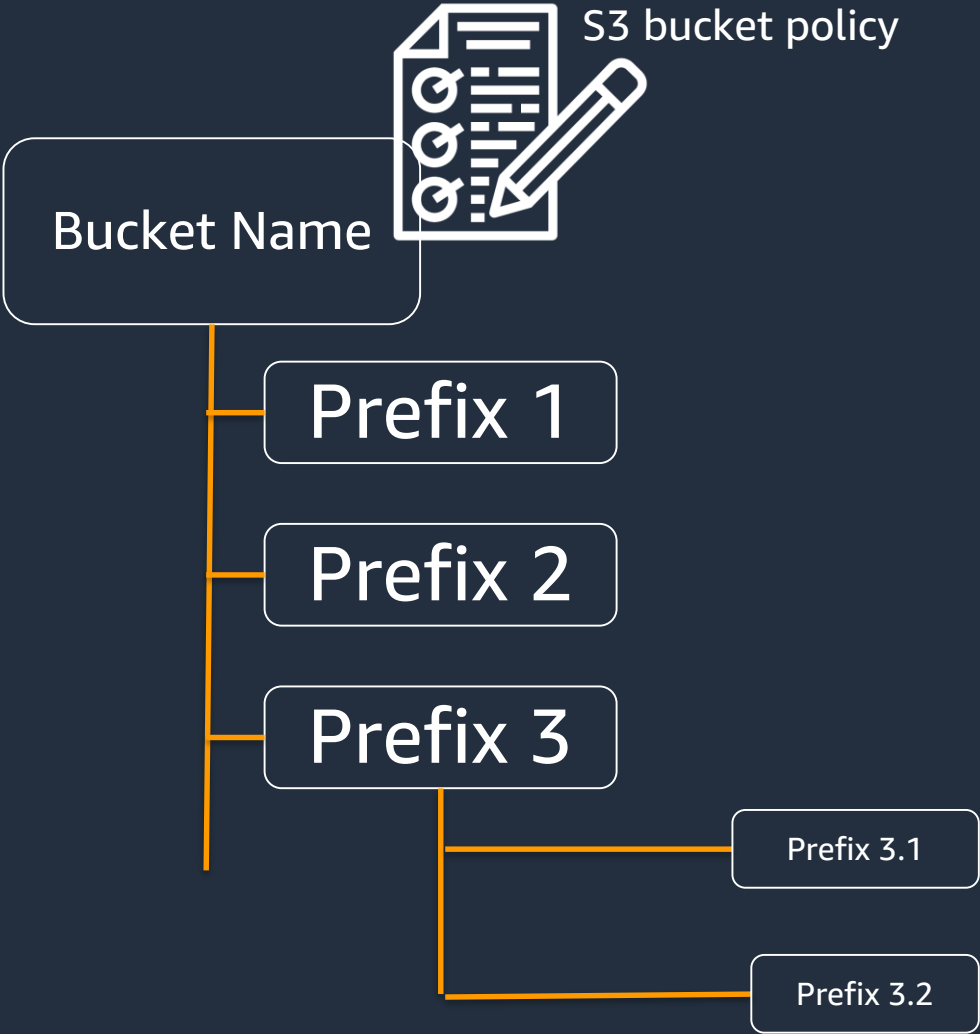


Simplified Amazon S3 policy management



- Written in JSON
- Very flexible
- No one enjoys writing and maintaining these policies

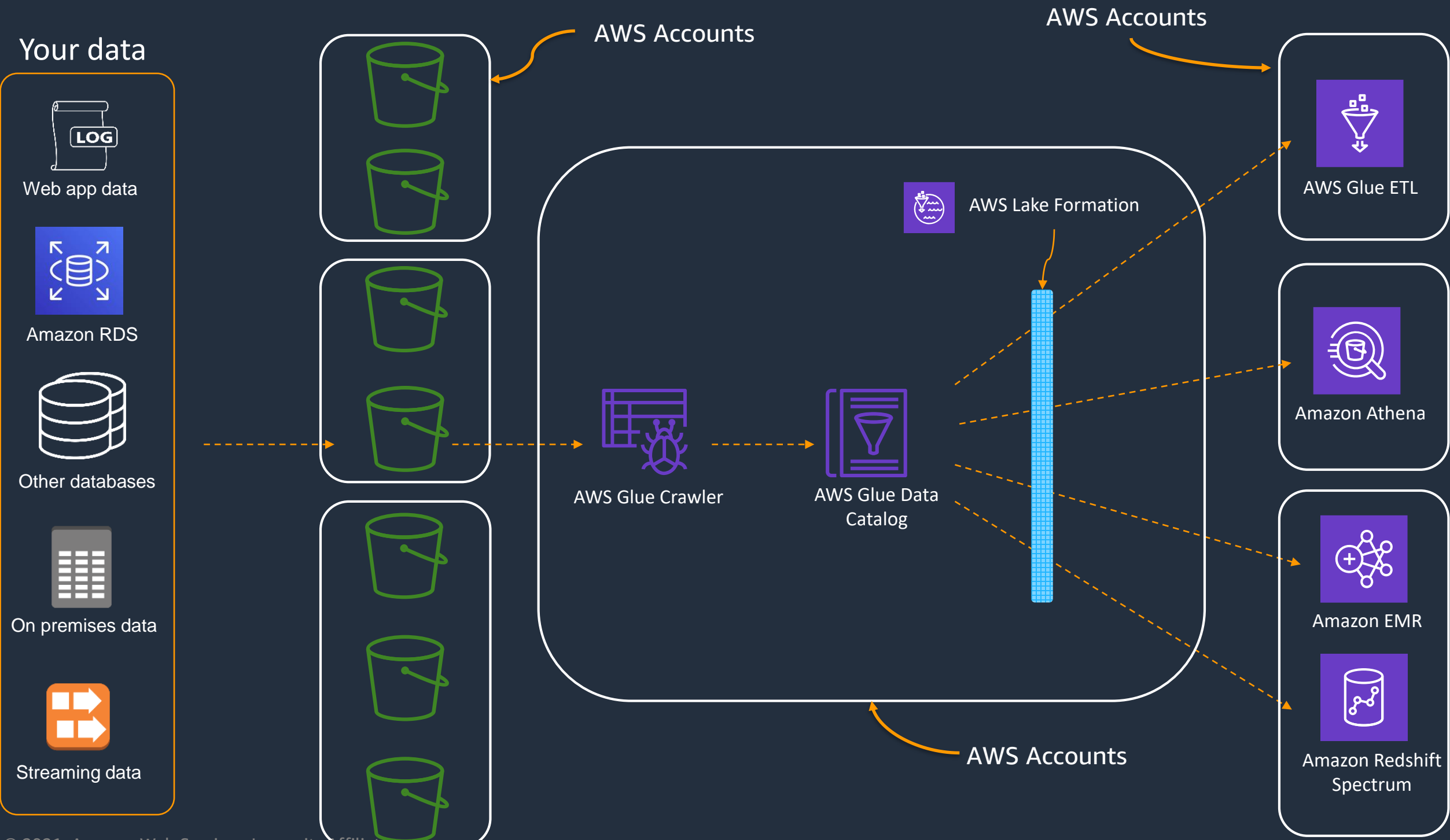
Simplified Amazon S3 policy management



```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"AddCannedAcl",
      "Effect":"Allow",
      "Principal":{"AWS":
["arn:aws:iam::111122223333:root","arn:aws:iam::444455556666:root"]
},
      "Action":["s3:PutObject","s3:PutObjectAcl"],
      "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
      "Condition":{"StringEquals":{"s3:x-amz-acl":["public-read"]}}
    }
  ]
}
```

