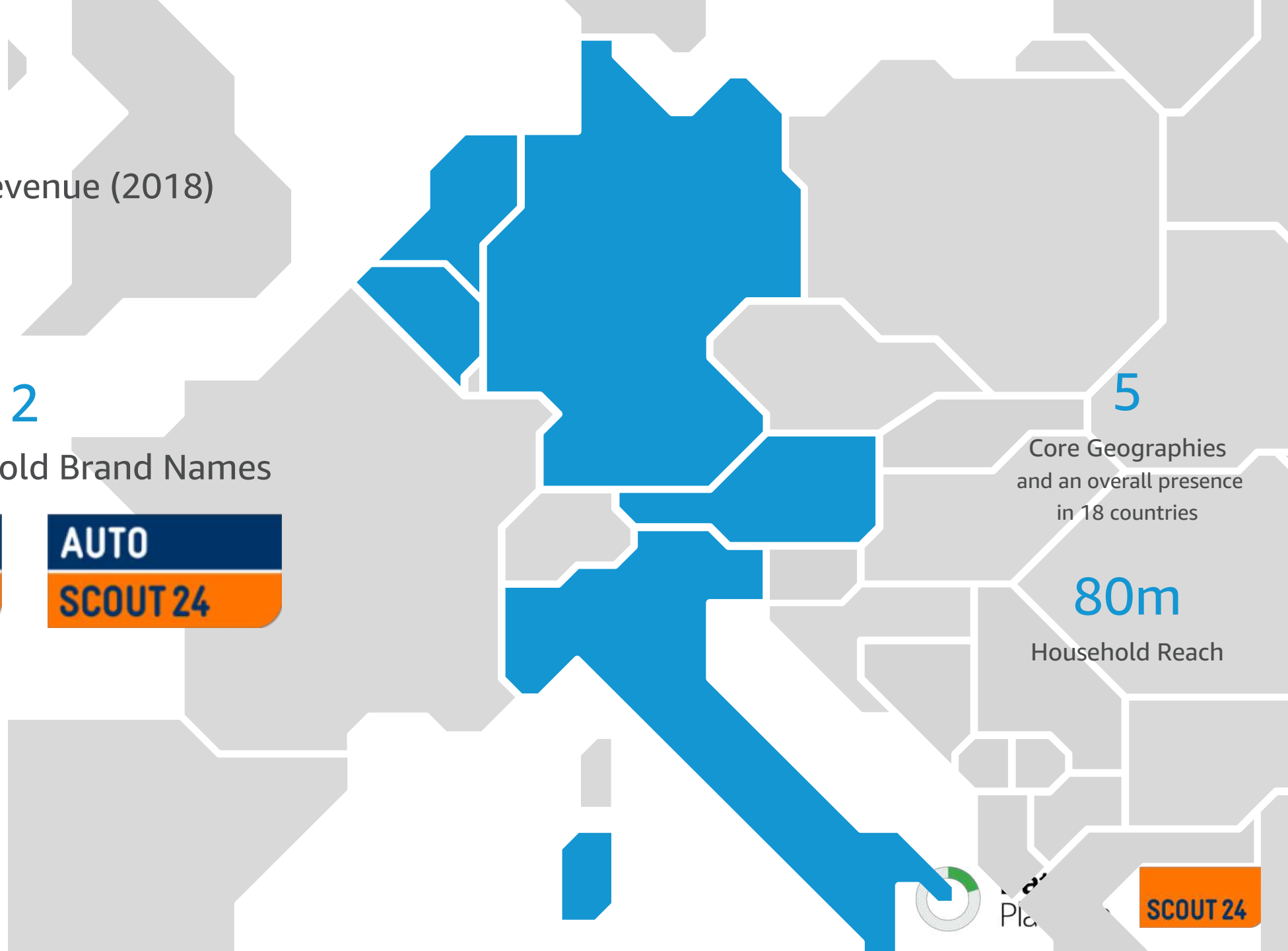
The Scout24 Data Platform
A Technical Deep Dive

www.scout24.com

Munich | March 20, 2019 |Christian Dietze, Olalekan Elesin, Raffael Dzikowski

Scout24 AG

- MDAX
- € 531.7 million revenue (2018)



2

Major Household Brand Names

IMMOBILIEN SCOUT 24

AUTO SCOUT 24

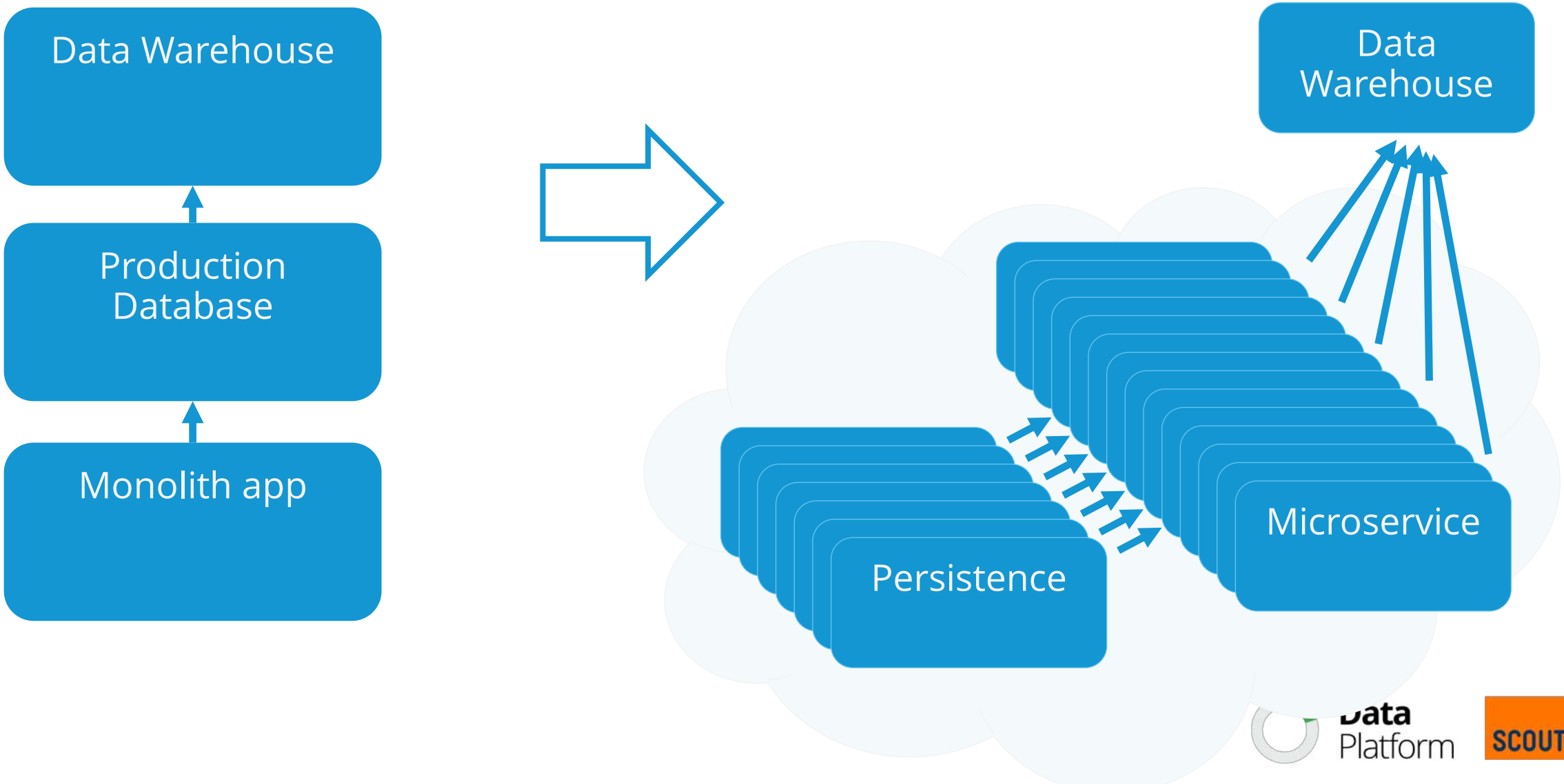
5

Core Geographies
and an overall presence
in 18 countries

80m

Household Reach

Our technical evolution



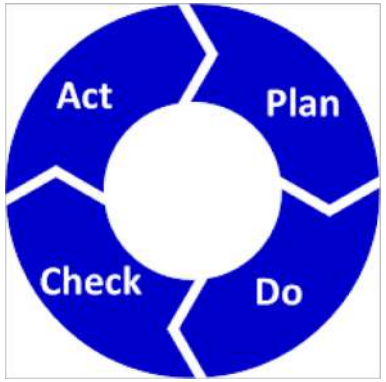
Our data warehouse was a bottle neck

Data Platform

SCOUT 24

Scout24 wants to become a truly data-driven company

Fast & easy data-driven product development…



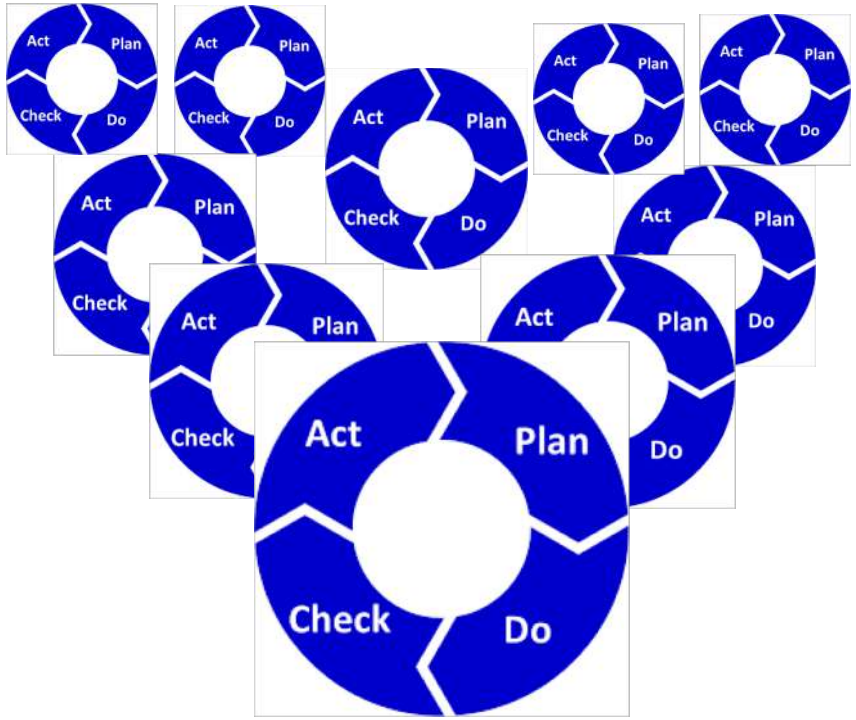
…supported by Data & Analytics

Data Platform

SCOUT 24

Scout24 wants to become a truly data-driven company

Everywhere in the company...



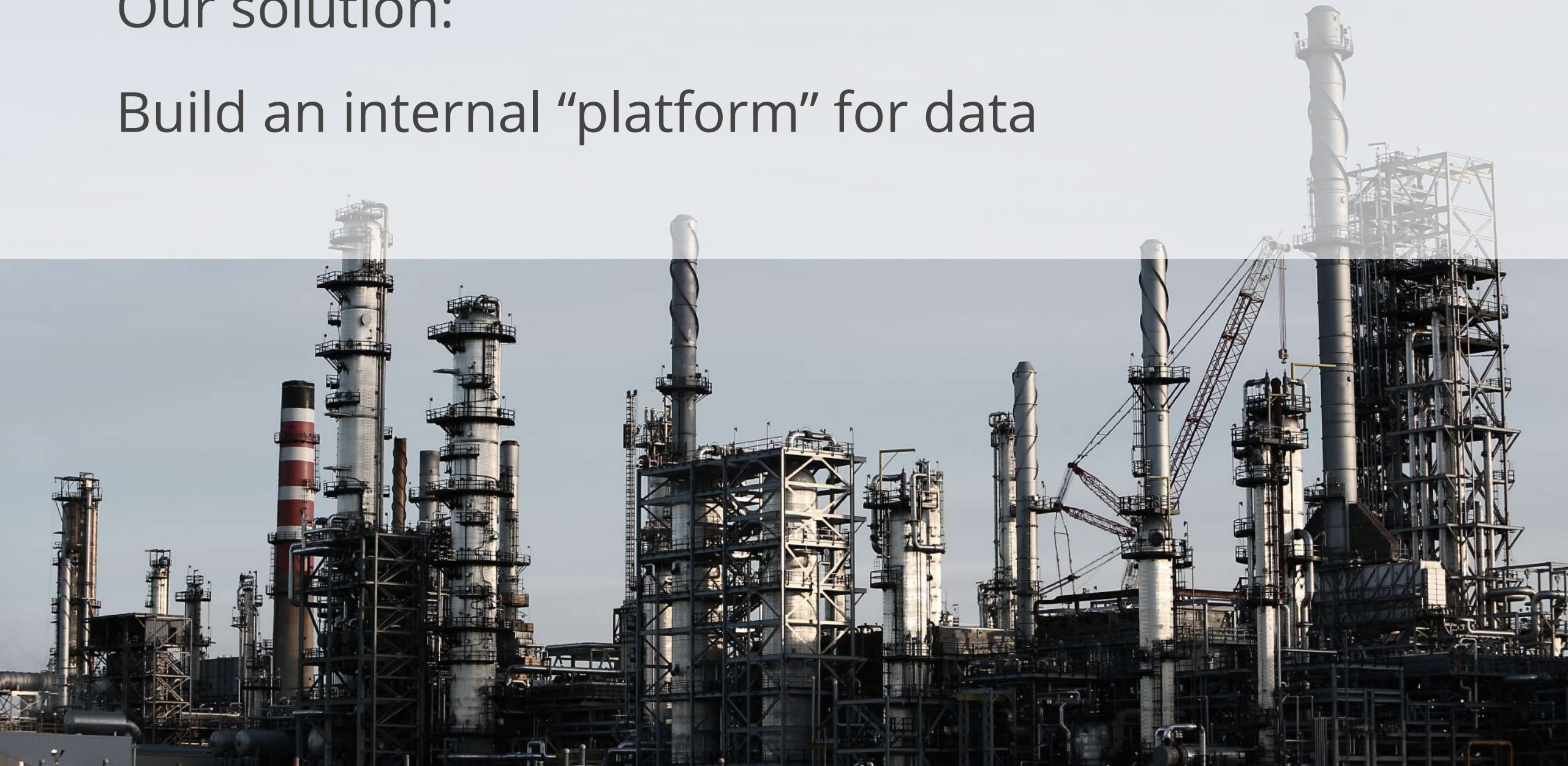
...without bloating up Data & Analytics

Data Platform

SCOUT 24

Our solution:

Build an internal “platform” for data



What is the Scout24 Data “Platform”?

We think of our Data Platform as a Product

- Just like AWS, Salesforce, etc. – the platform is a **generic layer** upon which Scout24's products can be built

- BUT, we have a very, very small number of customers.
- That means, product teams get **personalized support** and there is lots of **opportunity for collaboration.**

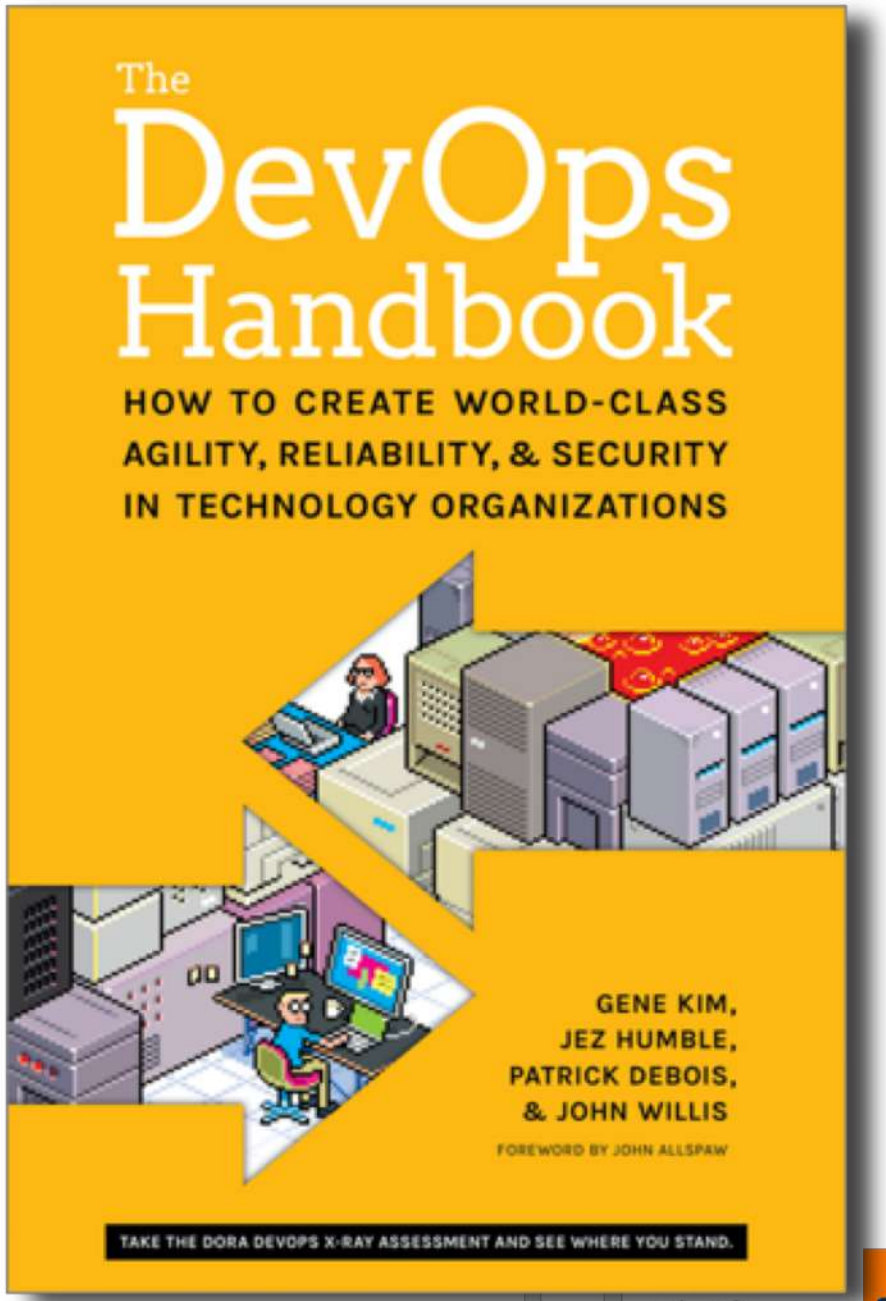
Data Platform

SCOUT 24

We don’t dictate anything.
We just try to make certain things easier by offering a “paved path”

Product teams are **fully empowered** to make their own choices about what is the best use of their resources.

“In almost all cases, we will not mandate that internal team use these platforms and services—**these platform teams will have to win over and satisfy their internal customers**, sometimes even competing with external vendors.”