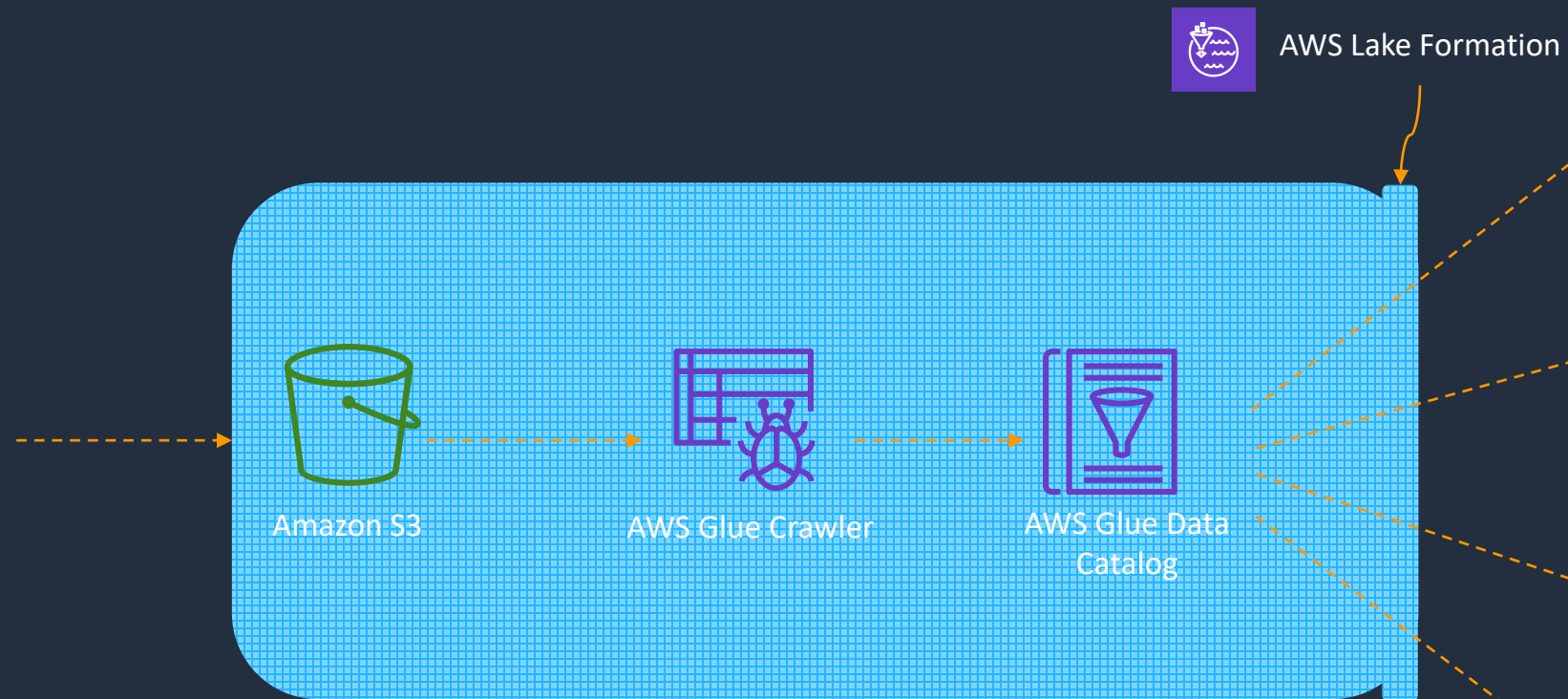
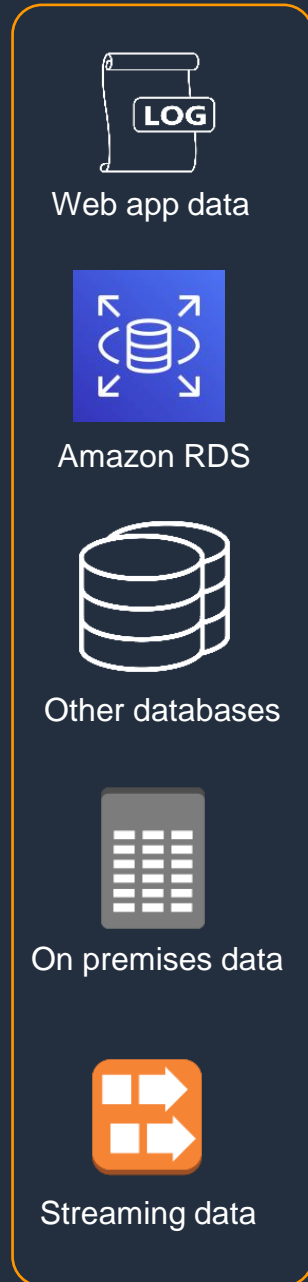