Guiding principle of the platform

- Autonomy for producers and consumers

Self-service Analytics

Self-service Data Ingestion

Self-service ETL

Data Landscape Manifesto

SCOUT 24

Data Landscape Manifesto

SCOUT 24

Data Landscape Manifesto

Data Landscape Manifesto

A federal landscape of data producers and consumers with just enough rules to ensure seamless cooperation without severely impeding autonomy

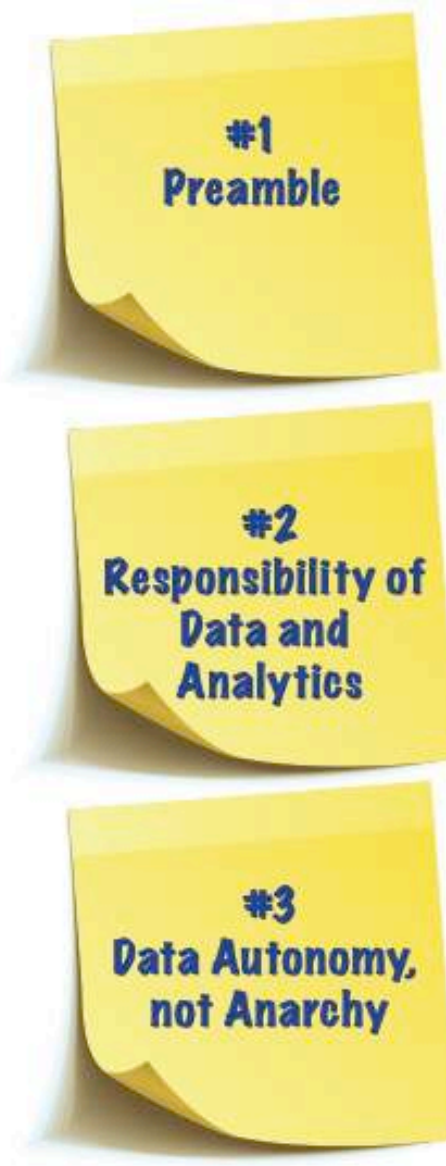
Data Landscape Manifesto

Data Landscape Manifesto

A federal landscape of data producers and consumers with just enough rules to ensure seamless cooperation without severely impeding autonomy

SCOUT 24

Data Landscape Manifesto



Data Landscape Manifesto

A federal landscape of data producers and consumers with just enough rules to ensure seamless cooperation without severely impeding autonomy

SCOUT 24

Data Landscape Manifesto

Data Landscape Manifesto

SCOUT 24

Data Landscape Manifesto

SCOUT 24

Data Landscape Manifesto

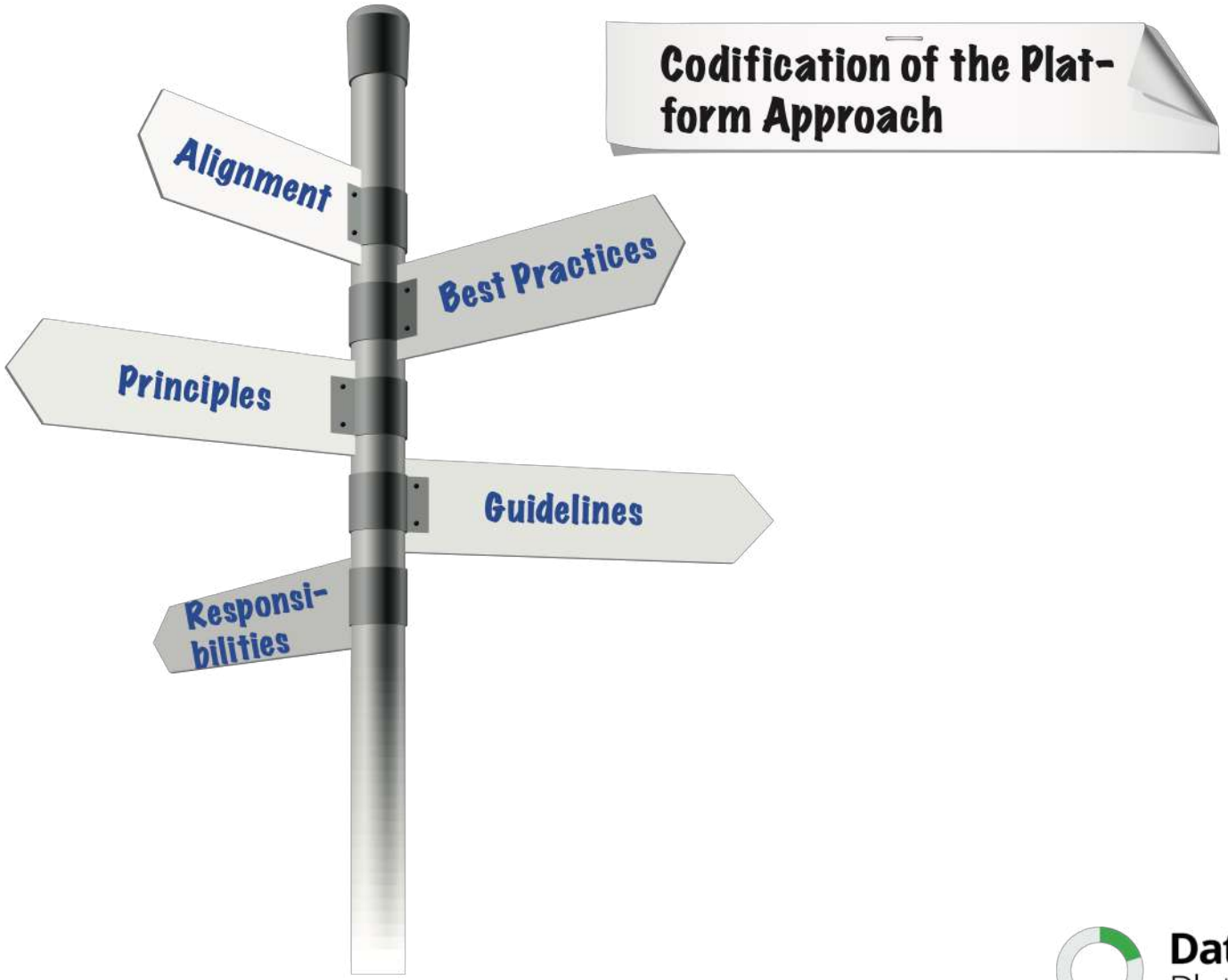
SCOUT 24

Data Landscape Manifesto – Purpose

Data Platform

SCOUT 24

Data Landscape Manifesto – Purpose



Data Platform

SCOUT 24

Necessary Cultural Changes during Migration from Data Warehouse to Cloud Data Platform

Dr. Sean Gustafson, Scout24 AG

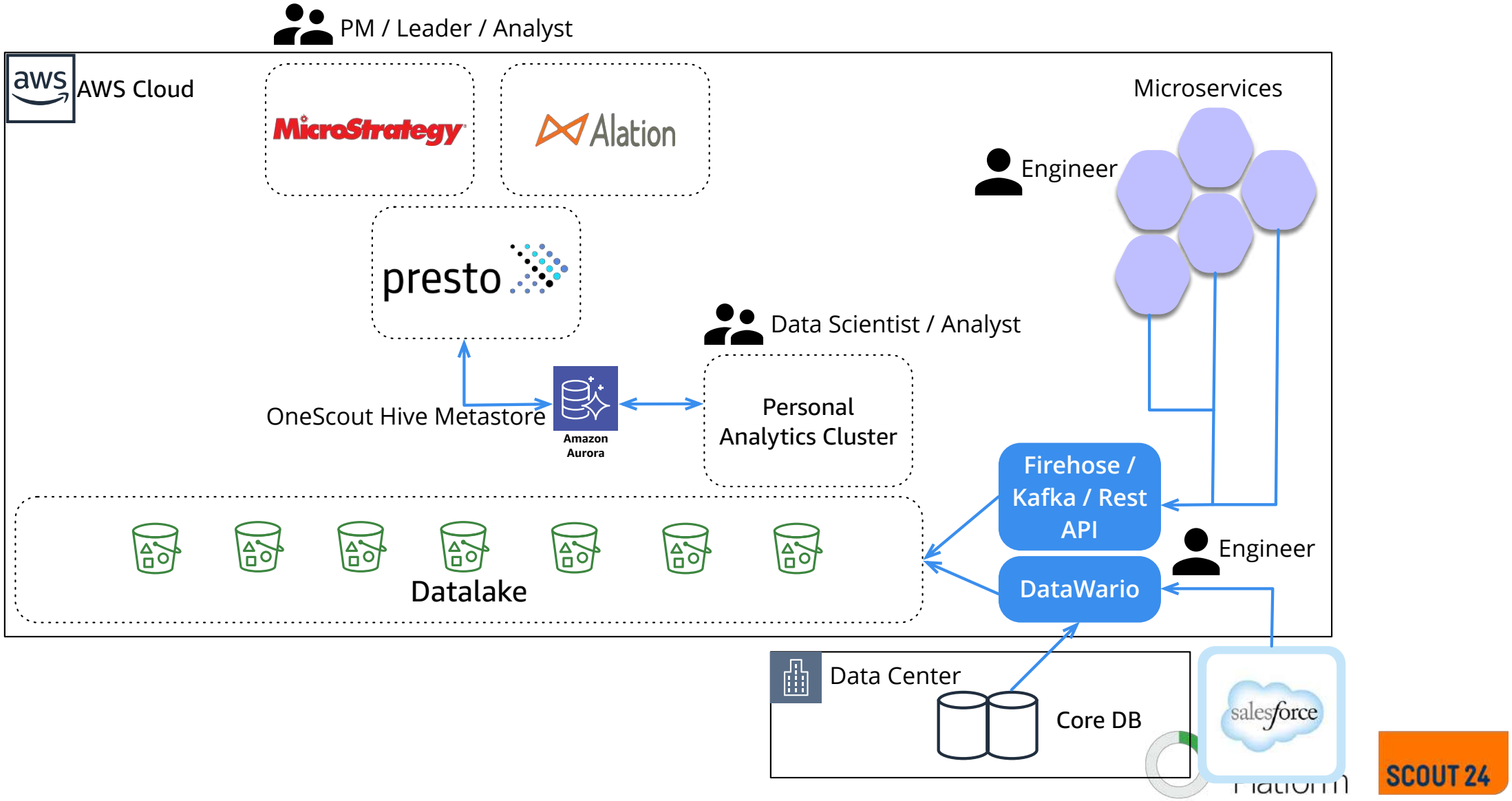
Data Festival 2018

Watch the Video | Read the Blog Post on Medium

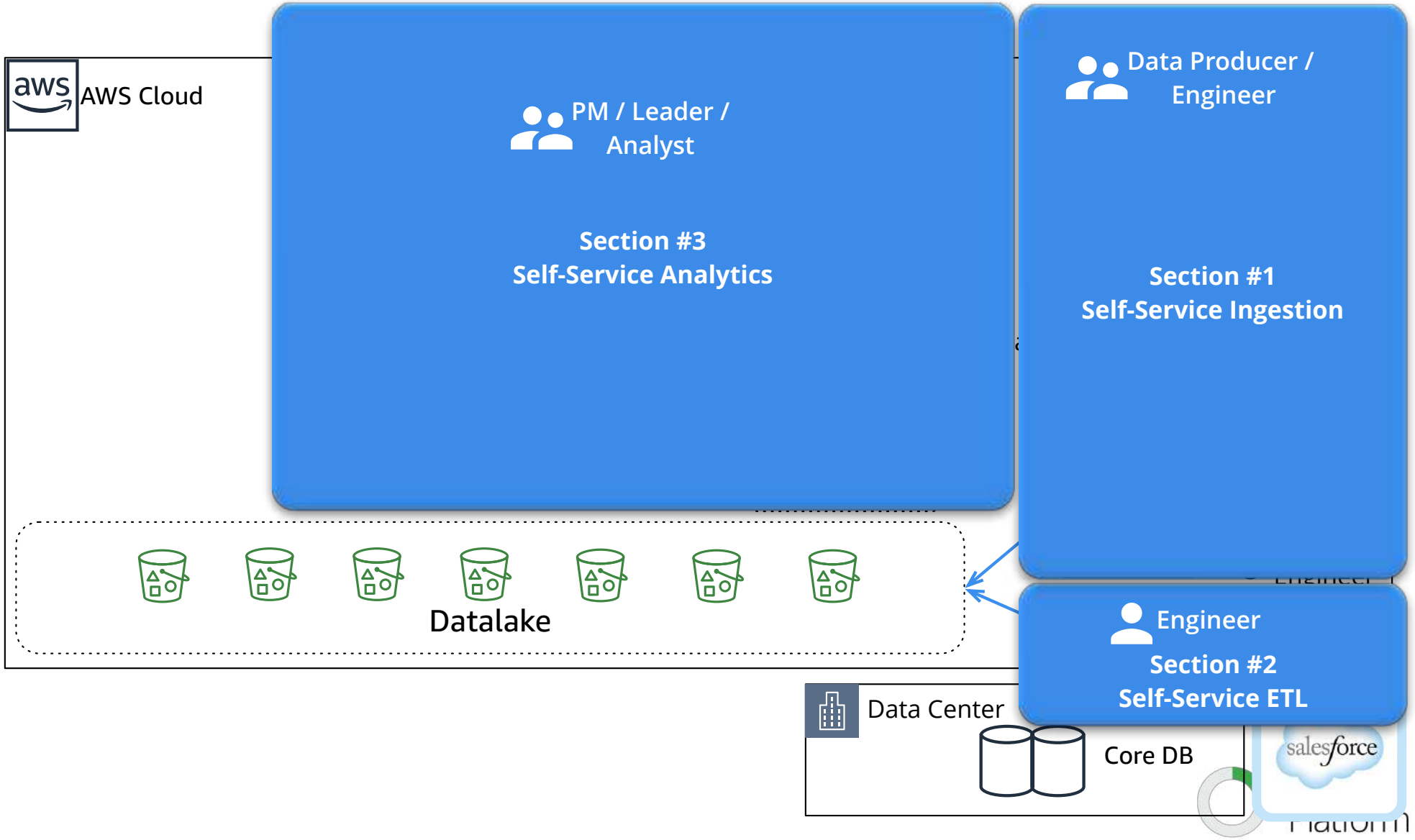
Data Platform

SCOUT 24

10,000 Foot Architecture Overview



10,000 Foot Architecture Overview

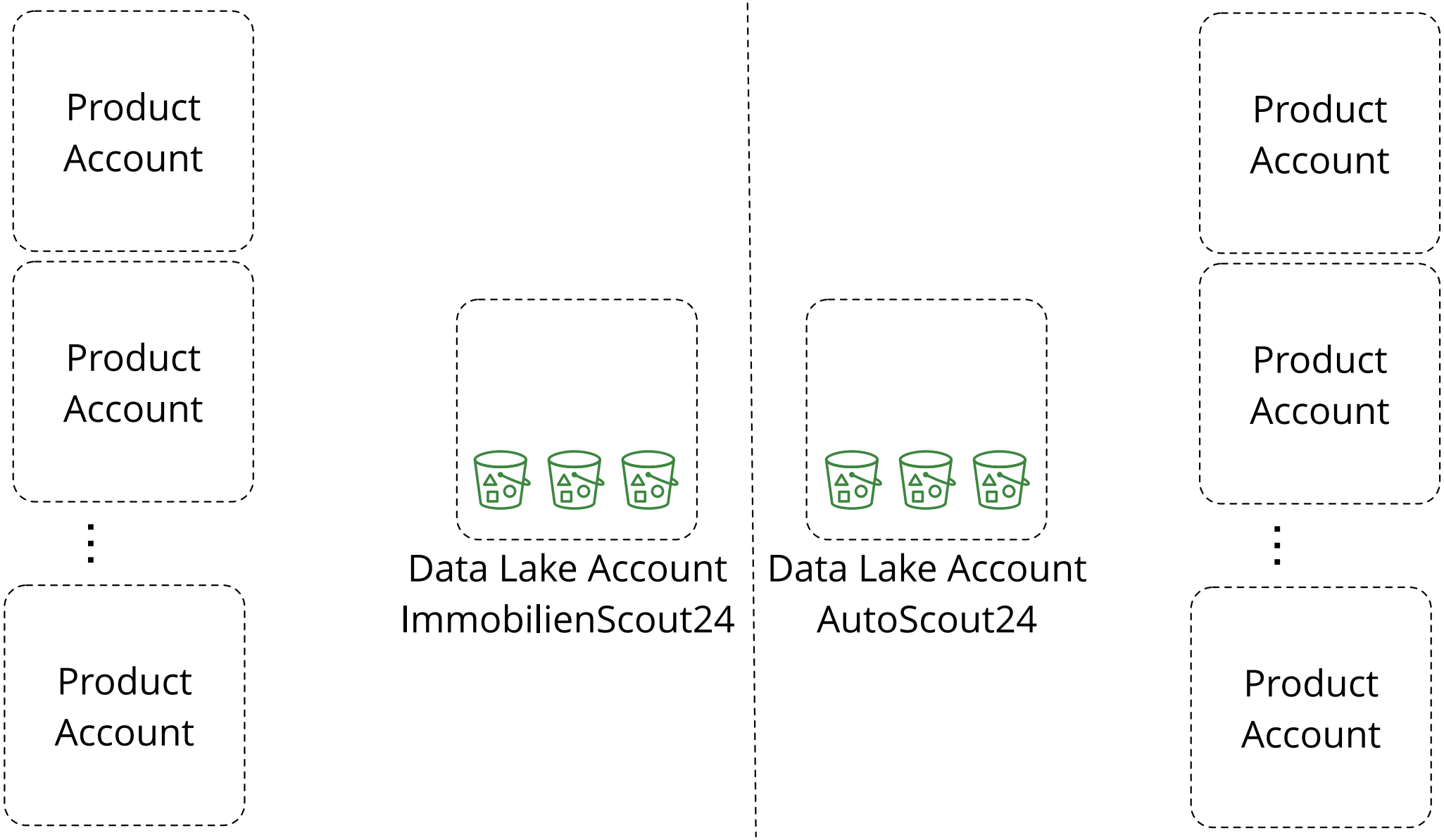


Self-Service Ingestion

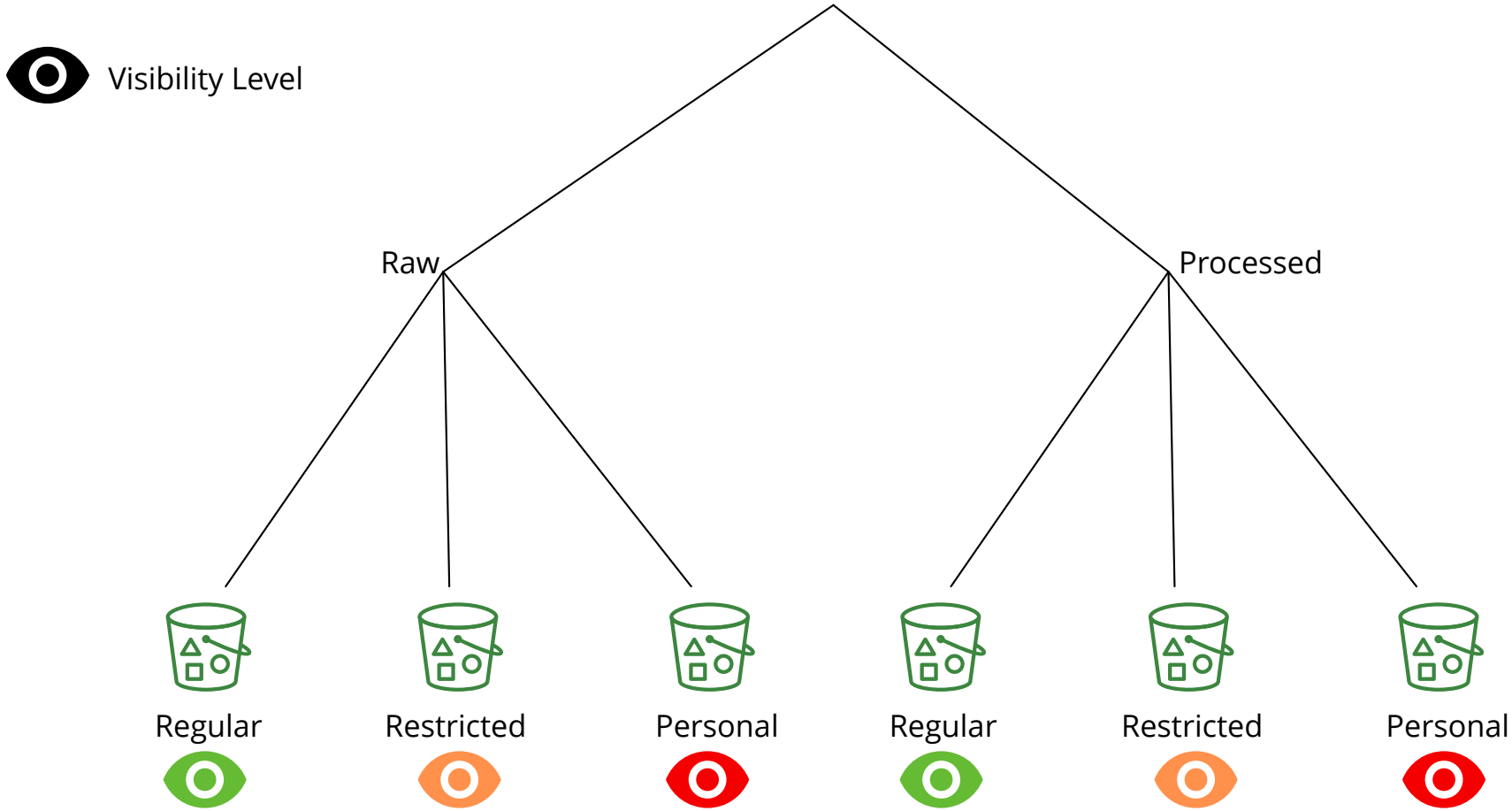
Data
Platform

SCOUT 24

Multi-Account Setting



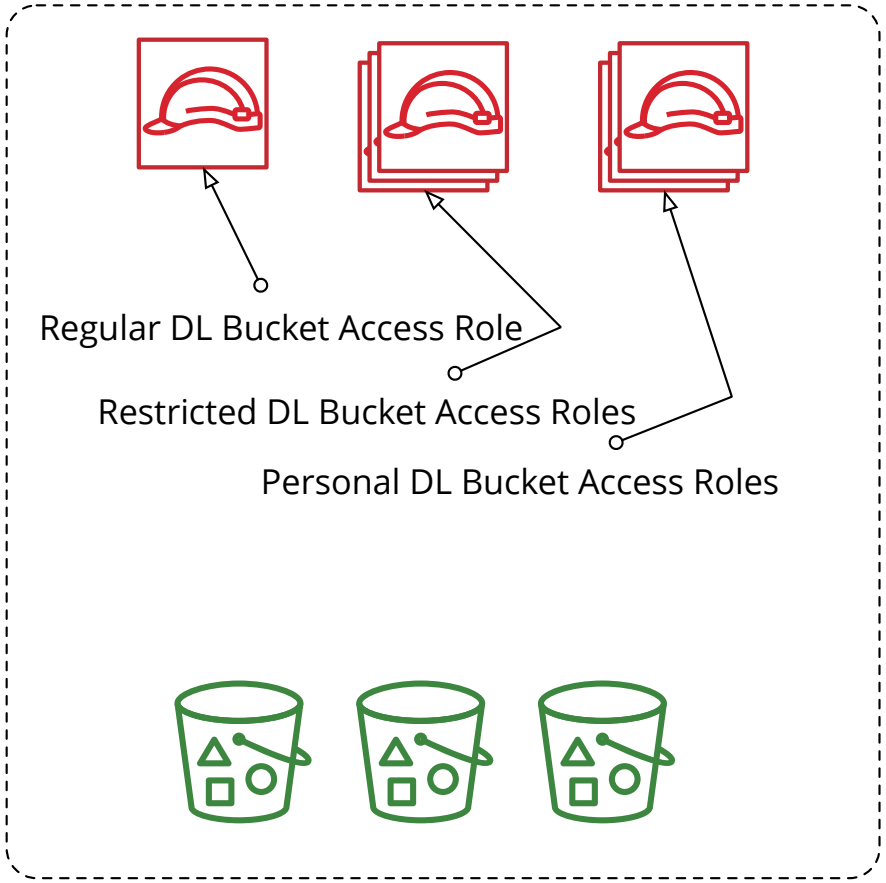
Data Lake Bucket Types



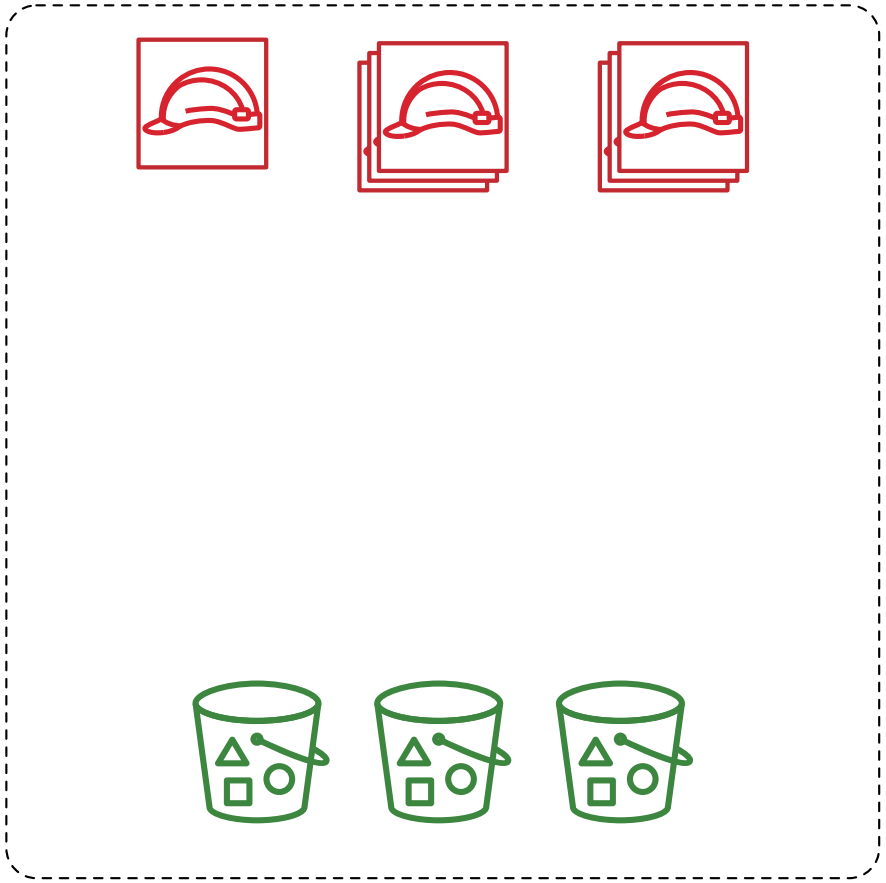
Data Platform

SCOUT 24

Data Lake Access Roles



Data Lake Account
ImmobilienScout24

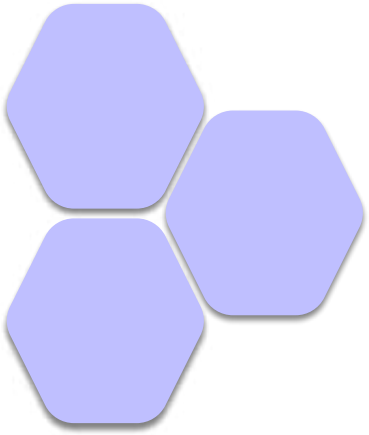


Data Lake Account
AutoScout24

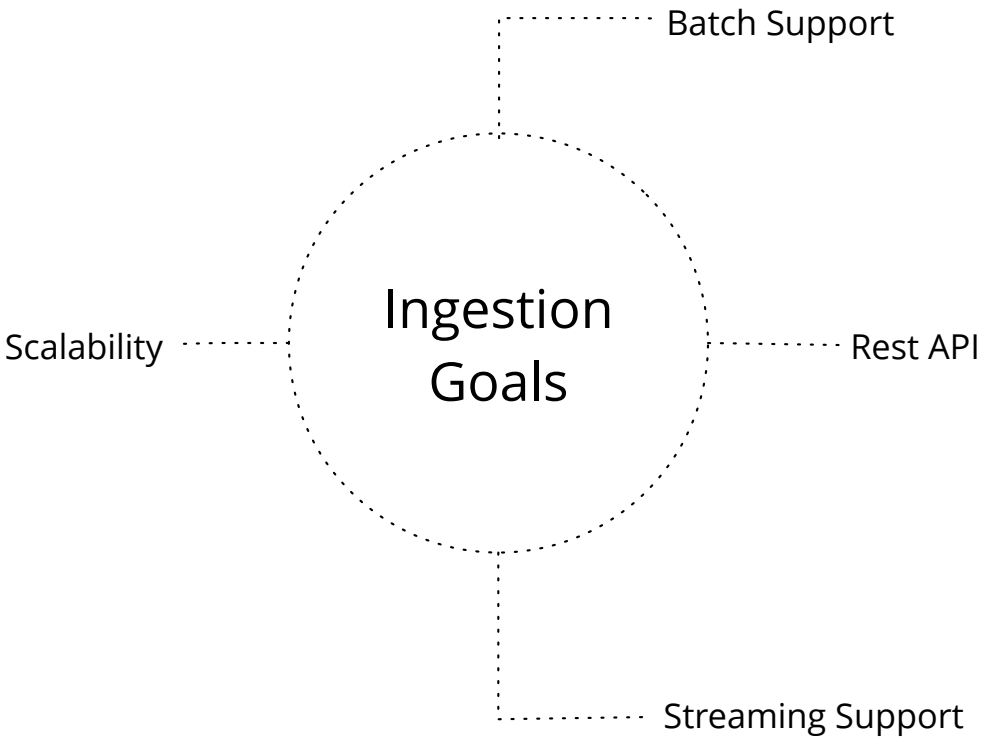
Data Platform

SCOUT 24

Data Lake Ingestion Goals



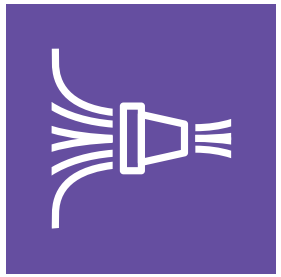
Microservices



Data Platform

SCOUT 24