Simplified Amazon S3 policy management

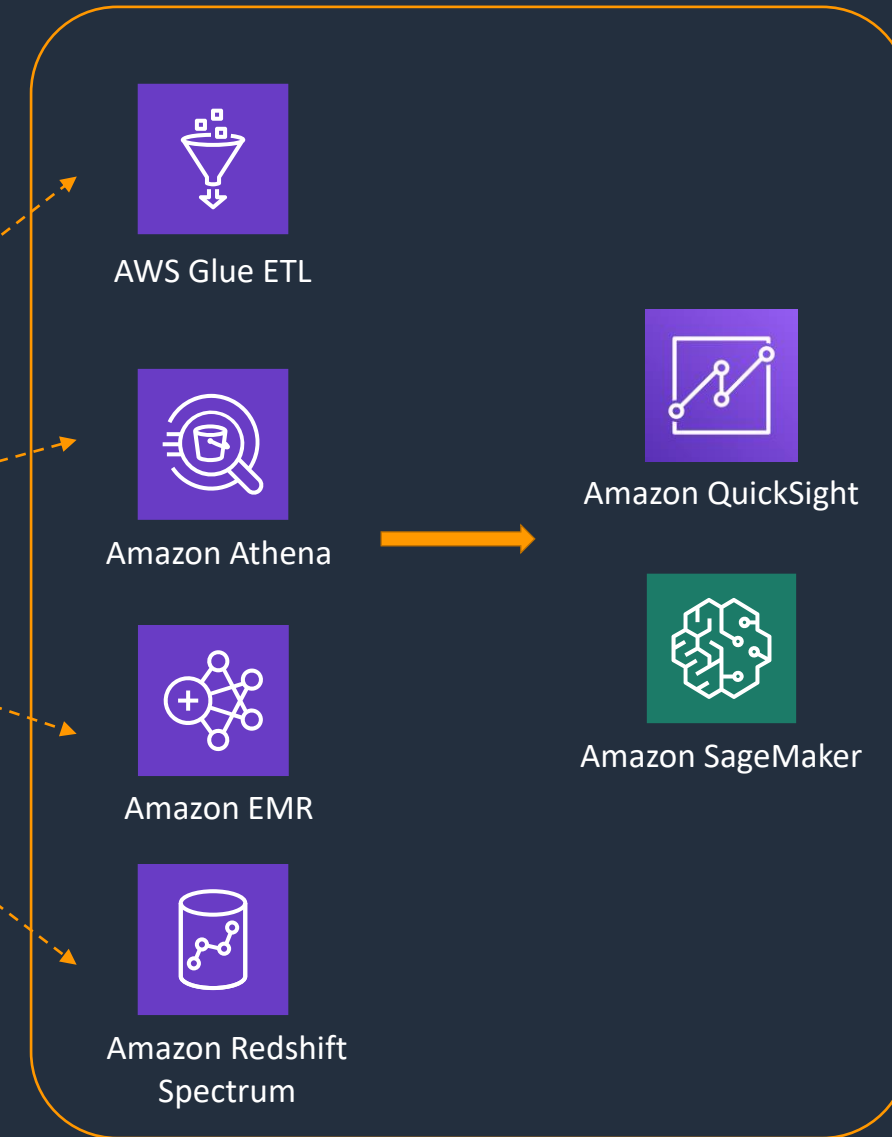


Simplified Amazon S3 policy management

Your data



Analytics and ML



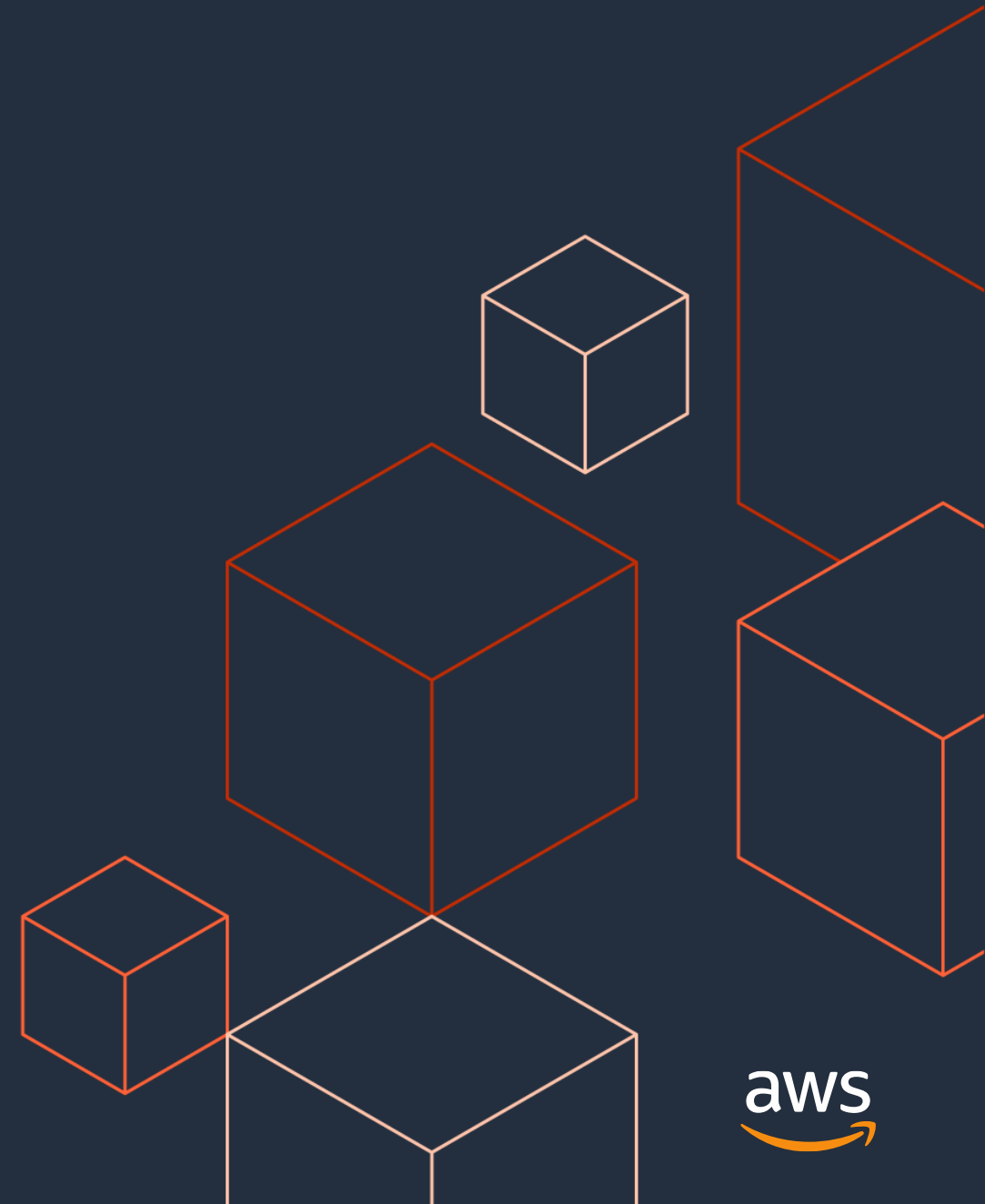
Simplified Amazon S3 policy management

The screenshot shows the AWS Lake Formation console interface. On the left is a navigation sidebar with the following items: Dashboard, Data catalog (expanded), Databases, Tables, Settings, Register and ingest (expanded), Data lake locations (highlighted with a red arrow), Blueprints, Crawlers, Jobs, and Permissions. The main content area is titled 'Data lake locations' and contains a search bar with the placeholder text 'Find data lake storage'. Below the search bar is a table with two columns: 'Amazon S3 path' and 'IAM role'. The table lists five data lake locations, each with a radio button, a path, and an IAM role. Red arrows point to the first row's path and role cells. The IAM role for all entries is 'AWSServiceRoleForLakeFormationDataAccess'.