Ingestion Options



Amazon Kinesis Data Firehose

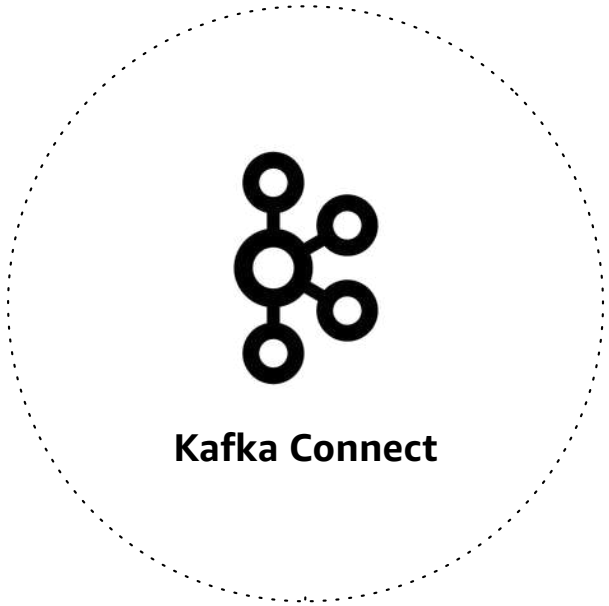
Kafka Connect

Hadoop Rest API

Data Platform

SCOUT 24

Ingestion Options



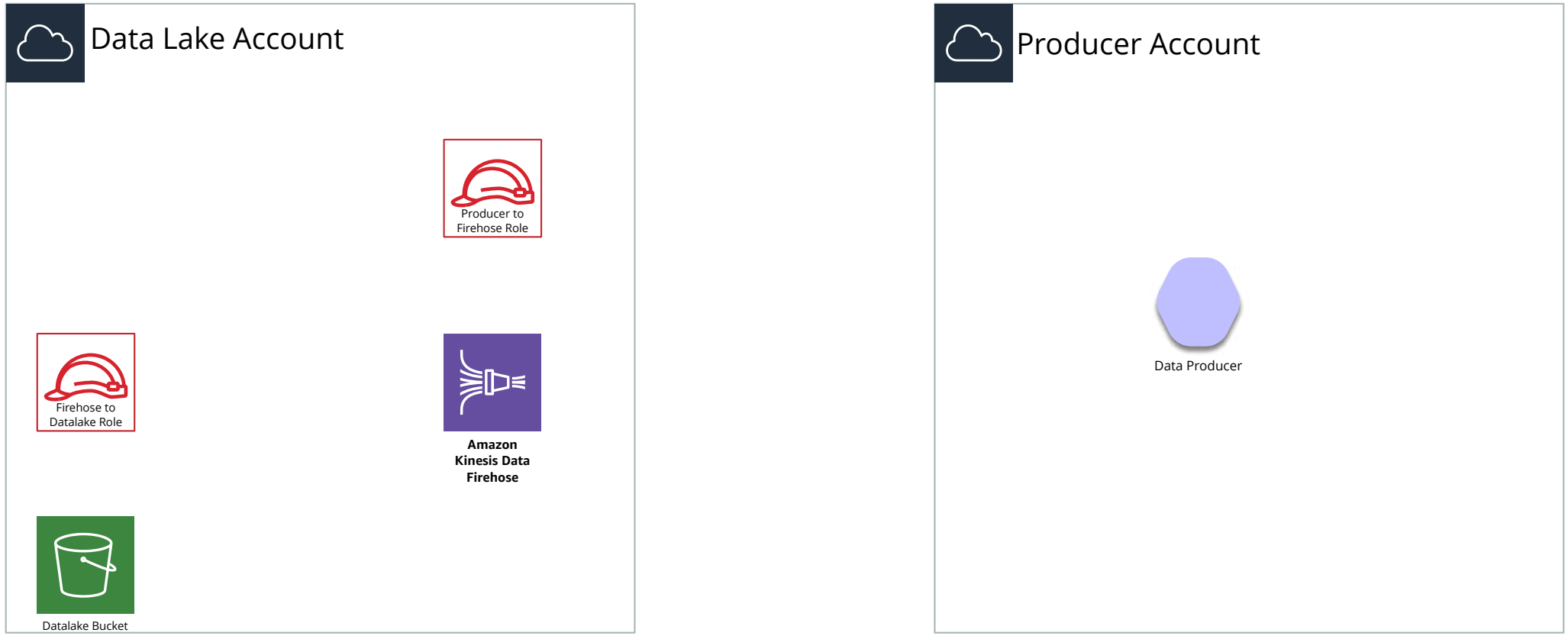
Configuration Free

Data Platform

SCOUT 24

Kinesis Data Firehose Ingestion Mechanism

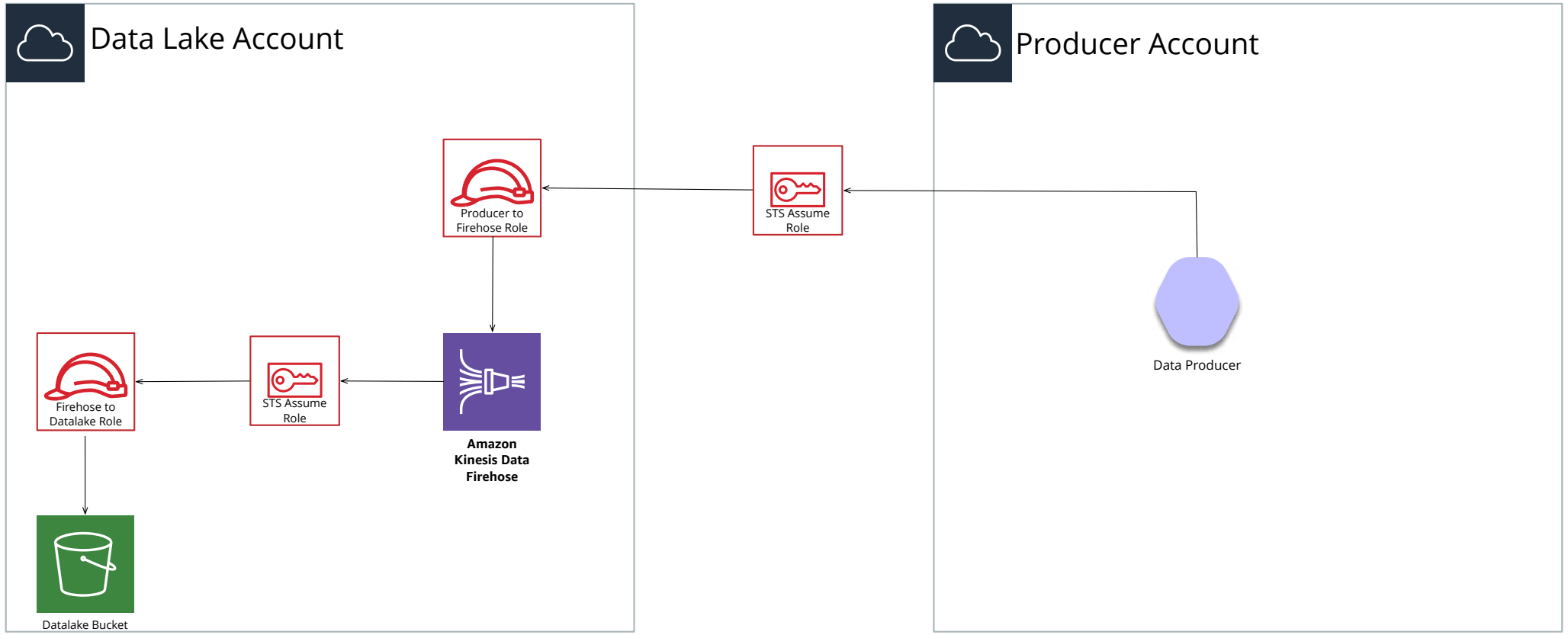
Firehose Ingestion Architecture



Data Platform

SCOUT 24

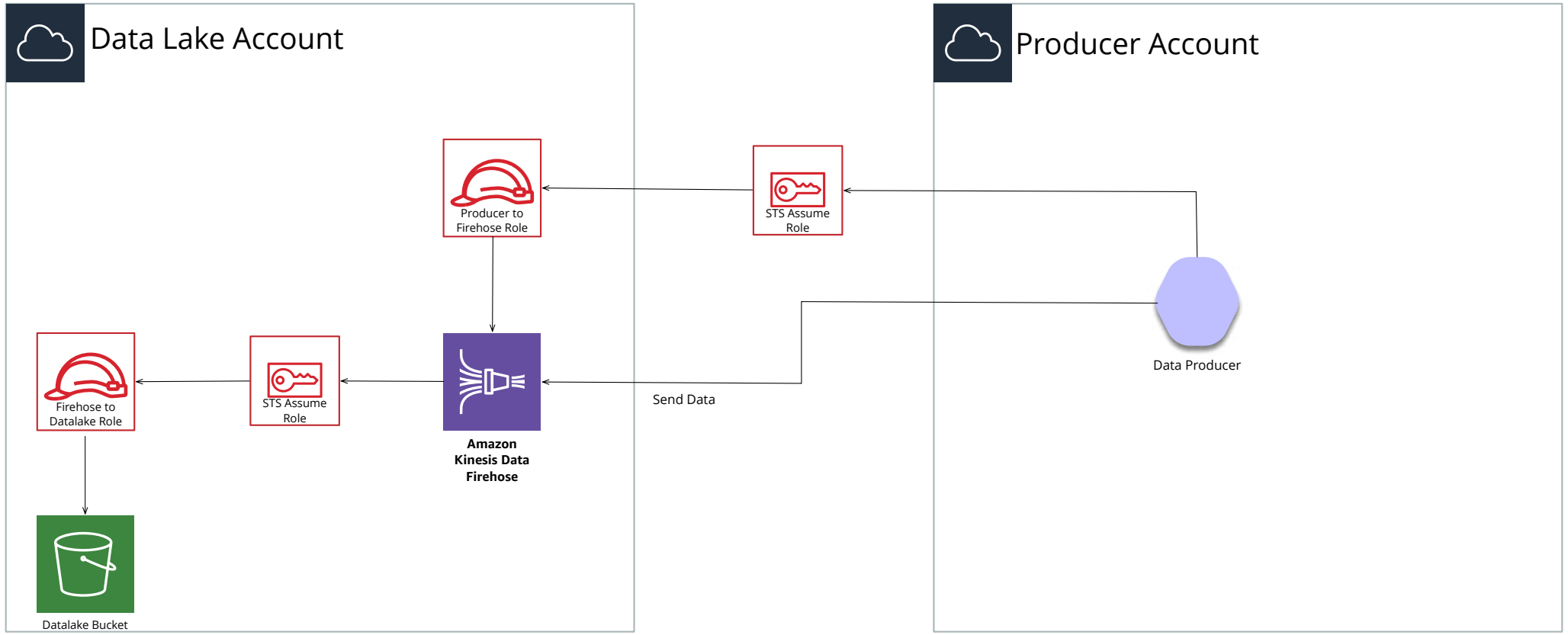
Firehose Ingestion Architecture



Data Platform

SCOUT 24

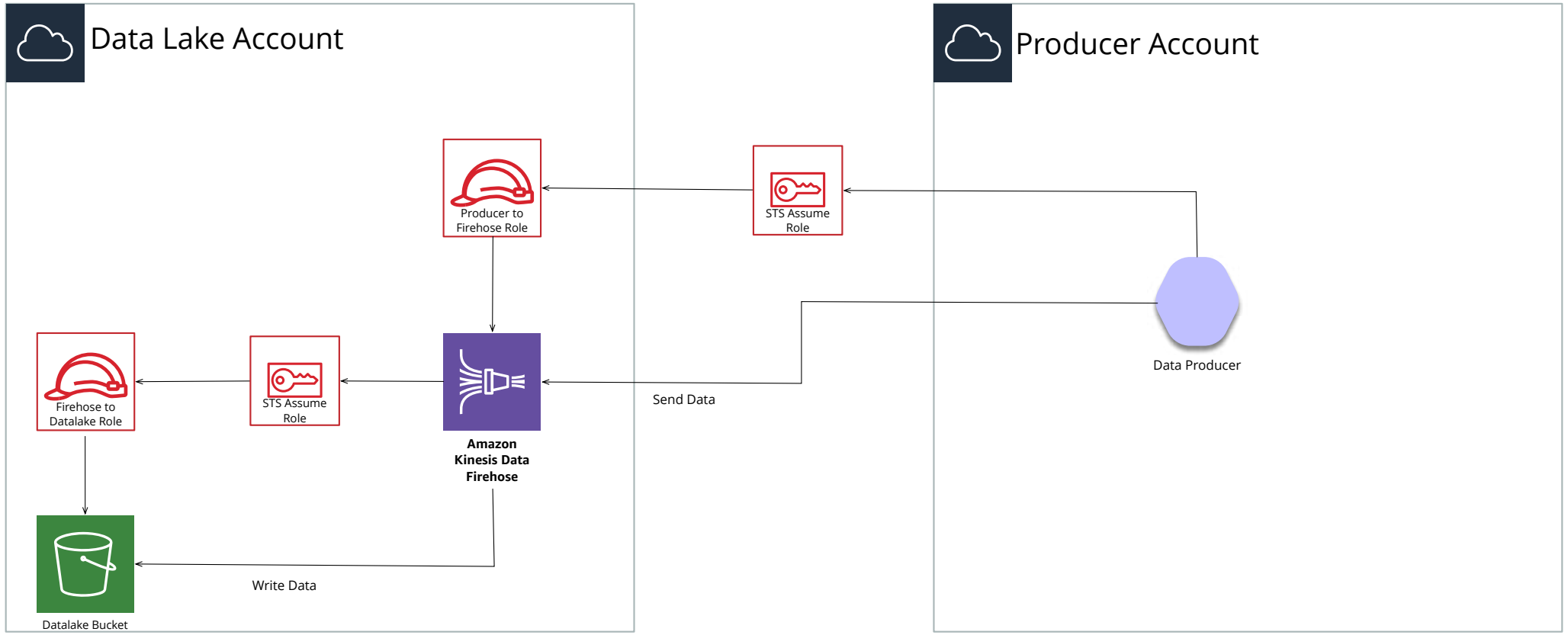
Firehose Ingestion Architecture



Data Platform

SCOUT 24

Firehose Ingestion Architecture



Data Platform

SCOUT 24

Kafka Connect Ingestion Mechanism

Data
Platform

SCOUT 24

Kafka Connect

Data Platform

SCOUT 24

Kafka Connect

ActiveMQ

Kafka

Elasticsearch

Amazon S3

Cassandra

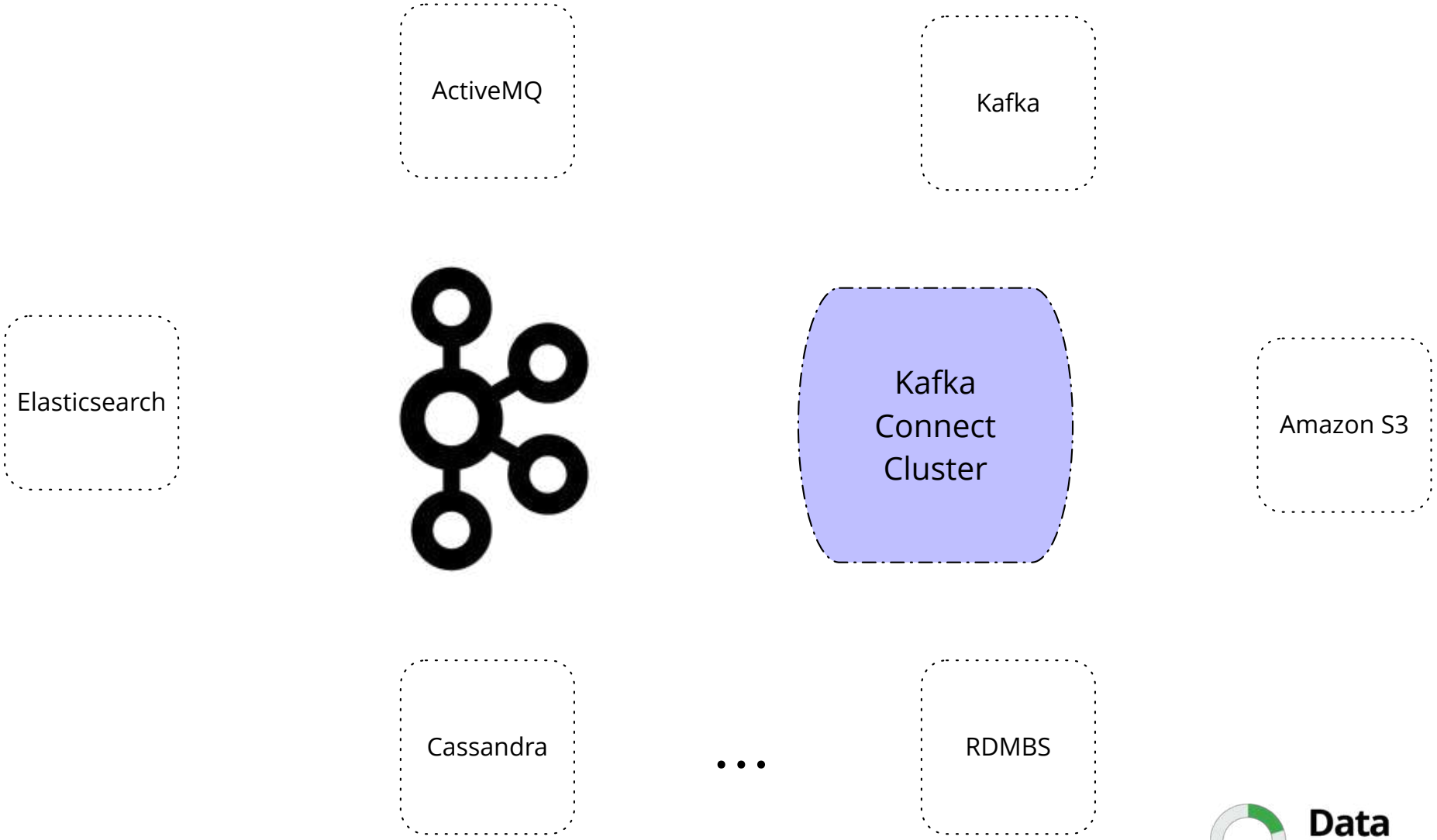
...

RDMBS

Data Platform

SCOUT 24

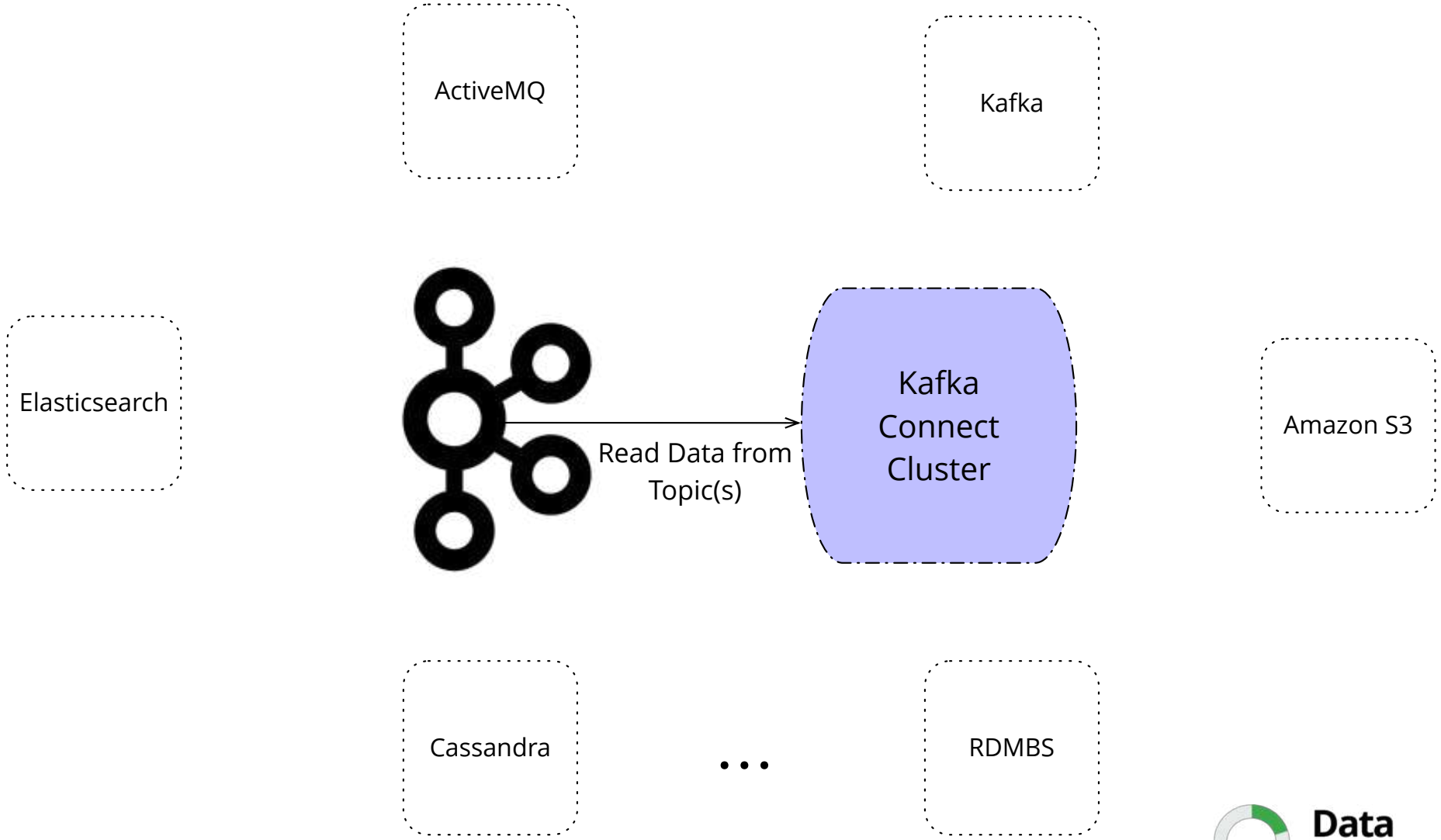
Kafka Connect



Data Platform

SCOUT 24

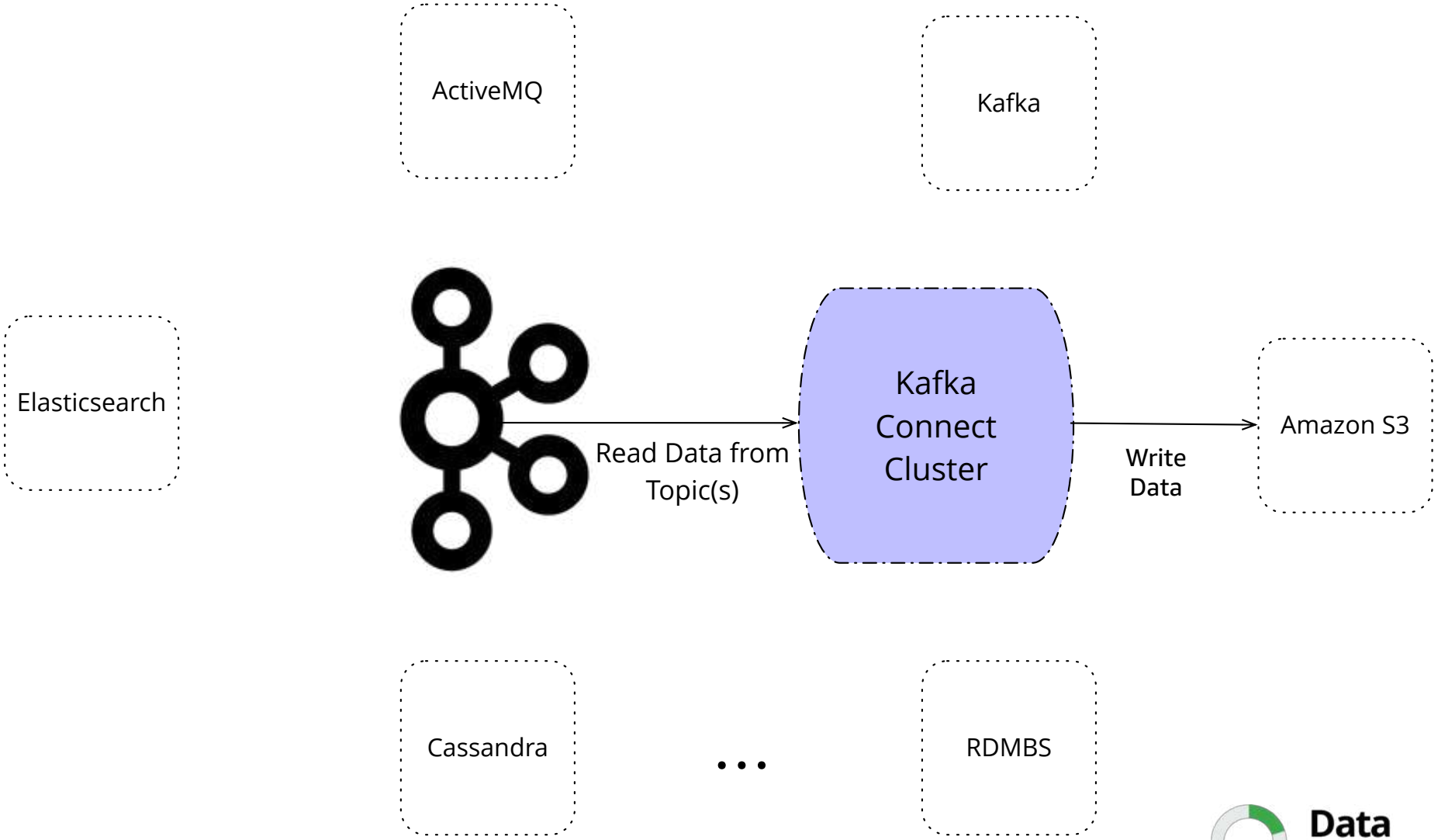
Kafka Connect



Data Platform

SCOUT 24

Kafka Connect



Data Platform

SCOUT 24

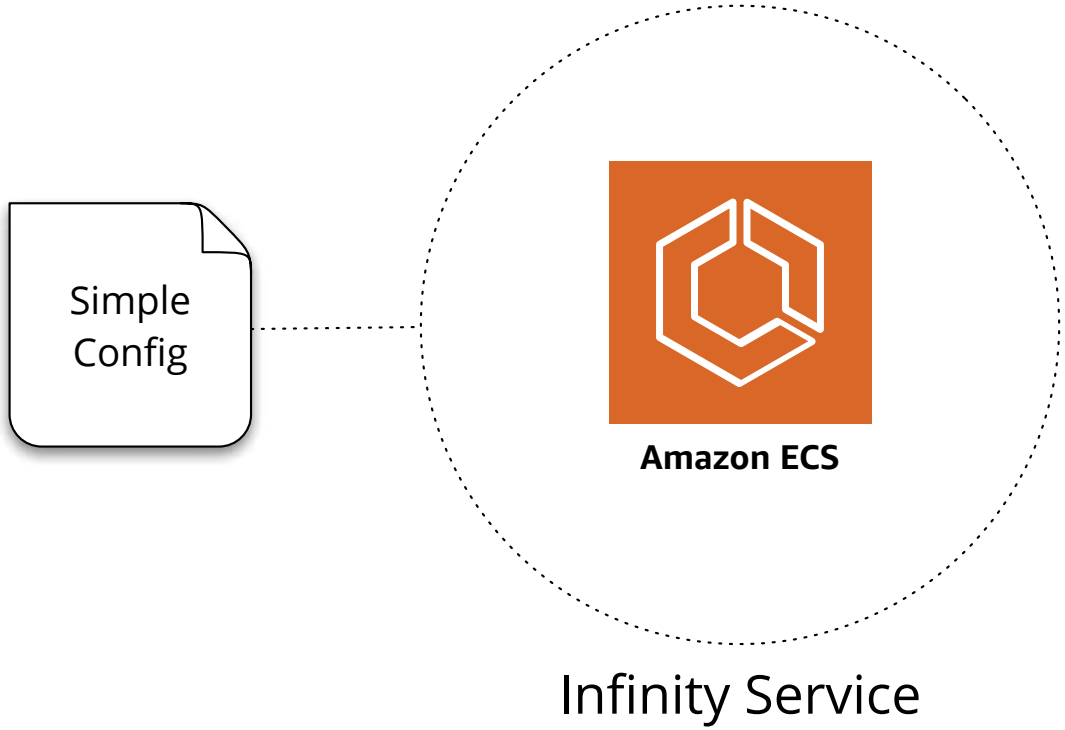
Scout24 Infinity Cluster

Infinity Service

Data Platform

SCOUT 24

Scout24 Infinity Cluster



Data Platform

SCOUT 24

Scout24 Infinity Cluster

Data Platform

SCOUT 24

To Infinity and Beyond Handling Heterogenous Container Clusters in AWS

Christine Trahe, Scout24 AG

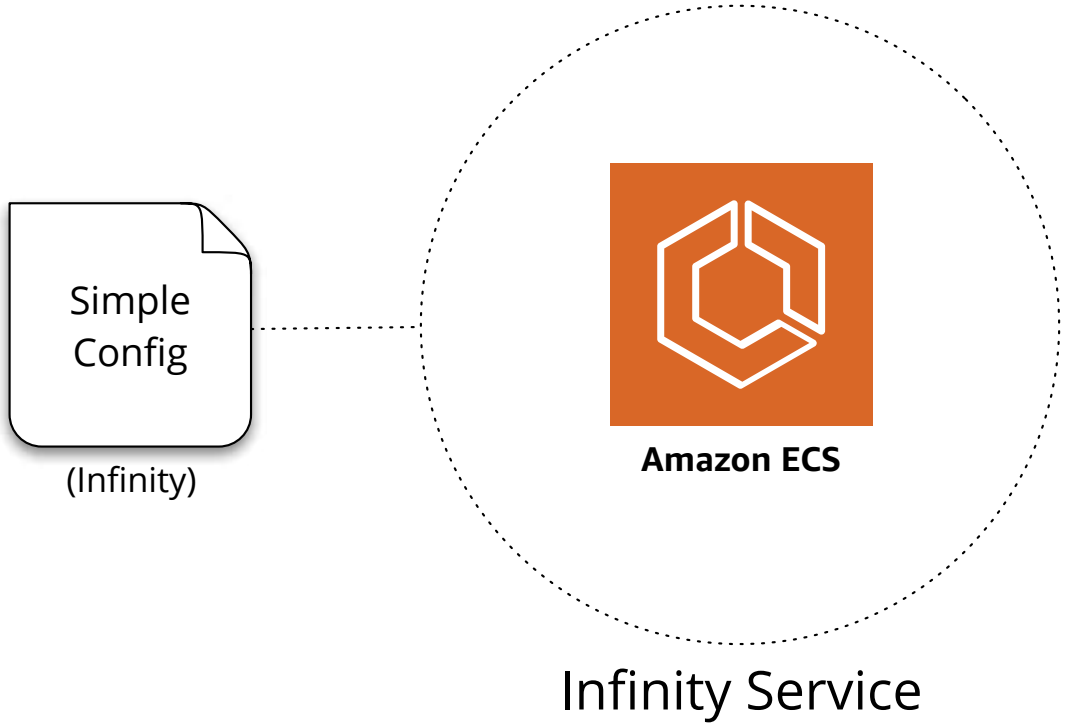
AWS Summit Berlin 2019

Get the Slide Deck

Data Platform

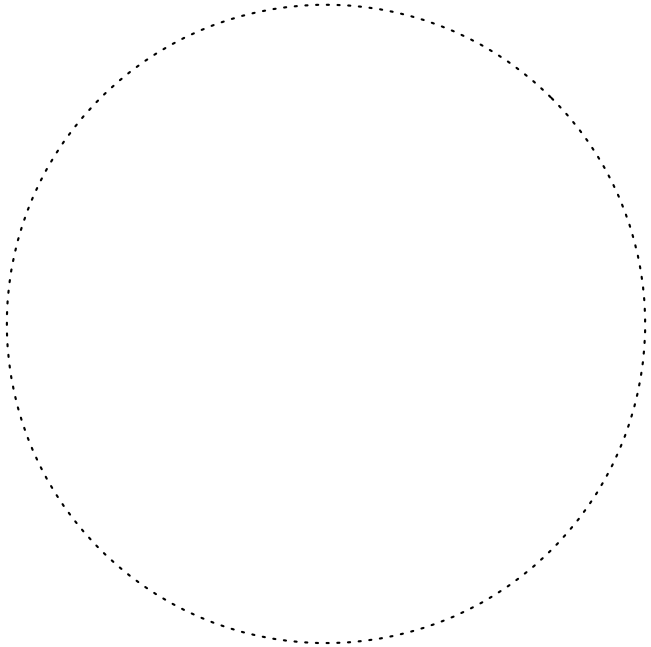
SCOUT 24

Kafka Connect on Infinity



Data Platform SCOUT 24

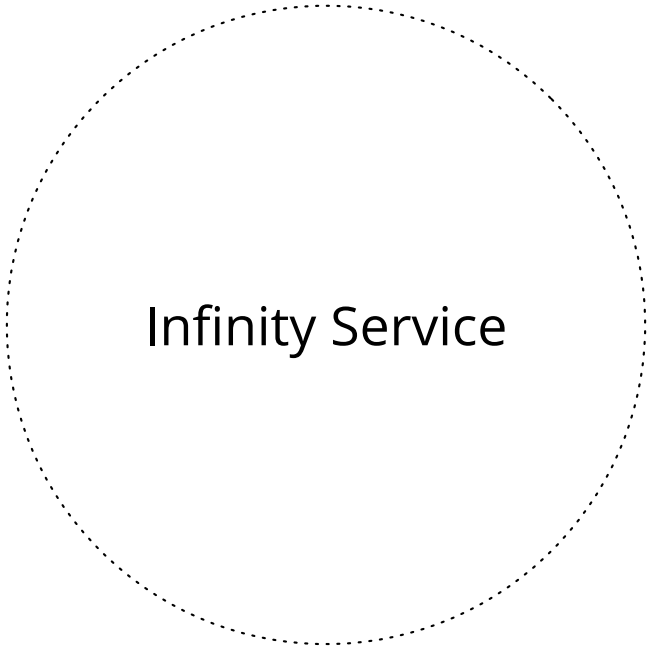
Kafka Connect on Infinity



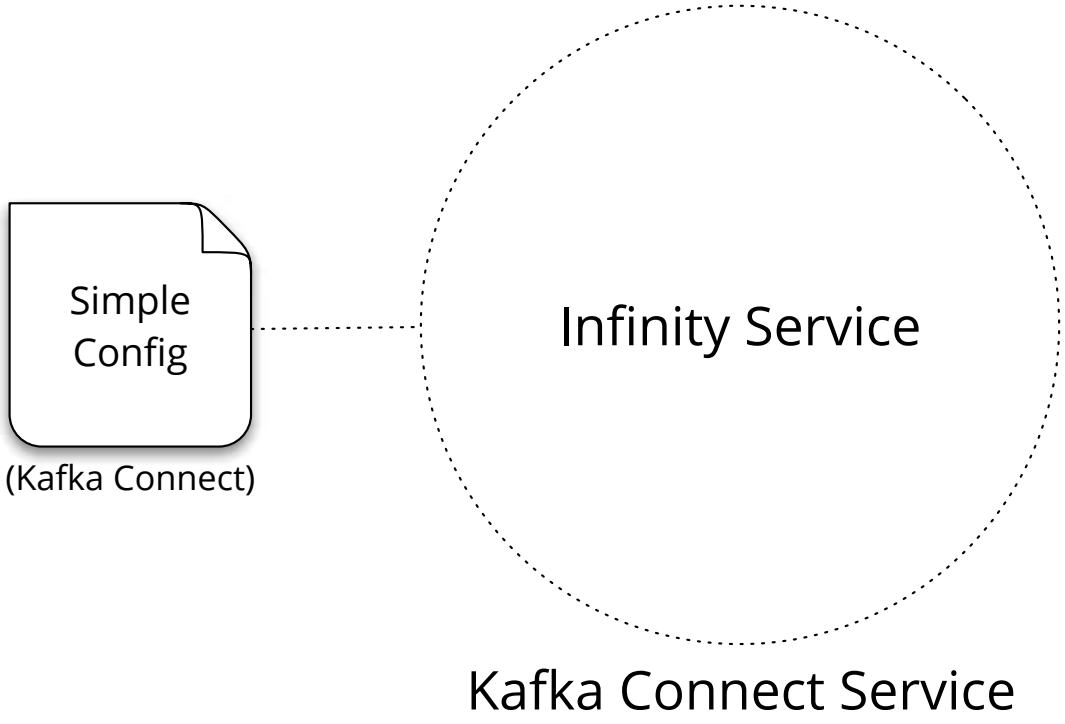
Data Platform

SCOUT 24

Kafka Connect on Infinity



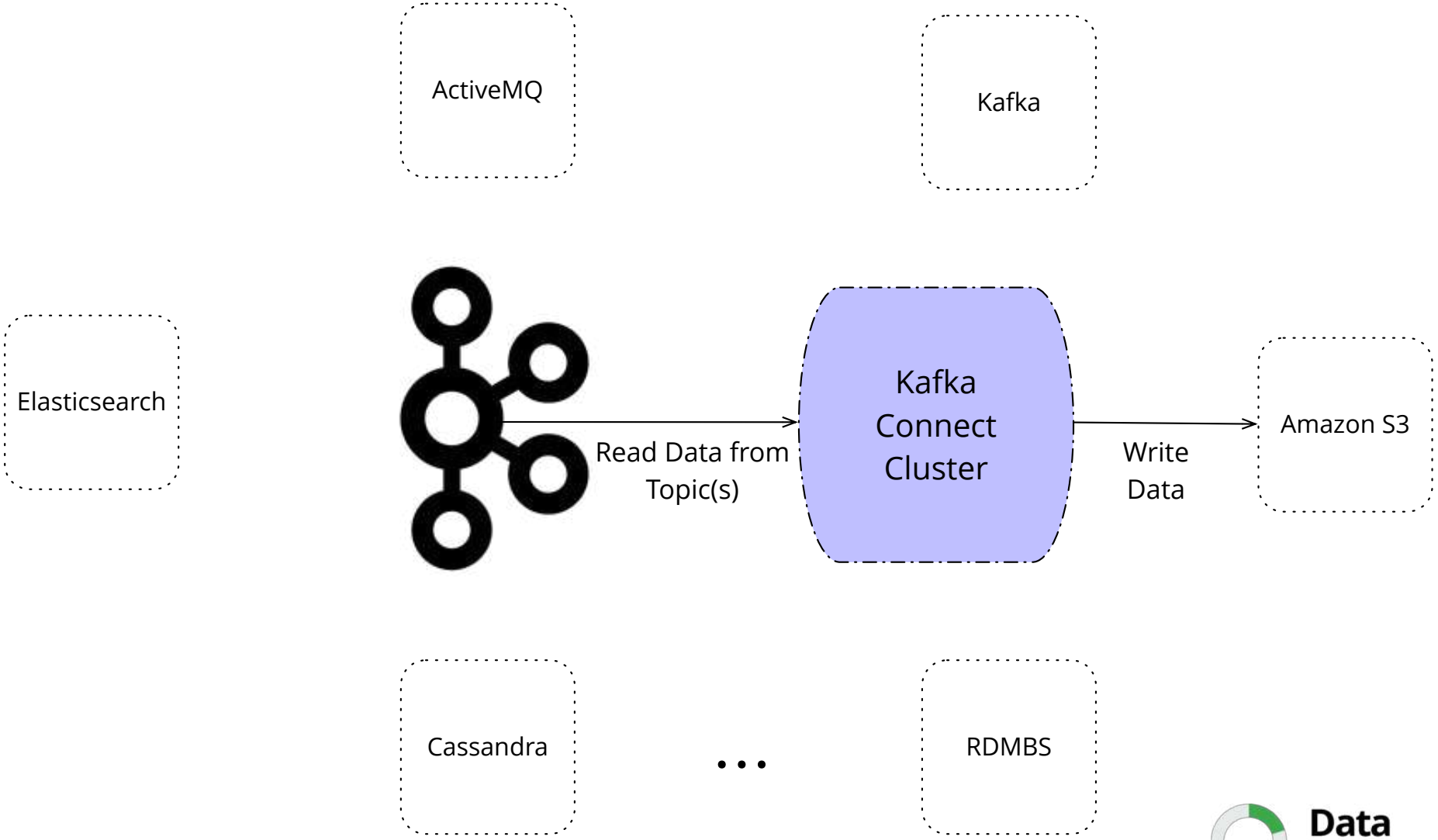
Kafka Connect on Infinity



Data Platform

SCOUT 24

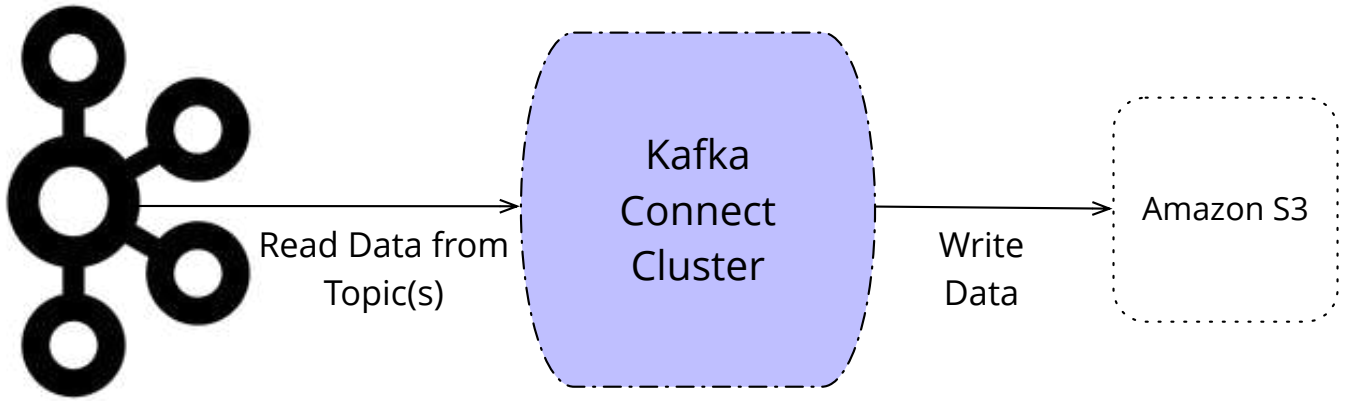
Kafka Connect Deployment



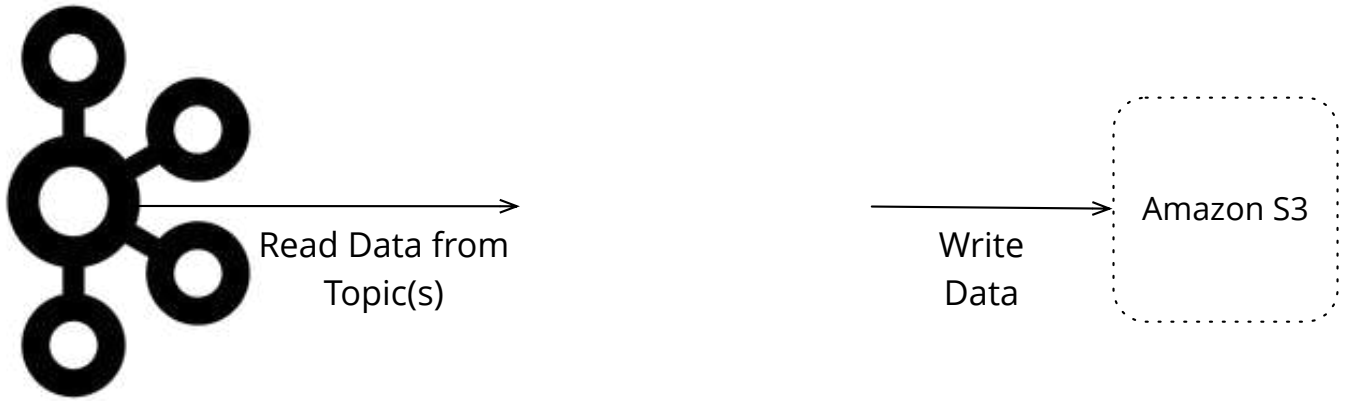
Data Platform

SCOUT 24

Kafka Connect Deployment



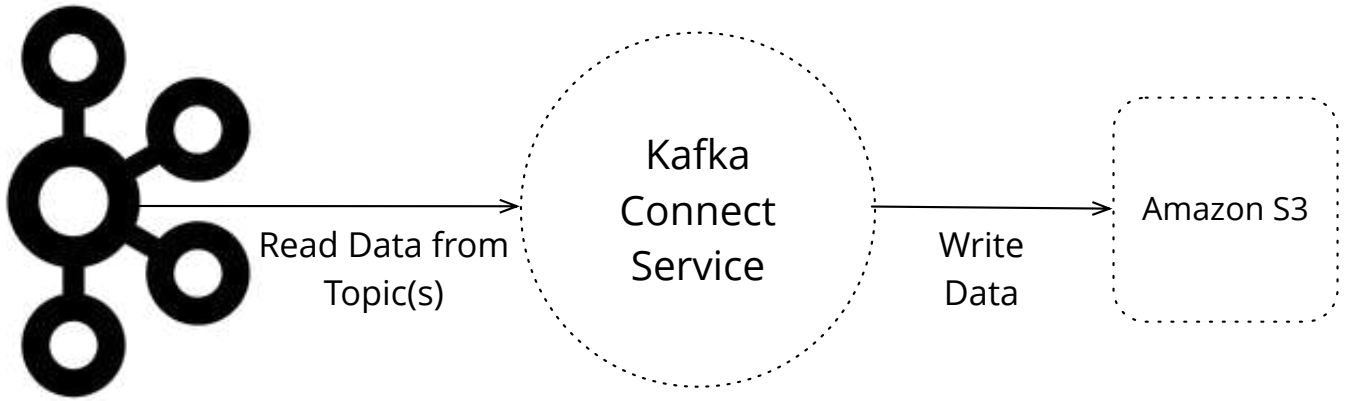
Kafka Connect Deployment



Data Platform

SCOUT 24

Kafka Connect Deployment



Kafka Connect Deployment

Data Platform

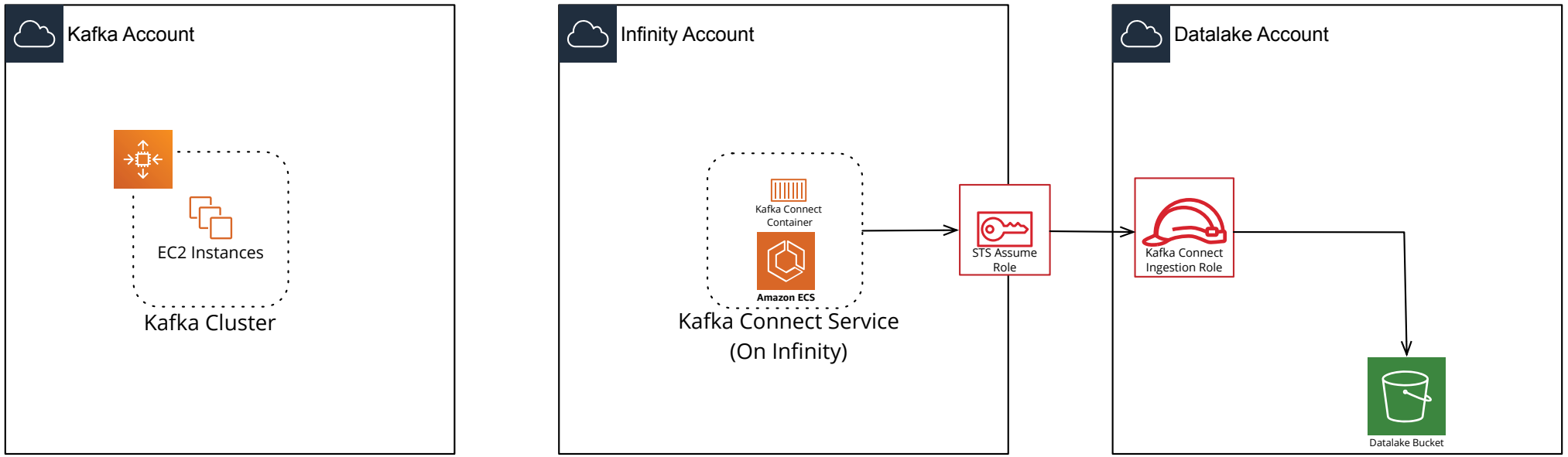
SCOUT 24

Kafka Connect Deployment

Data Platform

SCOUT 24

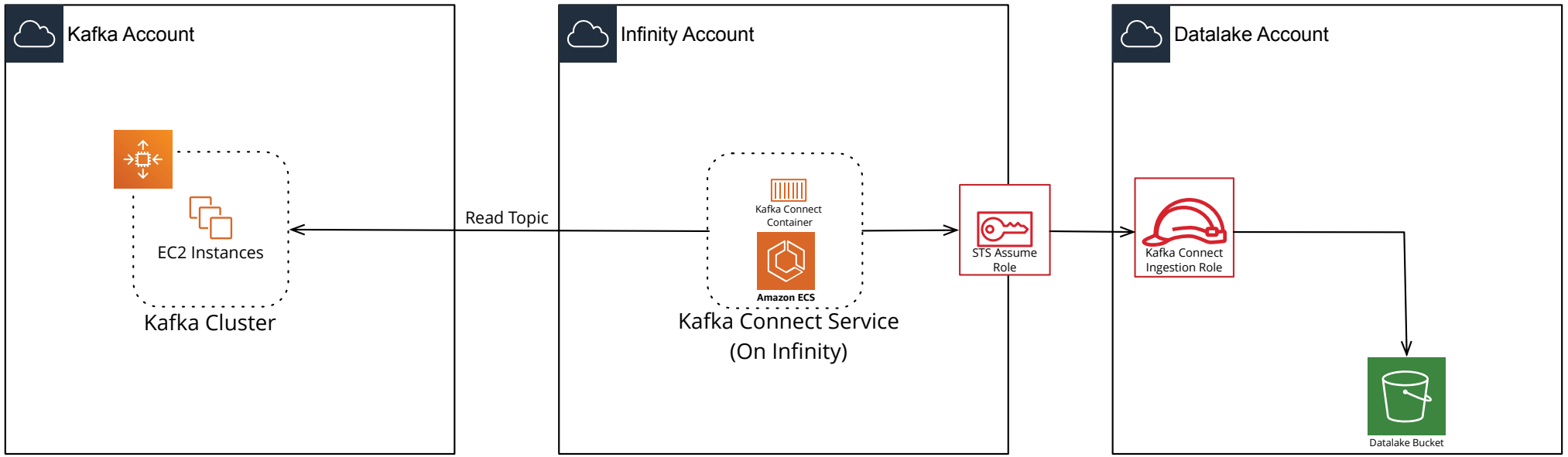
Kafka Connect Deployment



Data Platform

SCOUT 24

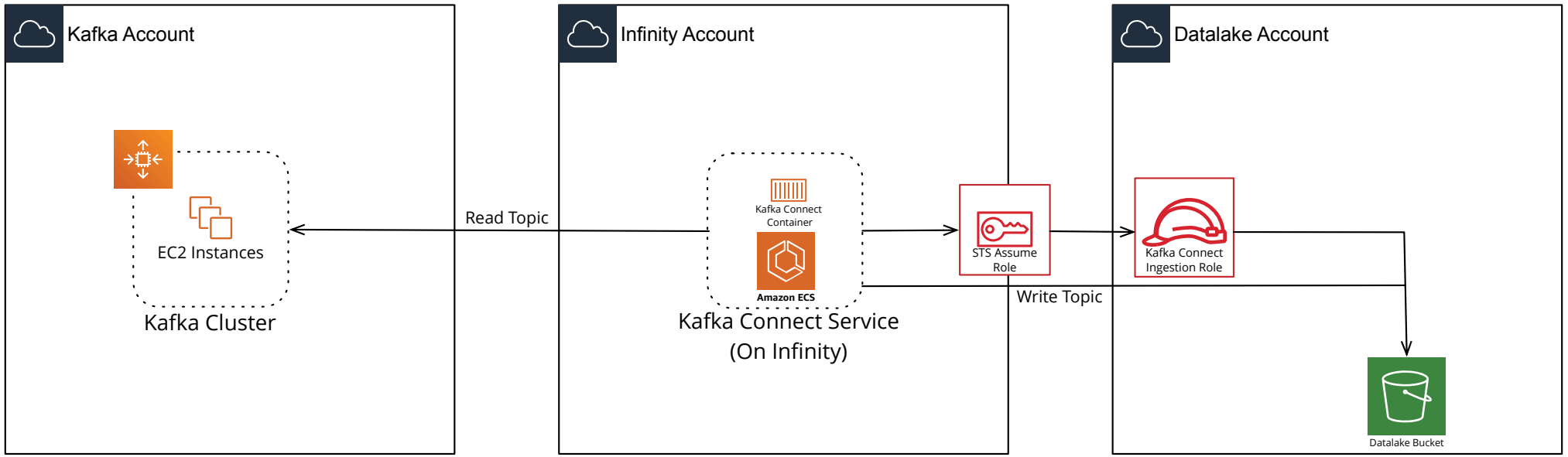
Kafka Connect Deployment



Data Platform

SCOUT 24

Kafka Connect Deployment



Data Platform

SCOUT 24

Hadoop Rest API Ingestion Mechanism

Data Platform

SCOUT 24

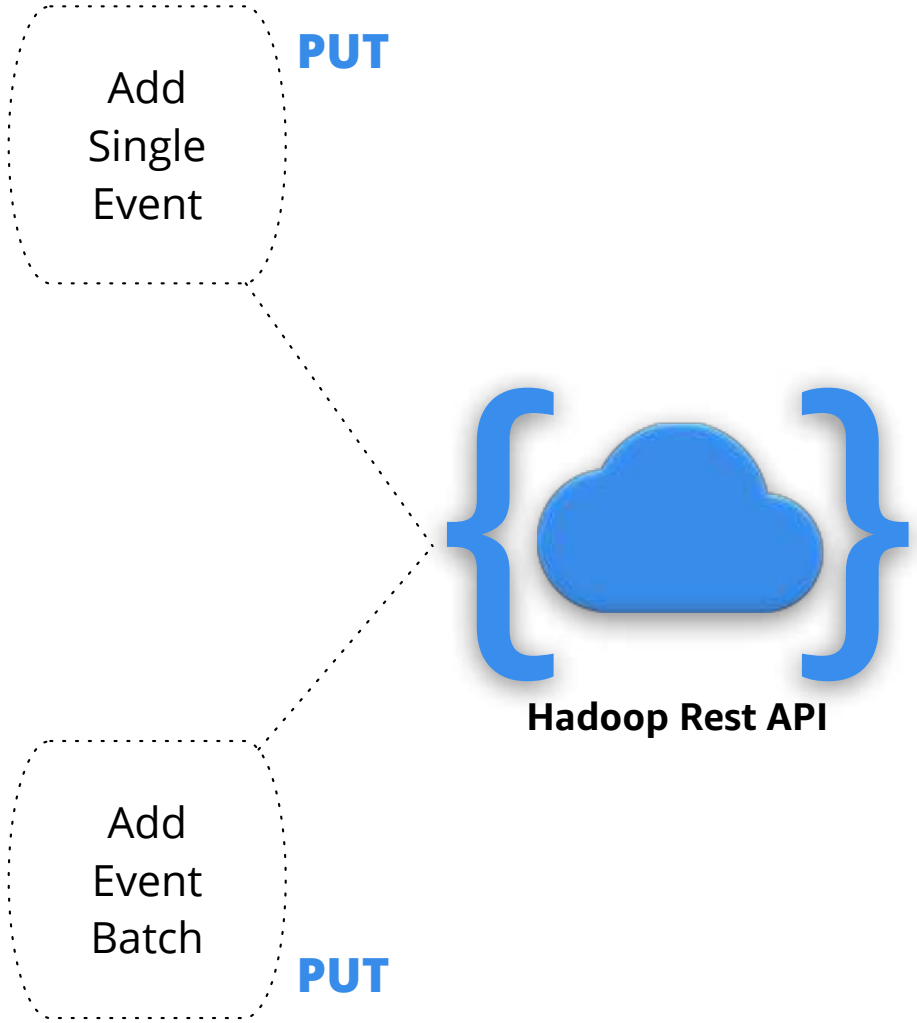
Hadoop Rest API – A Motivation

- Unified Set of Ingestion Operations
- Stable User-Facing Endpoints, Varying Backend
- Adds Reception Timestamp Field
- Performs Event Validation