	Amazon S3 path	IAM role
<input type="radio"/>	s3://raghuoutput/example4	AWSServiceRoleForLakeFormationDataAccess
<input type="radio"/>	s3://raghuoutput/example2	AWSServiceRoleForLakeFormationDataAccess
<input type="radio"/>	s3://raghuoutput/example	AWSServiceRoleForLakeFormationDataAccess
<input type="radio"/>	s3://raghuoutput/amazonreviewscsv	AWSServiceRoleForLakeFormationDataAccess
<input type="radio"/>	s3://amazonreviews201	AWSServiceRoleForLakeFormationDataAccess



Share Data



Facts about data

Having a data lake is just the **beginning** – Data needs to be queried, reports need to be built and distributed, machine learning models need to be developed and deployed, only then data is **monetized**. A data lake that doesn't get used has no value and serves no purpose.

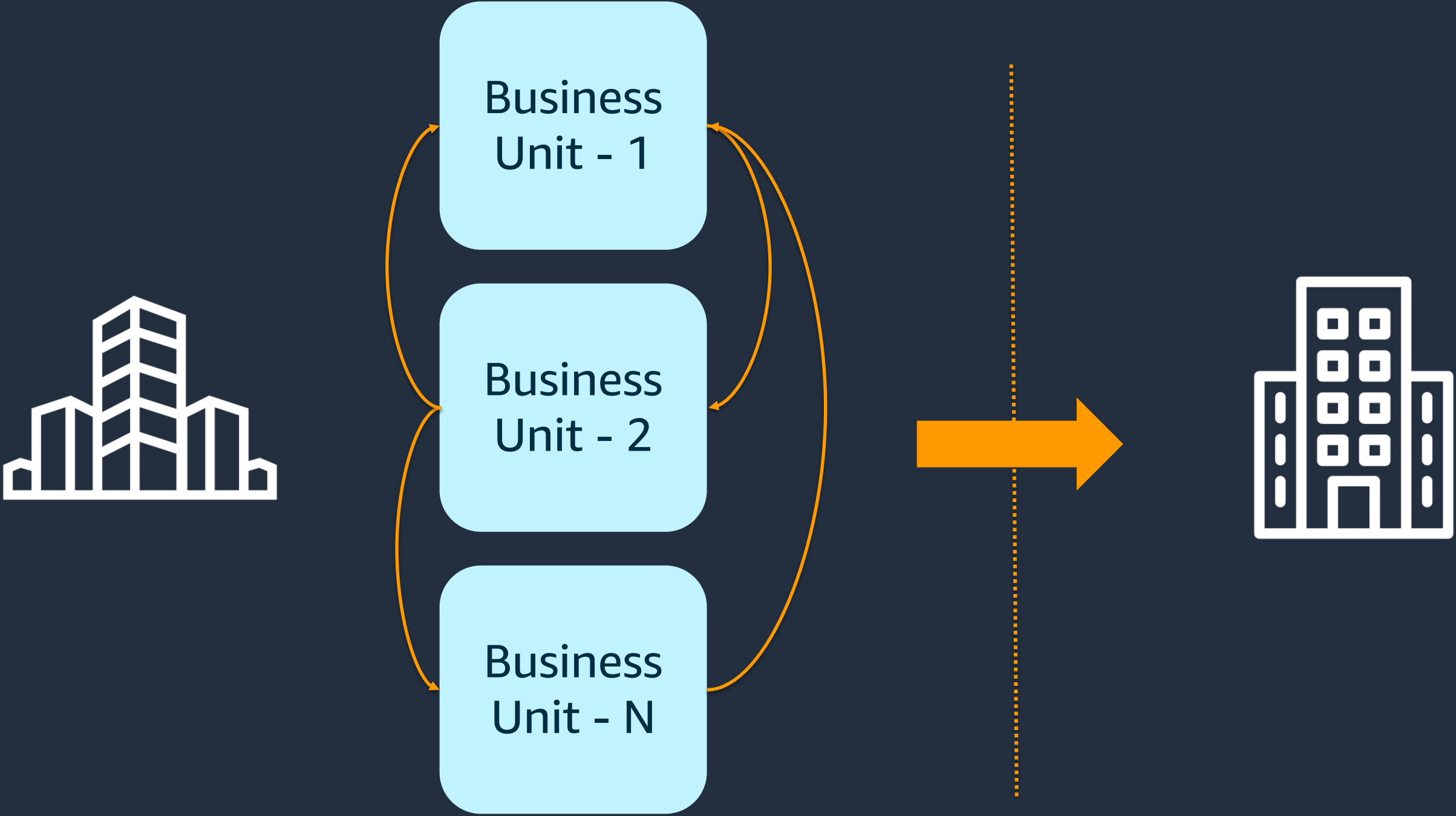
Easier we make it to **secure** the data, **share** the data and **access** the data, better will be the utilization of your data lake.

If data has to be **copied** over and over again as a sharing mechanism, it simply doesn't **scale** and there are too many **trust barriers** to overcome, increasing adoption friction.

Key tenets for data sharing

- Data should **never** have to be copied for data sharing
- It should be possible to **share as much data** (a whole database) or **as little data** (part of a table) with a consumer
- Once shared, the **sharer** should be out of the picture and the **receiver** should be able to make decisions on who will be able to access this shared data.
- The **receiver** of the data should be able to make **independent decisions** on how they will consume the shared data and the **sharer** cannot impose any technology restrictions on the **receiver**
- The **receiver** will pay for all compute to query the shared data
- **Centralized** access logs will need to be provided
- The **sharer** can decide to **stop sharing** the resource at any time and can take action to **withdraw** the share

Sharing your data is easy with AWS Lake Formation





Goal - Share databases and tables across AWS Accounts

 AWS Organizations ✓

✓  AWS Org. Unit (OU 1)