Data Platform

SCOUT 24

Hadoop Rest API – Operation Overview



Automatic Partitioning by Ingestion Date

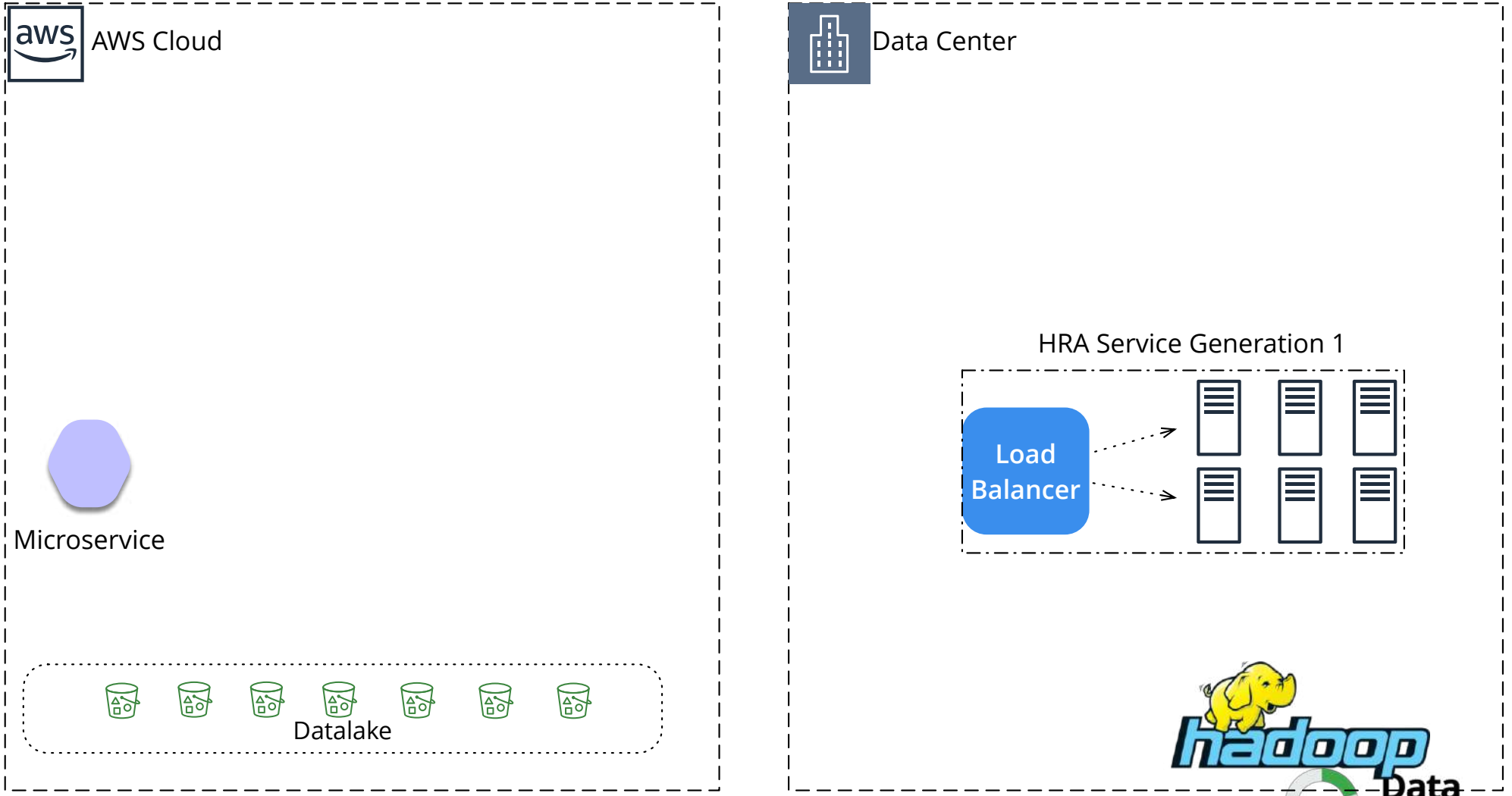
Persisting to Disk Initially, Periodically Flushing to Target Store

Also Publishing to Kinesis Stream

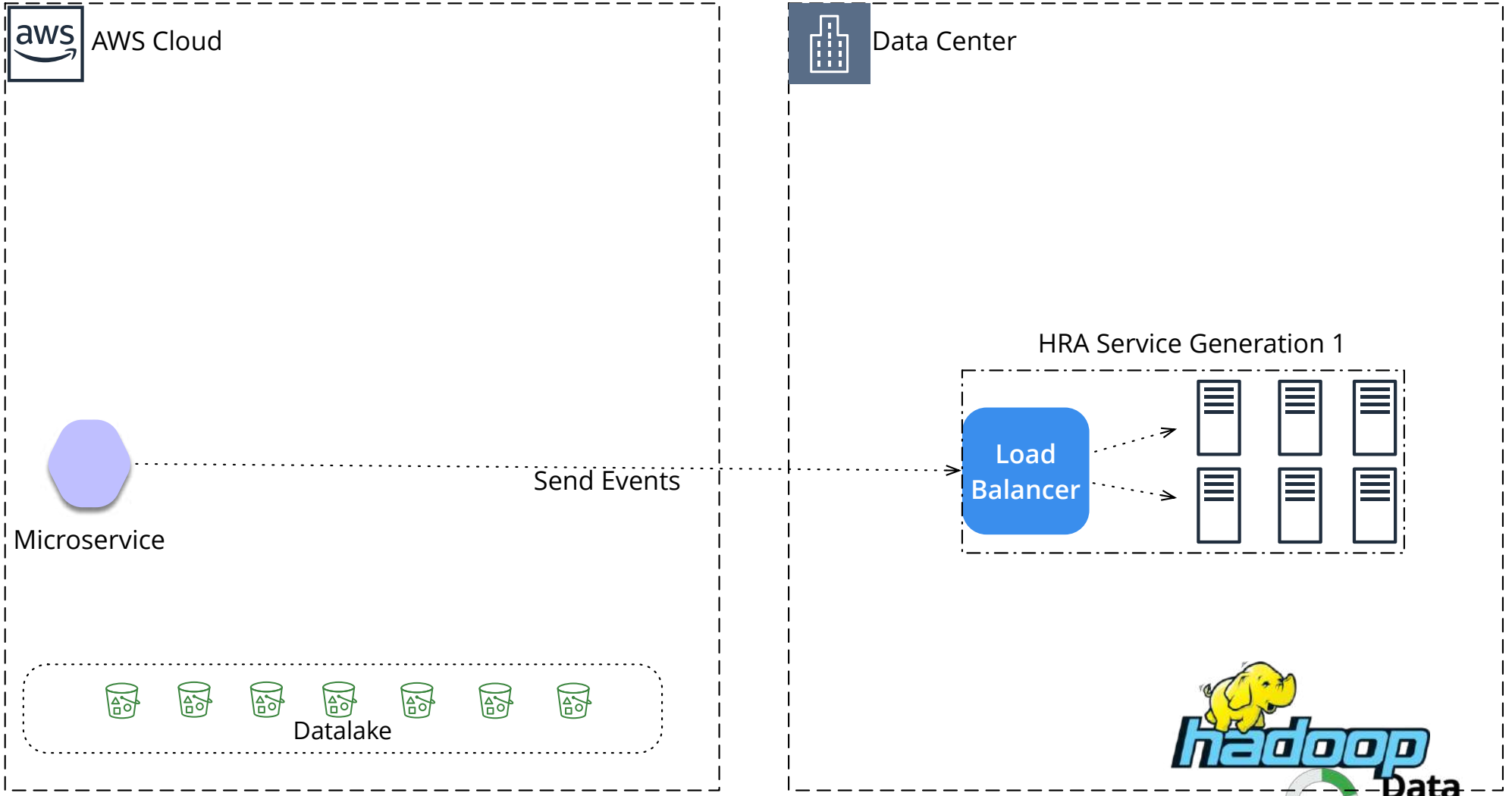
Data Platform

SCOUT 24

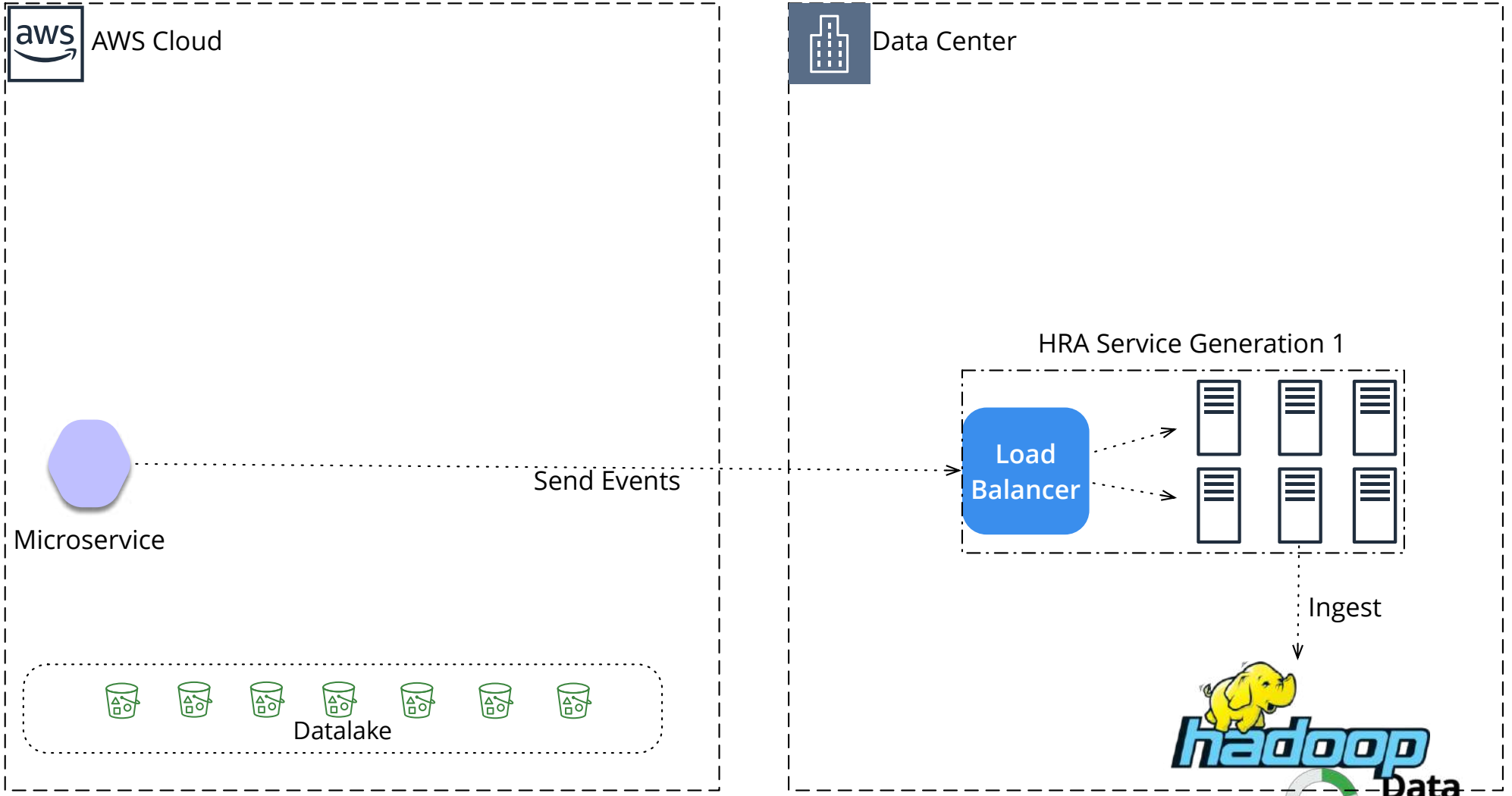
Hadoop Rest API – Architecture Evolution (Phase 1)



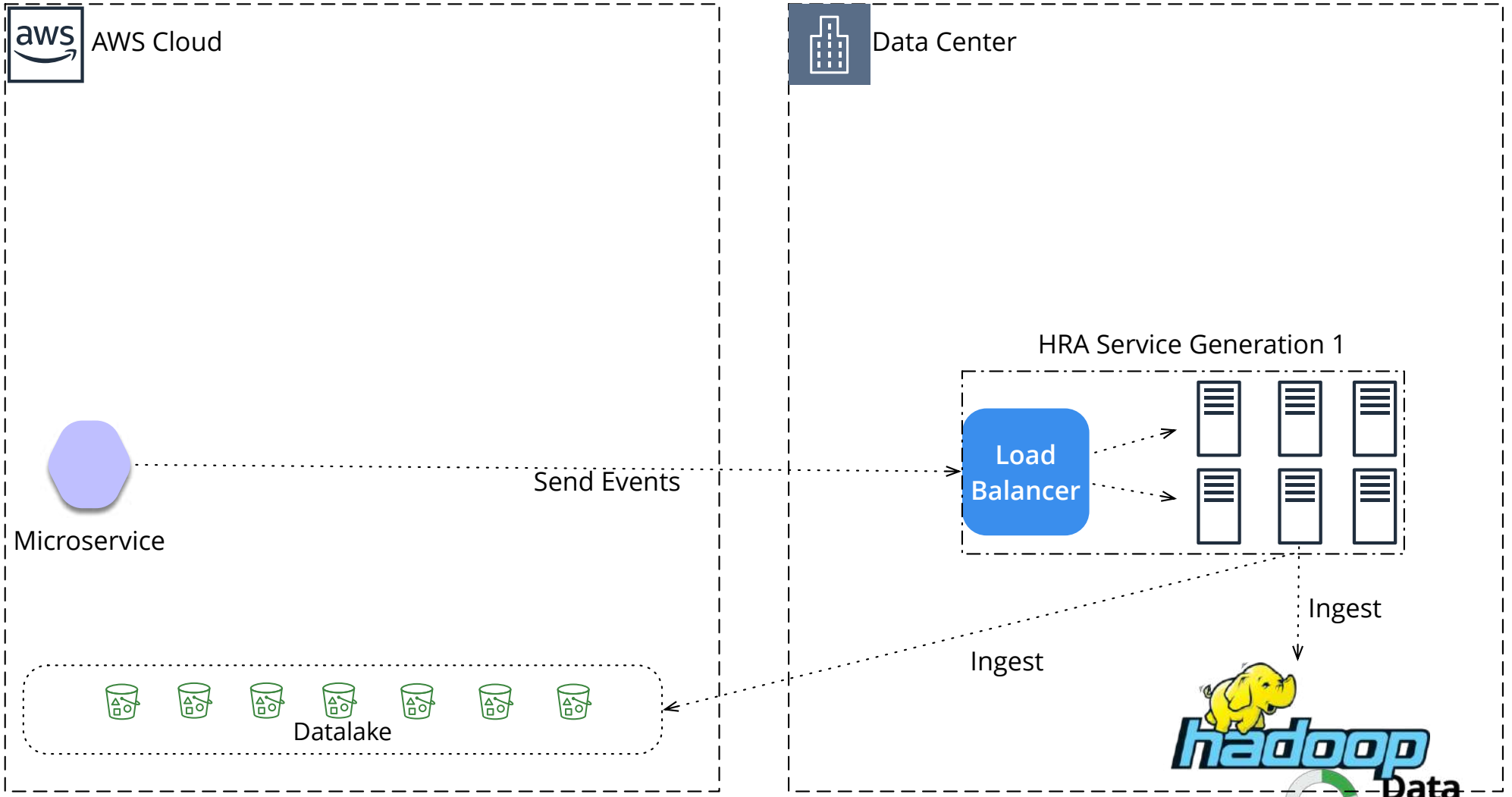
Hadoop Rest API – Architecture Evolution (Phase 1)



Hadoop Rest API – Architecture Evolution (Phase 1)

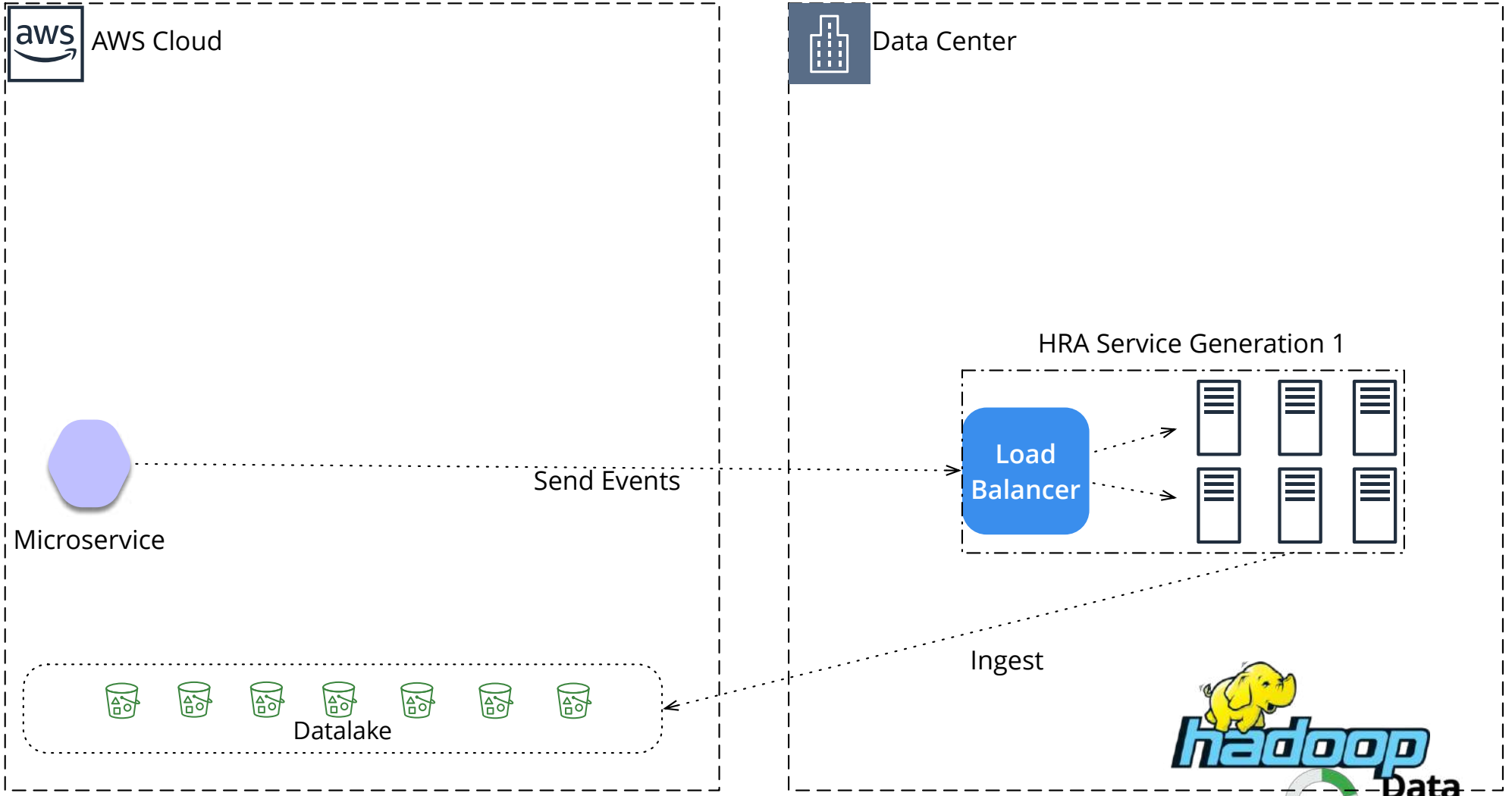


Hadoop Rest API – Architecture Evolution (Phase 2)