 AWS Org. Unit (OU 2) ✓

 AWS Account - A


AWS Lake Formation

Databases
Tables

 AWS Account - B ✓

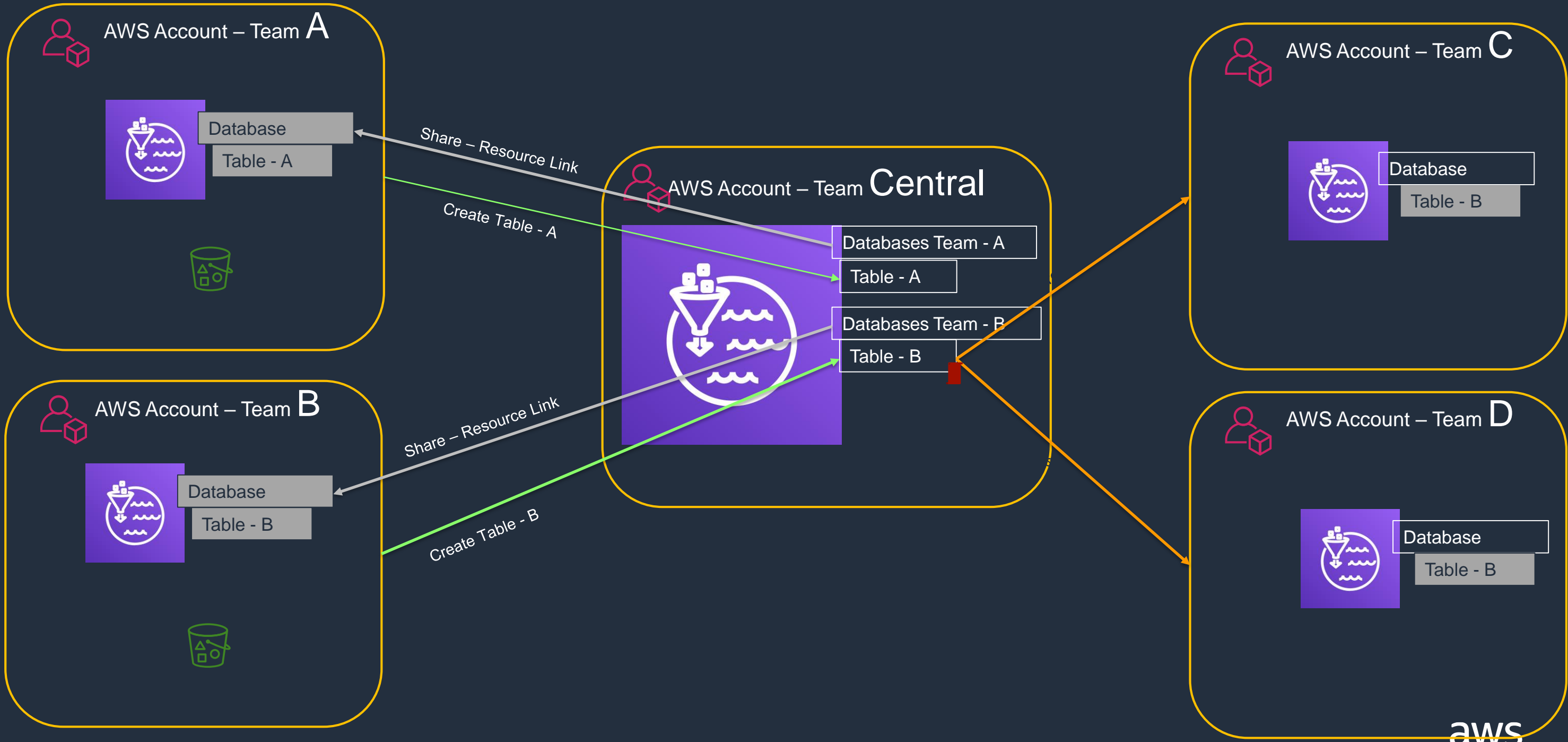
 AWS Account - C ✓

 AWS Account - External ✓

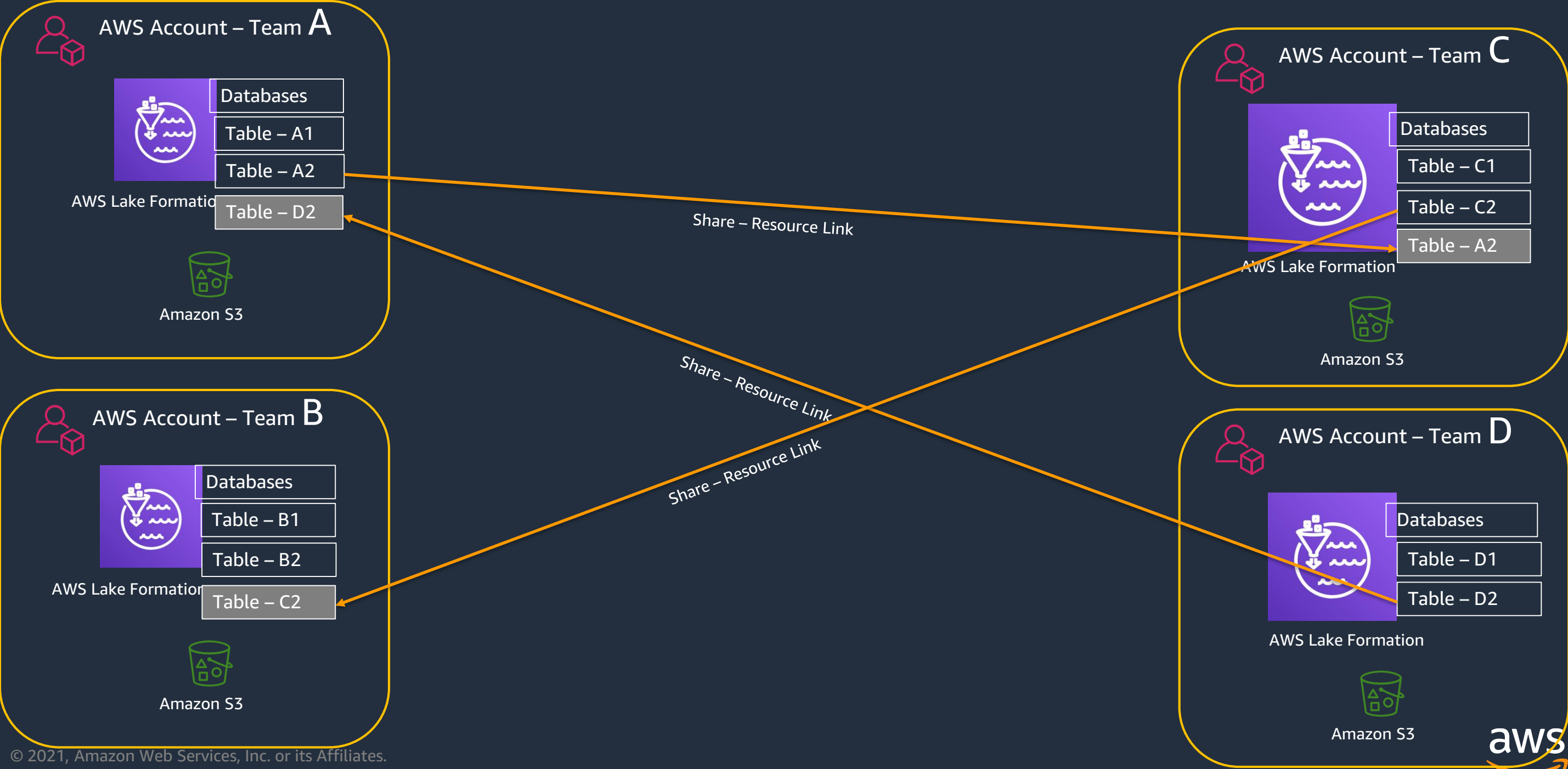
What can you share?