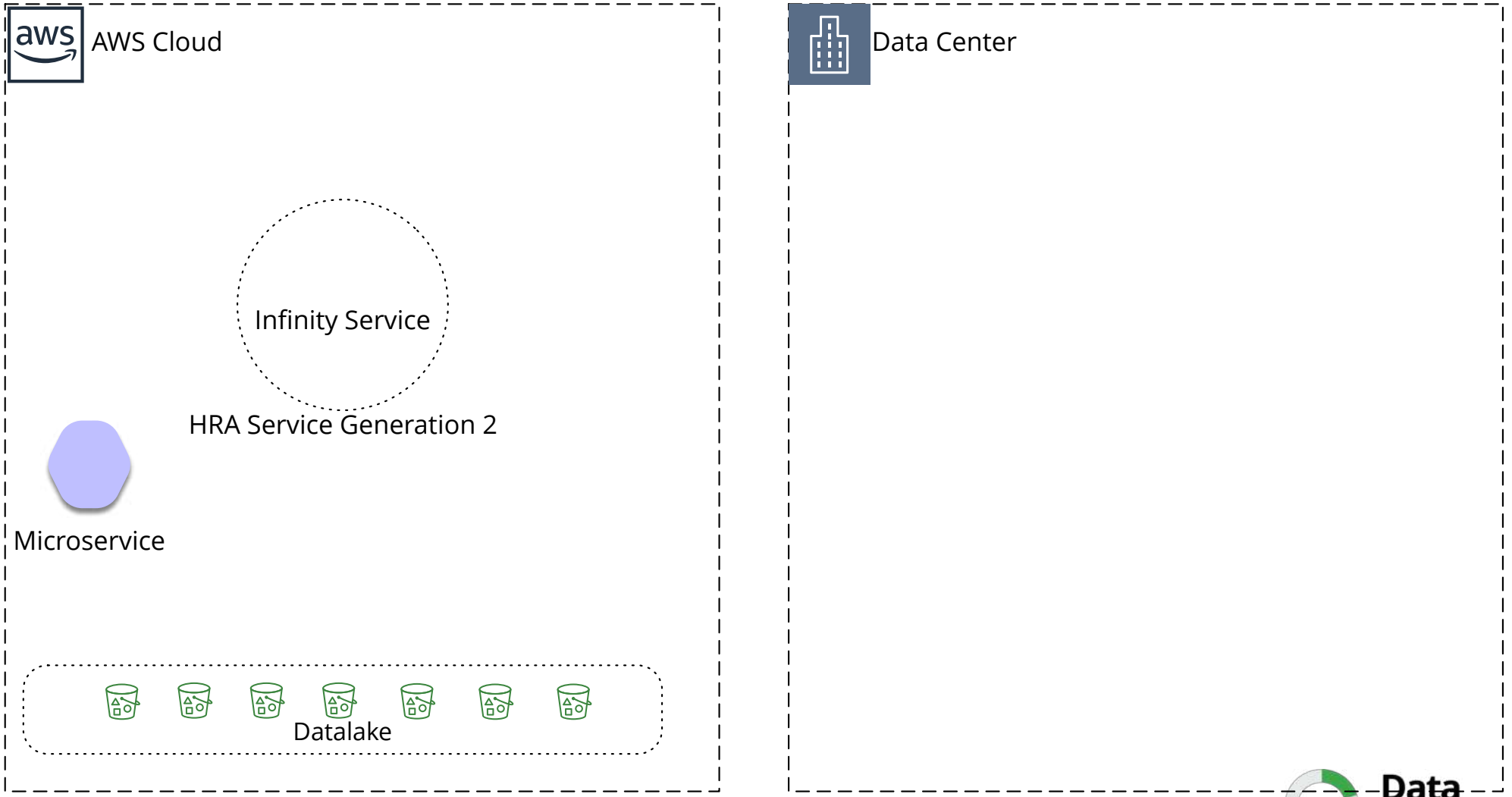


SCOUT 24

Hadoop Rest API – Architecture Evolution (Phase 2)



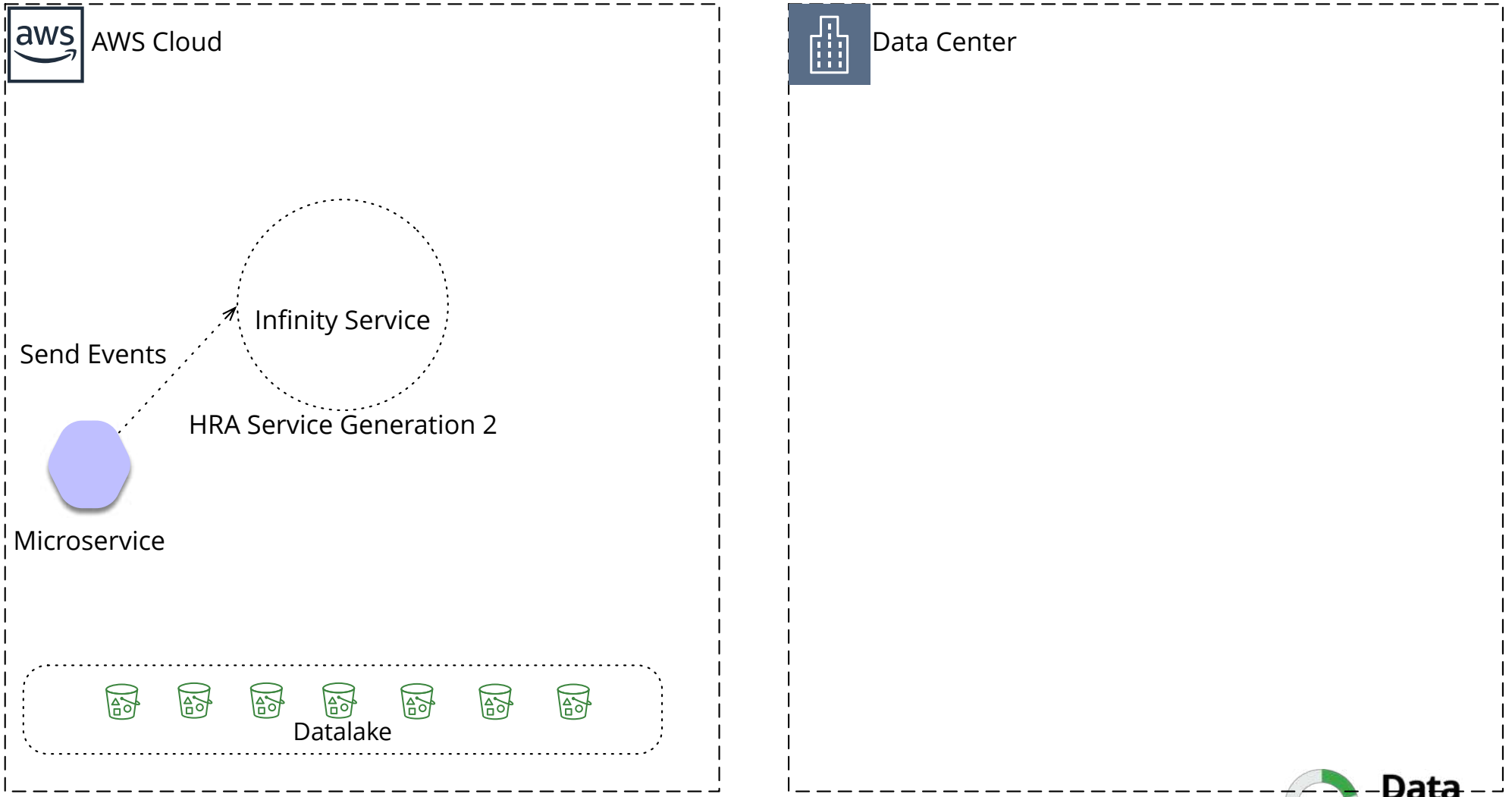
Hadoop Rest API – Architecture Evolution (Phase 3)



Data Platform

SCOUT 24

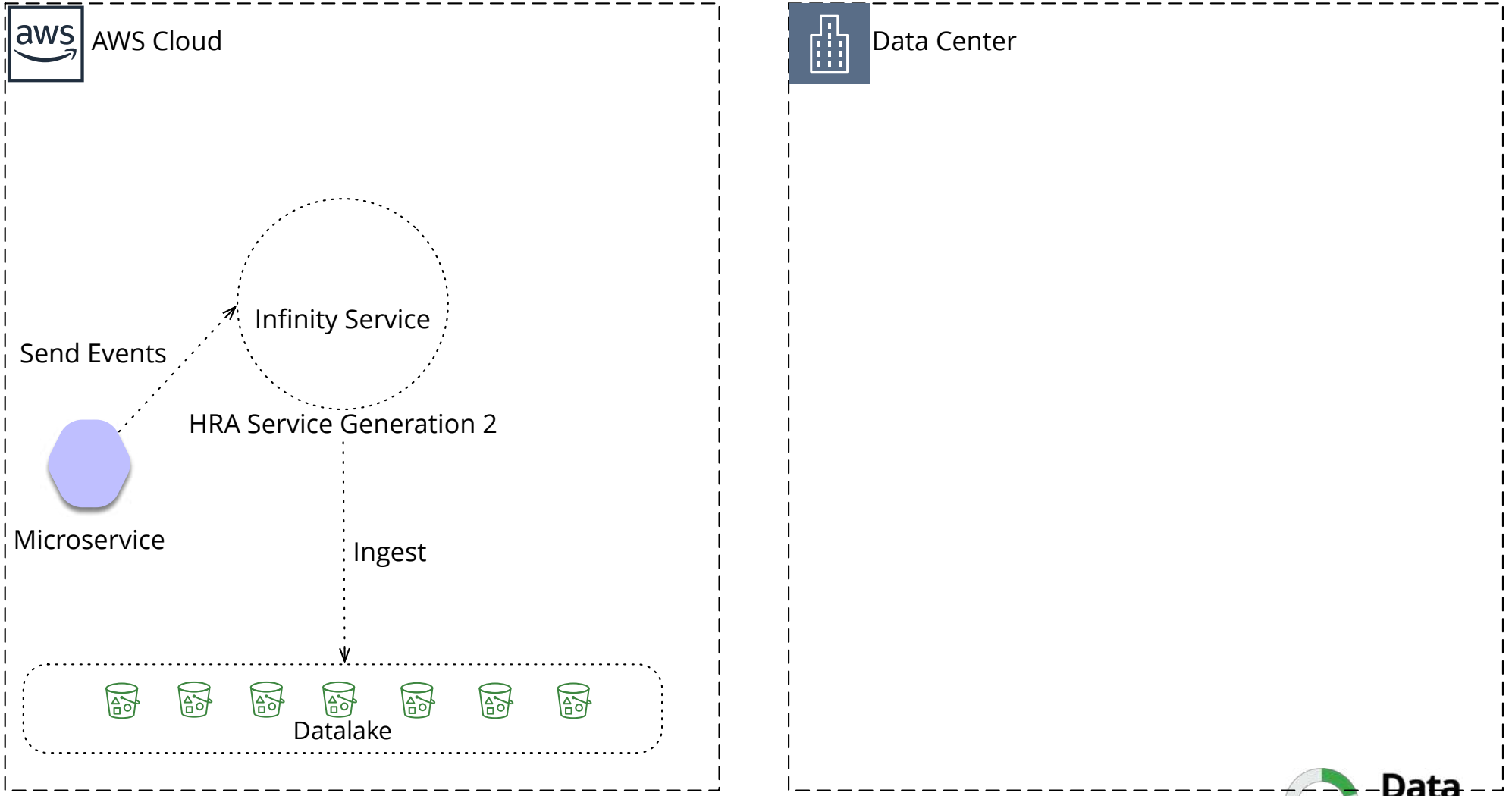
Hadoop Rest API – Architecture Evolution (Phase 3)



Data Platform

SCOUT 24

Hadoop Rest API – Architecture Evolution (Phase 3)



Data Platform

SCOUT 24

Self-Service ETL

Data
Platform

SCOUT 24

Self Service ETL Requirements

Data Landscape Manifesto Principle #5: Data consumers are responsible for the definition and visualization of metrics and for driving the implementation and maintenance of these metrics.

➔Data Consumers get a lot responsibility

➔The Data Platform must empower users to fulfill this responsibility

➔It must be easy to develop, test and maintain data processes

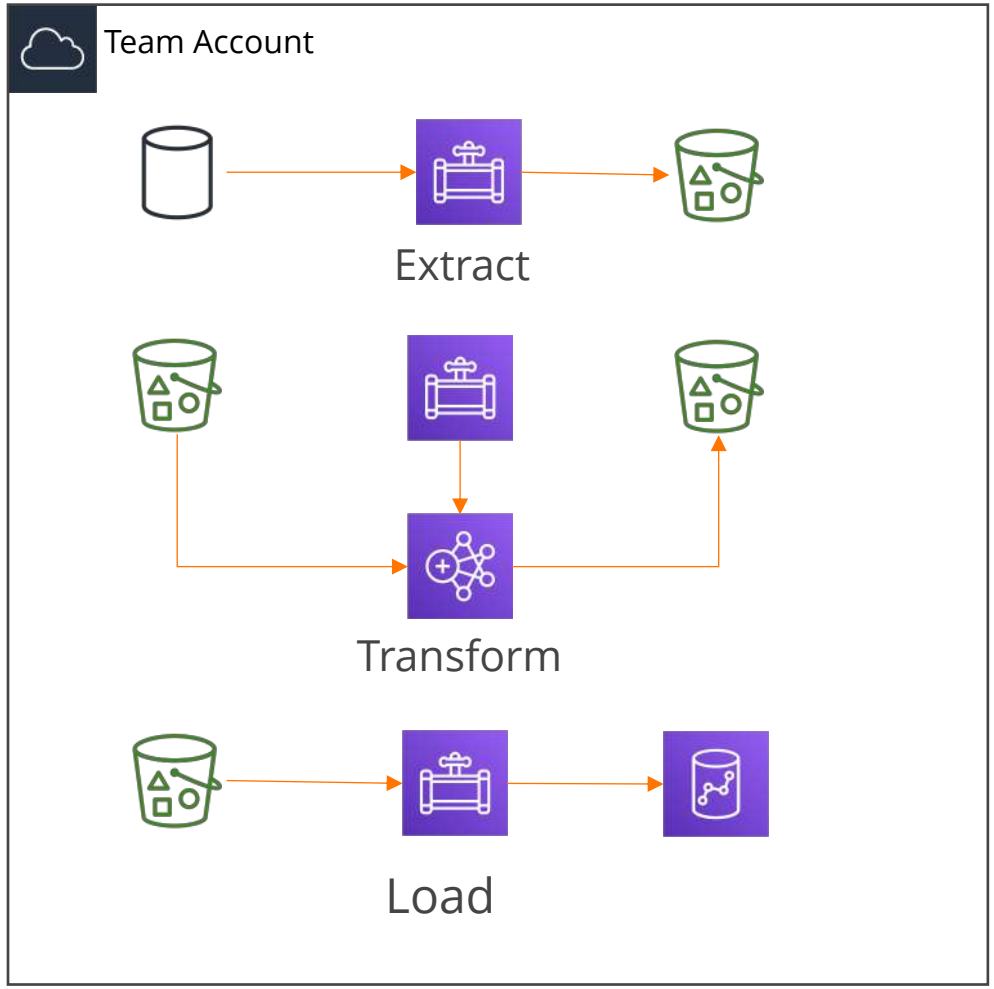
Multi-Account setup: Data processes must run in data consumer's account

➔No centrally run processing tool

➔AWS Managed service is preferred

Data Platform SCOUT 24

AWS Data Pipeline

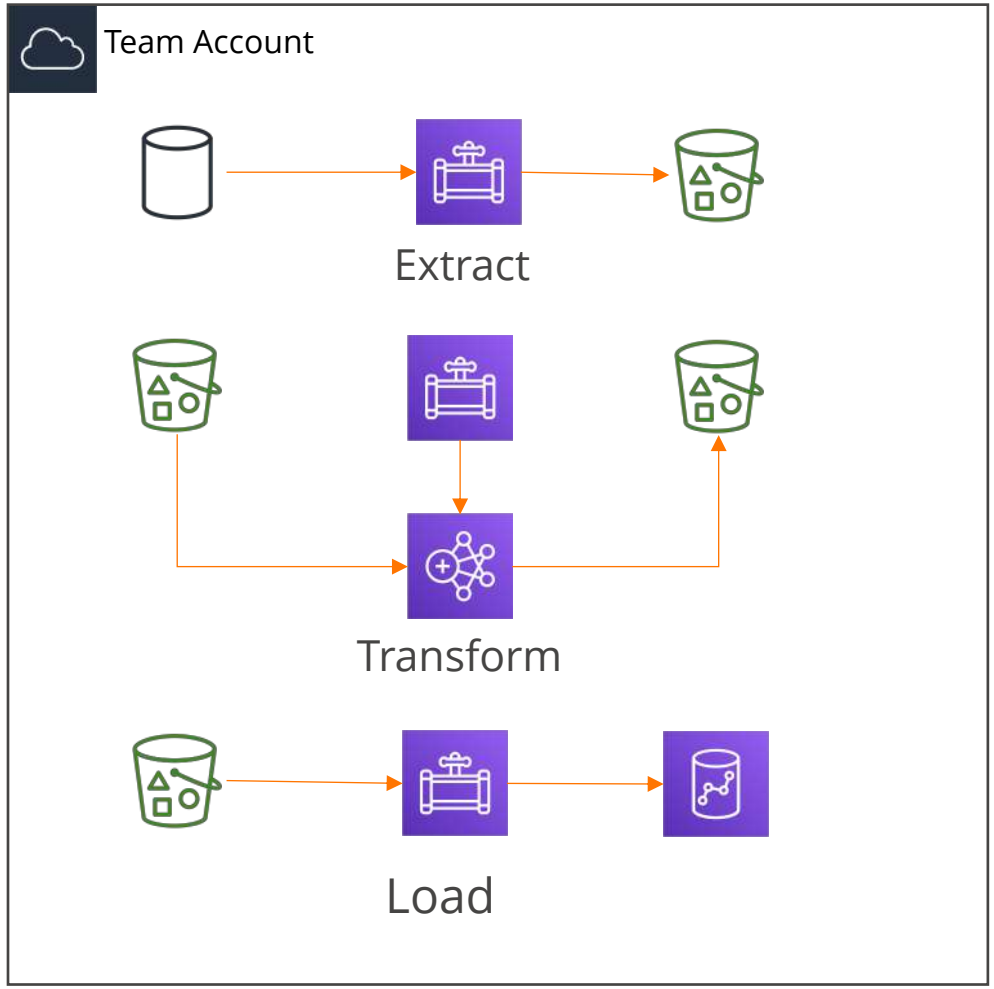


ETL runs in team account

Data Platform

SCOUT 24

AWS Data Pipeline



Teams run ETL in their accounts

It's easy .... Or is it?