- Entire Databases
- Tables
- Specific columns within tables

Create sharable metadata table directly in the central data catalog



Peer to peer data catalog Lake Formation architecture



Value of AWS Lake Formation data sharing features

- **One schema** – share everywhere architecture
- **Share** AWS Lake Formation databases and tables to **any AWS accounts**
- No Amazon **S3 policy** edits required
- Receiver of the data can spin-up their **choice** of analytics service for which they can **pay** for separately
- The receiver of the shared resource can manage security of the shared resource **independent** of the sharer
- **Centralized** access logs

You can do all this with no special edits to S3 policies

How does it work?

Grant permissions: amazonreviews

Choose the access permissions to grant.

My account
User or role from this AWS account.

External account
AWS account or AWS organization outside of my account.

AWS account ID or AWS organization ID

Enter AWS account ID or AWS organization ID

AWS accounts

- Account 2
AWS account
- Raghu Prabhu AWS 4
AWS account
- Raghu Prabhu AWS 3
AWS account

Organizations

- fsy
AWS Organization
- vj - GlueLFXacctTest
AWS organizational unit
- vj - GlueLFXacctTest
- Raghavendra Prabhu
AWS account
- Glue data catalog upgrade
AWS account

Grant permissions: amazonreviews

Choose the access permissions to grant.

My account
User or role from this AWS account.

External account
AWS account or AWS organization outside of my account.

AWS account ID or AWS organization ID

Enter AWS account ID or AWS organization ID

123456789012
Account

Enter one or more AWS account IDs or AWS organization IDs. Press Enter after each ID.

Table
Add one or more tables.

Choose tables

Database permissions
Choose the specific access permissions to grant.

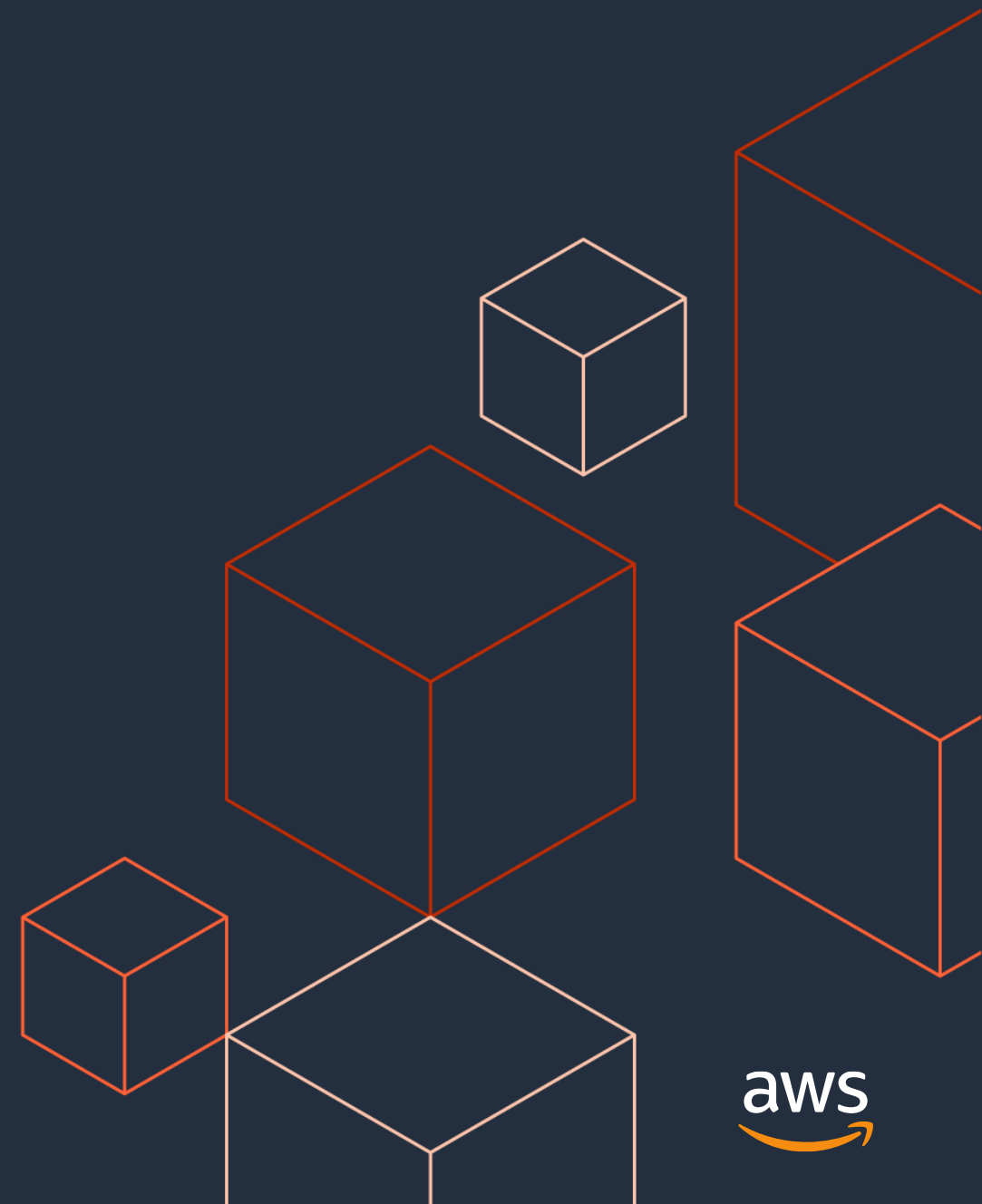
Create table Alter Describe

Grantable permissions
Choose the permissions that may be granted to others.

Create table Alter Describe

Cancel Grant

Scale



Fine grained access control

Database

Grant permissions: amazonreviews
Choose the access permissions to grant.

My account
User or role from this AWS account.

External account
AWS account or AWS organization outside of my account.

IAM users and roles
Add one or more IAM users or roles.

Choose IAM principals to add

Raghu X
User

testalfuser X
User

SAML and Amazon QuickSight users and groups
Enter a SAML user or group ARN or Amazon QuickSight ARN. Press Enter to add additional ARNs.

Ex: `arn:aws:iam::<AccountId>:saml-provider/<SamlProviderName>:user/<UserName>`

Database permissions
Choose the specific access permissions to grant.

Create table Alter Drop Describe

IAM Users

SAML Assertion

Table

Grant permissions: amazonreviews101
Choose the access permissions to grant.

My account
User or role from this AWS account.

External account
AWS account or AWS organization outside of my account.

IAM users and roles
Add one or more IAM users or roles.

Choose IAM principals to add

Raghu X
User

testalfuser X
User

SAML and Amazon QuickSight users and groups
Enter a SAML user or group ARN or Amazon QuickSight ARN. Press Enter to add additional ARNs.

Ex: `arn:aws:iam::<AccountId>:saml-provider/<SamlProviderName>:user/<UserName>`

Columns - optional
Choose filter type

Include columns

Include columns
Grant permissions to access the selected columns.

Choose one or more columns

customer_id X
string

product_title X
string

star_rating X
int

Table permissions
Choose the specific access permissions to grant.






Alter Insert Drop Delete Select Describe

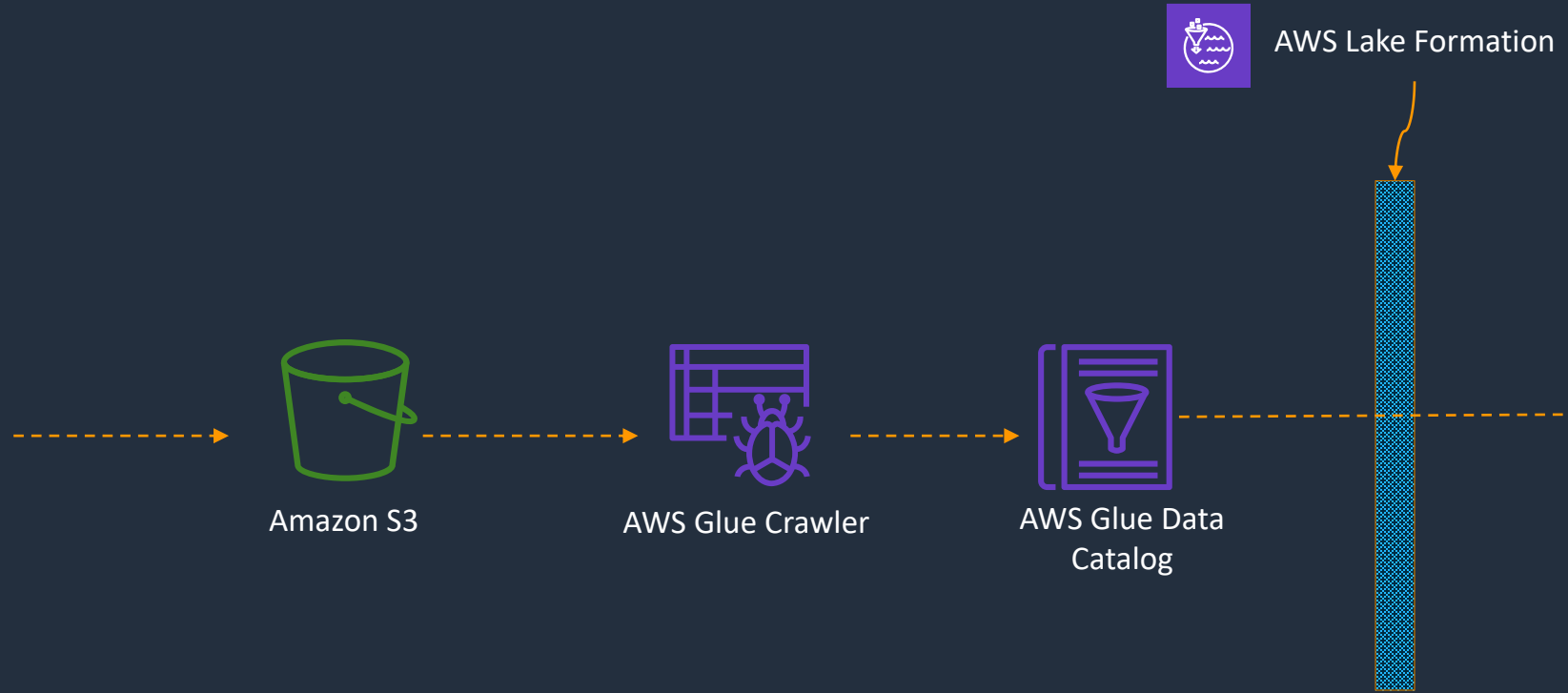
Super

Columns selected

Amazon Athena based Data Lake Architecture

Your data

-  Web app data
-  Amazon RDS
-  Other databases
-  On premises data
-  Streaming data

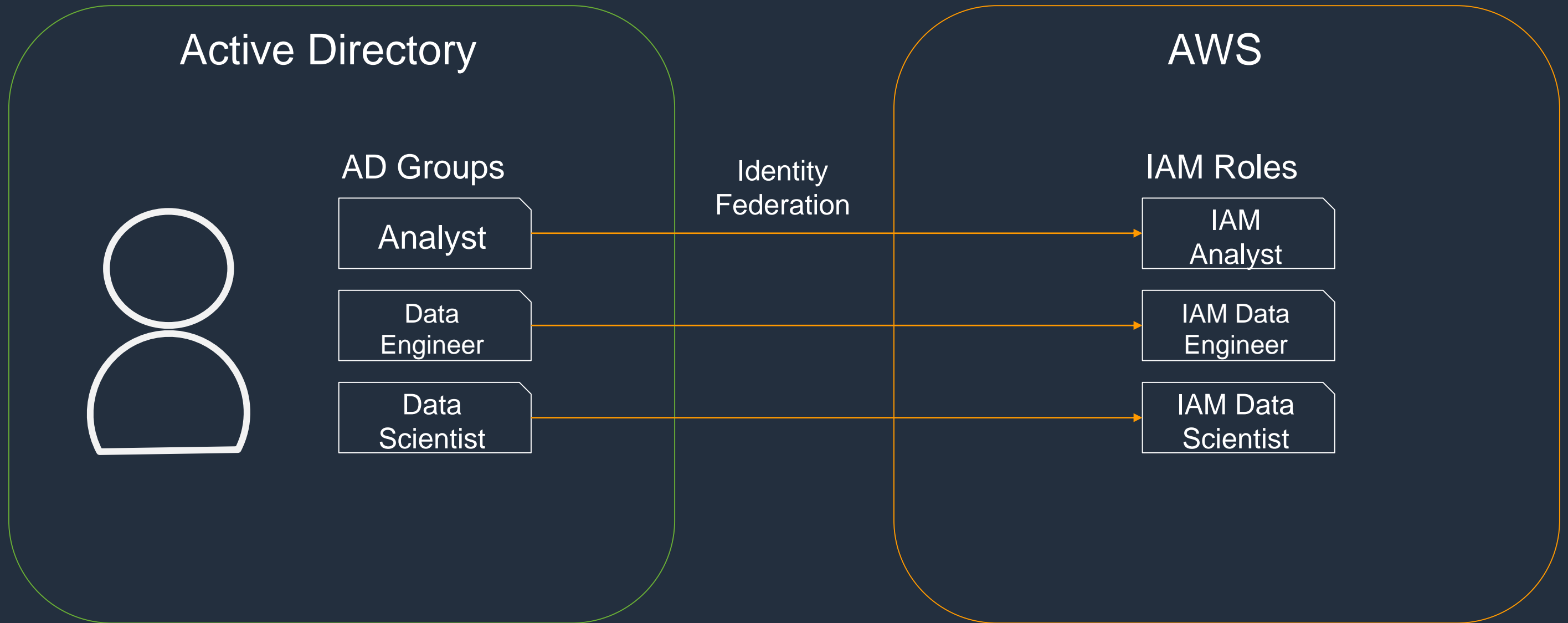


Analytics and ML

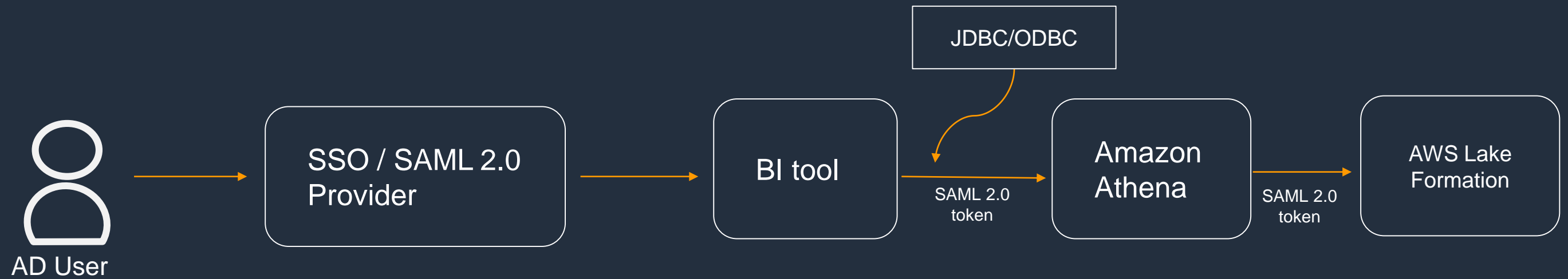
This section shows the consumption of data. Amazon Athena is used to query the data lake. The results are then processed by a BI Tool, which is accessed by users. The entire 'Analytics and ML' section is enclosed in a rounded rectangle.



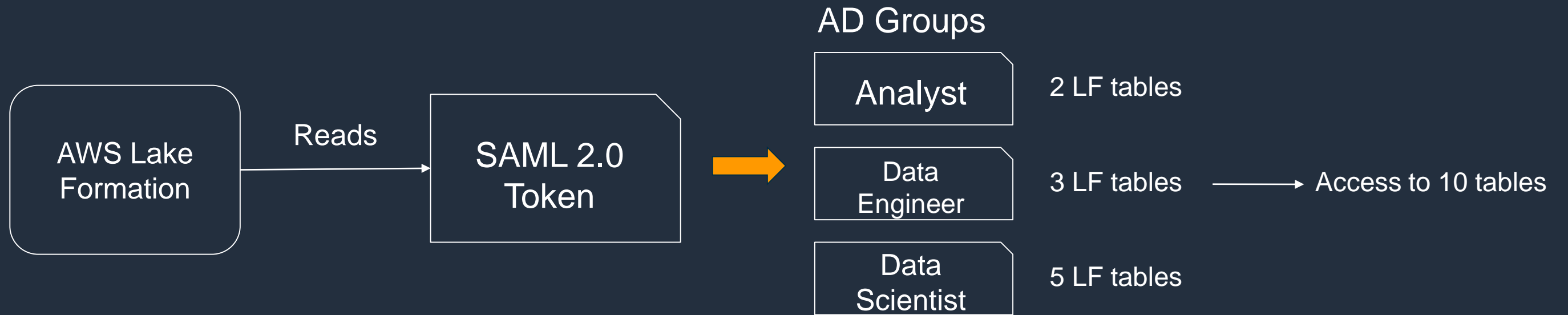
SAML 2.0 Athena + Lake Formation



SAML 2.0 based Athena + Lake Formation



SAML 2.0 based Athena + Lake Formation



Summary

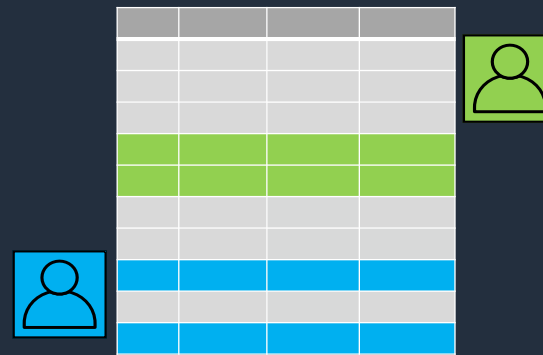
Build massively scalable Enterprise Data Platforms using AWS Lake Formation
You can secure, share and scale your Lake house architecture

New AWS Lake Formation update and access APIs (Preview open now)

Open and public: Build your own application



Atomic, consistent, isolated, and durable (ACID) transactions

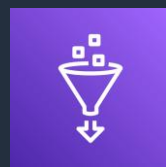


Row-level filtering for security

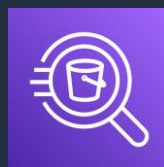


Optimizations for fast analytics on S3 data lakes

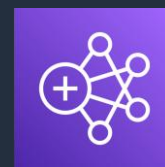
Integrations



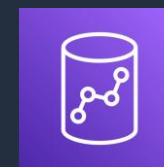
AWS Glue



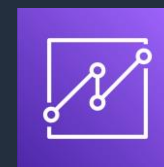
Amazon Athena



Amazon EMR



Amazon Redshift



Amazon QuickSight

Learn more about AWS Lake Formation

<https://aws.amazon.com/lake-formation/resources>

More coming soon in AWS Lake Formation Preview open now

Transaction support, Row level access control

https://pages.awscloud.com/Lake_Formation_Feature_Preview.html

Q&A

Raghu Prabhu