Data Platform

SCOUT 24

```
{
    "objects": [
      {
        "schedule": {
          "ref": "DefaultSchedule"
        },
        "directoryPath": "s3://sourceBucket",
        "name": "SourceS3DataNode",
        "id": "DataNodeId_yIW0k",
        "type": "S3DataNode"
      },
      {
        "period": "1 days",
        "name": "Every 1 day",
        "id": "DefaultSchedule",
        "type": "Schedule",
        "startAt": "FIRST_ACTIVATION_DATE_TIME"
      },
      {
        "output": {
          "ref": "RedshiftDataNodeId_sAlcV"
        },
        "input": {
          "ref": "S3DataNodeId_VJmwh"
        },
        "schedule": {
          "ref": "DefaultSchedule"
        },
        "name": "Load",
        "id": "CopyActivityId_jPabN",
        "runsOn": {
          "ref": "ResourceId_YzpFO"
        },
        "type": "RedshiftCopyActivity",
        "insertMode": "OVERWRITE_EXISTING"
      },
      {
        "schedule": {
          "ref": "DefaultSchedule"
        },
        "resourceRole": "DataPipelineDefaultResourceRole",
        "role": "DataPipelineDefaultRole",
        "name": "Ec2Instance",
        "id": "ResourceId_YzpFO",
        "type": "Ec2Resource"
      },
      {
        "schedule": {
          "ref": "DefaultSchedule"
        },
        "database": {
          "ref": "DatabaseId_nwLUx"
        },
        "name": "SourceSqlDataNode",
        "id": "SqlDataNodeId_nhC9I",
        "type": "SqlDataNode",
        "table": "sourceTable"
      },
      {
        "output": {
          "ref": "DataNodeId_yIW0k"
        },
        "input": {
          "ref": "SqlDataNodeId_nhC9I"
        },
        "schedule": {
          "ref": "DefaultSchedule"
        },
        "name": "Extract",
        "workerGroup": "local-worker-1",
        "id": "CopyActivityId_OyKN4",
        "type": "CopyActivity"
      },
      {
        "schedule": {
          "ref": "DefaultSchedule"
        },
        "name": "EmrResource",
        "releaseLabel": "emr-5.21.0",
        "id": "ResourceId_tBzz1",
        "type": "EmrCluster"
      },
      {
        "schedule": {
          "ref": "DefaultSchedule"
        },
        "name": "Transform",
        "runsOn": {
          "ref": "ResourceId_tBzz1"
        },
        "id": "ShellCommandActivityId_NM2Ga",
        "type": "ShellCommandActivity",
        "command": "echo \"Transform\""
      },
      {
        "failureAndRerunMode": "CASCADE",
        "schedule": {
          "ref": "DefaultSchedule"
        },
        "resourceRole": "DataPipelineDefaultResourceRole",
        "role": "DataPipelineDefaultRole",
        "scheduleType": "cron",
        "name": "Default",
        "id": "Default"
      },
      {
        "schedule": {
          "ref": "DefaultSchedule"
        },
        "database": {
          "ref": "RedshiftDatabaseId_1IMYz"
        },
        "name": "TargetRedshiftDataNode",
        "id": "RedshiftDataNodeId_sAlcV",
        "type": "RedshiftDataNode",
        "tableName": "targettable"
      },
      {
        "schedule": {
          "ref": "DefaultSchedule"
        },
        "directoryPath": "s3://target-bucket",
        "name": "TargetS3DataNode",
        "id": "S3DataNodeId_VJmwh",
        "type": "S3DataNode"
      },
      {
        "connectionString": "jdbc:oracle:thin:/host/db",
        "*password": "geheim",
        "name": "SourceDB",
        "id": "DatabaseId_nwLUx",
        "type": "JdbcDatabase",
        "jdbcDriverClass": "com.oracle.Driver",
        "username": "user"
      },
      {
        "connectionString": "jdbc:redshift:/cluster//",
        "*password": "passwo
        "name": "TargetDB",
        "id": "RedshiftDatab
        "type": "RedshiftDat
        "username": "user"
      }
    ],
    "parameters": []
}
```

DataWario to the rescue

```yaml
emr:
ec2:
databases:
  sourceDb:
    driverClass: oracle.jdbc.OracleDriver
    driverUri: ojdbc6.jar
    user: aws_export
    password: wonttell
    url: jdbc:oracle:thin:@//myDB
  targetDb:
    type: redshift
    clusterId: redshift-cluster-id
    user: redshift-user
    password: redshift-password
    dbName: cloud_dwh
steps:
  - database_to_s3:
      dbRef: sourceDb
      format: csv
      selectQuery: select 1 from dual
      s3Url: s3://someBucket/some.csv
  - shell:
      command: "aws s3 cp --recursive s3://someBucket/some.csv s3://someOtherBucket/some.csv"
      runsOn: emr
  - s3_to_database:
      runsOn: ec2
      dbRef: targetDb
      insertMode: OVERWRITE_EXISTING
      commandOptions:
          - delimiter '\t'
          - gzip
          - truncatecolumns
          - maxerror 1000
          - blanksasnull
      table: thetable
      schema: myschema
      s3Url: s3://someOtherBucket/some.csv
```

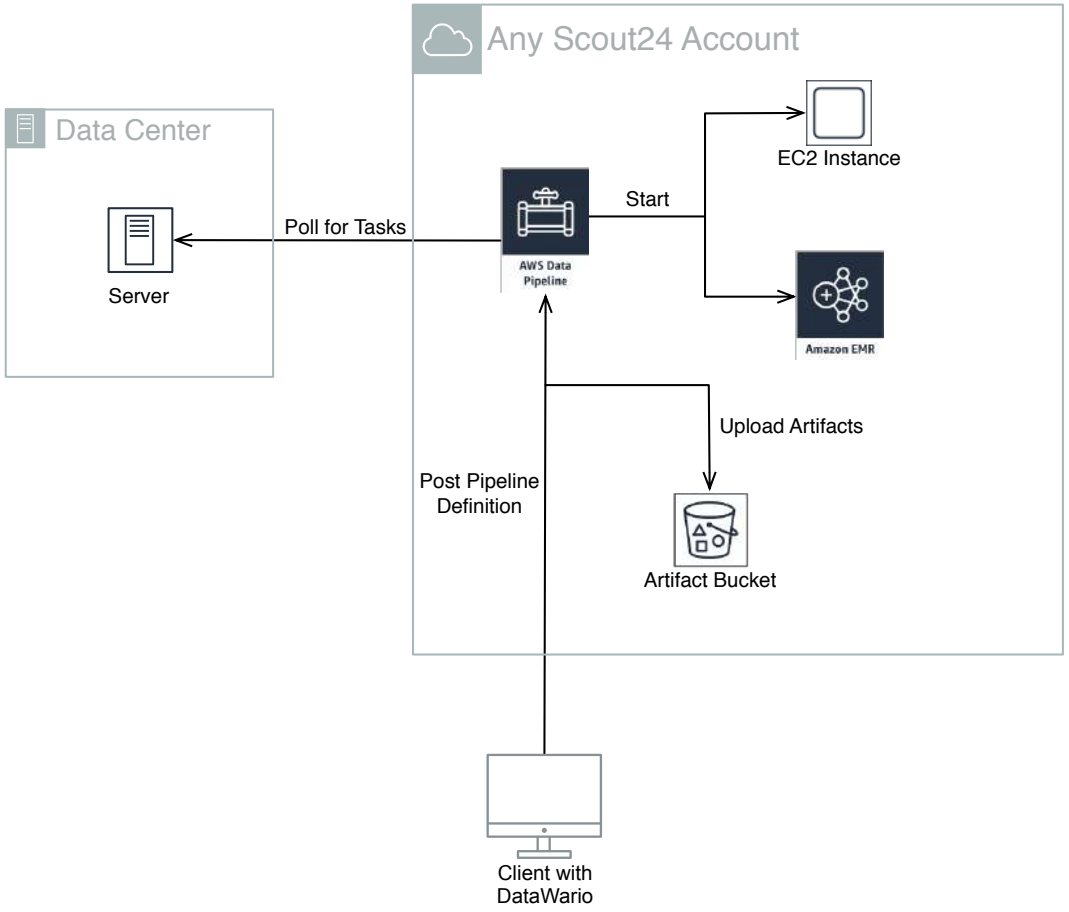
Data Platform

SCOUT 24

DataWario

1. Client for AWS Data Pipeline
2. Transforms YAML into JSON format understood by Data Pipeline
3. Integrates with the Scout24 account setup and sets sensible configuration defaults
4. Augments Data Pipeline with additional features
5. Makes testing and debugging faster

DataWario – Architecture



- Java Application, distributed as a JAR
- Shell wrapper (`dw`) for easy usage from the command line

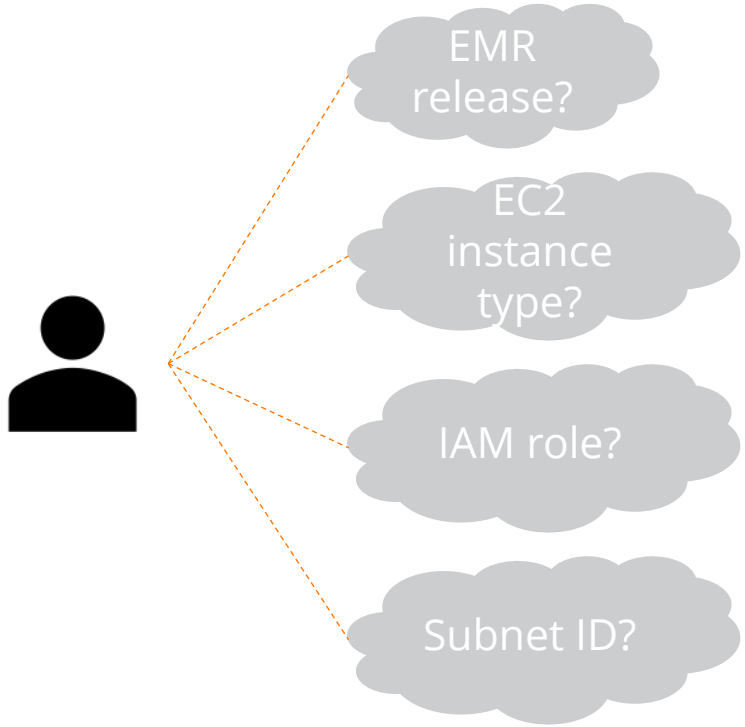
Deployment of a pipeline

1. User issues `dw deploy`
2. DataWario
    1. uploads artifacts from user's machine to S3
    2. generates JSON definition from YAML
    3. creates data pipeline
3. AWS Data Pipeline runs the defined steps

Data Platform

SCOUT 24

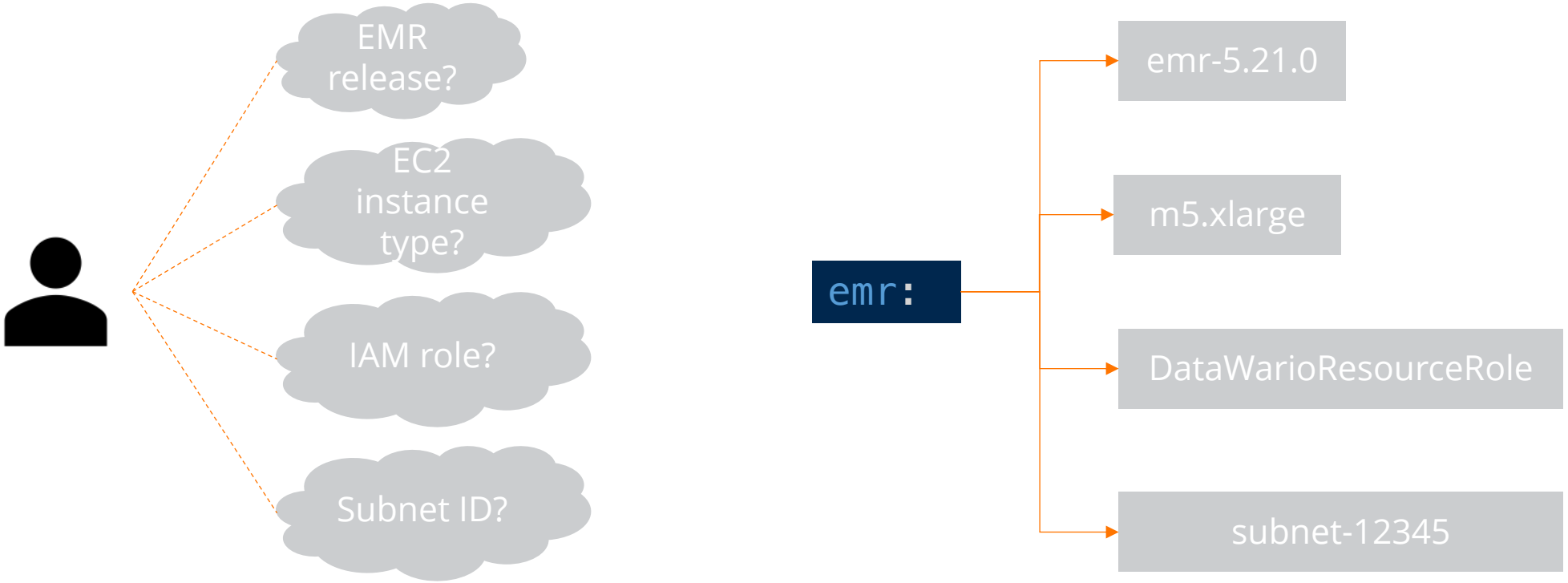
Sensible defaults



Data Platform

SCOUT 24

Sensible defaults



Data Platform

SCOUT 24

Augmenting data pipeline

ID	Make	Model	Price	Mileage
1	VW	Golf	22000	15000
2	BMW	320d	35000	40000

„Built-In“ Copy Activity

```
1,VW,Golf,22000,15000

2,BMW,320d,35000,40000
```

- CSV (or TSV)
- No schema

Data Platform SCOUT 24

Augmenting data pipeline

ID	Make	Model	Price	Mileage
1	VW	Golf	22000	15000
2	BMW	320d	35000	40000

„Built-In“ Copy Activity

```
1,VW,Golf,22000,15000
2,BMW,320d,35000,40000
```

- CSV (or TSV)
- No schema

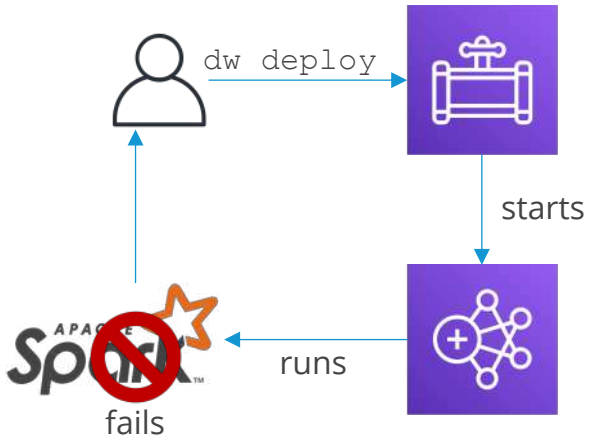
„Custom“ Copy Activity

+

```
{
    "fields": [
        {
            "name": "id",
            "type": "long"
        },
        {
            "name": "make",
            "type": "string"
        },
        {
            "name": "model",
            "type": "string"
        },
        {
            "name": "price",
            "type": "double"
        },
        {
            "name": "mileage",
            "type": "double"
        }
    ]
}
```

Speed up pipeline development

Typical development cycle



takes ca. 10 min + job runtime

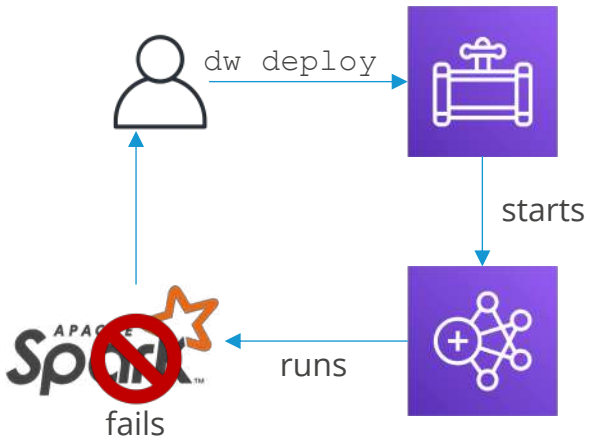
Data Platform

SCOUT 24

Speed up test and debug cycle

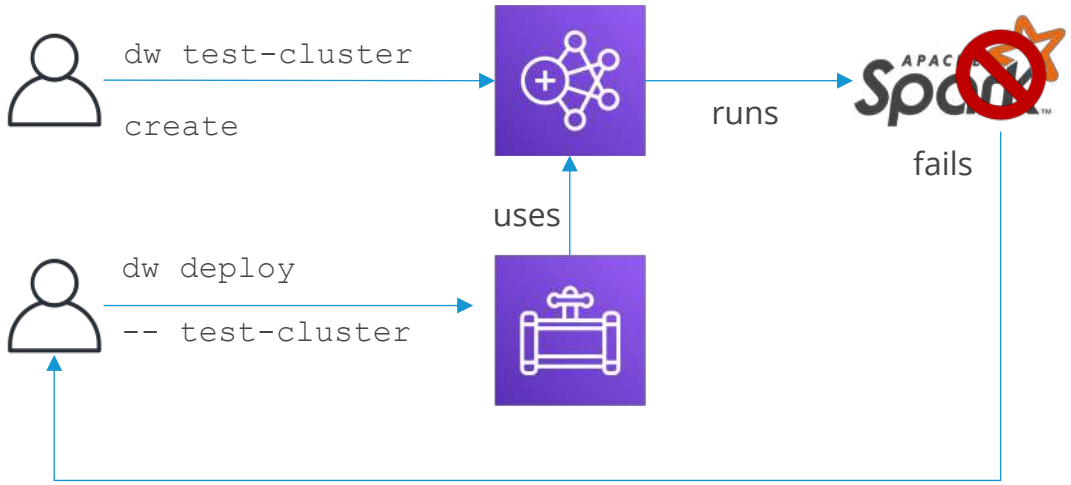
Typical development cycle

Without test cluster



takes ca. 10 min + job runtime

With test cluster



takes job runtime

Data Platform

SCOUT 24

DataWario brought us from here ...

photo by Dirk van der Made (https://commons.wikimedia.org/wiki/File:DirkvdM_cloudforest-jungle.jpg), „DirkvdM cloudforest-jungle“, https://creativecommons.org/licenses/by-sa/3.0/legalcode

Data Platform

SCOUT 24

to here

So, how do we get here?

Data Platform

SCOUT 24

Paving the path further

- We built a lot of pipelines, so did our users
- We see repetitive tasks emerge in these pipelines
- Reimplementing solutions over and over again is waste
- We don‘t like waste

➔ Let‘s build reusable tools for common tasks

Much of the power of the UNIX operating system comes from a style of program design that makes programs easy to use and, more important, **easy to combine with other programs**. This style has been called the use of *software tools*, and depends more on how the programs fit into the programming environment and how they can be used with other programs than on how they are designed internally. [...] This style was based on the use of **tools**: **using programs separately or in combination to get a job done**, rather than doing it by hand, by monolithic self-sufficient subsystems, or by special-purpose, one-time programs.

—

Rob Pike; Brian W. Kernighan (October 1984). "Program Design in the UNIX Environment"

Tools that do one thing, and do that well

- Advantages:
  - Easy to configure, only a few settings
  - Easy to test, because there are less branches to cover
  - Easy to debug, output of each tool can be checked independently

- Drawbacks:
  - More intermittent data written
  - longer runtimes

Event snapshotter

- Typical questions: What was the headline of a listing on Feb 22, 2016? Was it published? How about March 1?
- Input data: Change events for an entity (e.g. listing)
  - User changed headline of a listing, number of pictures
  - Listing got published/unpublished
- Snapshotter
  1. takes snapshot from the day before
  2. merges all change events from the day, keeps last
  3. writes new snapshot for day
- Simple config (input and output path, name of ID and timestamp column, output path, partition schema)
- No business logic, agnostic of type of entity

Event Snapshotter Example

```
{"id": 1, "changed": "2019-01-01T15:31:00.000+01:00", "headline": „nice appartment", "published": false}
{"id": 1, "changed": "2019-01-01T18:31:00.000+01:00", "headline": "A very nice appartment", "published": false}
{"id": 1, "changed": "2019-01-02T06:31:00.000+01:00", "headline": "A very nice appartment", "published": true}
{"id": 2, "changed": "2019-01-02T07:28:00.000+01:00", "headline": "Another nice appartment", "published": true}
```

January 1, 2019 (s3://snapshotBucket/snapshot_day=2019-01-01/):

```
{"id": 1, "changed": "2019-01-01T18:31:00.000+01:00", "headline": "A very nice appartment", "published": false}
```

January 2, 2019 (s3://snapshotBucket/snapshot_day=2019-01-02/):

```
{"id": 1, "changed": "2019-01-02T06:31:00.000+01:00", "headline": "A very nice appartment", "published": true}
{"id": 2, "changed": "2019-01-02T07:28:00.000+01:00", "headline": "Another nice appartment", "published": true}
```

Aggregator

- Typical Questions:
  - How many published listings did we have on Feb 1 2018?
  - What‘s the daily average number of pageviews per car make?
- Input data:
  - Interaction events of an entity (e.g. page viewed, email sent, number called)
  - output of Event Snapshotter
- Simple config (input and output paths, timestamp column, grouping column, aggregation method and column)

Spark Utils

- Collection of simple tools (usually one class per tool)
- Can be used
  - standalone in a pipeline step
  - as a dependency of your code
- Examples:
  - Create a Hive table on-top of existing data
  - Convert data from one Format to another (e.g. from JSON to Parquet for faster querying)
  - Flatten nested structures
  - publish data quality metrics to CloudWatch
  - Materializing views

Current State / Outlook

- Data Pipeline with DataWario widely used throughout the company
- Challenges:
  - Data Pipeline not getting much love from AWS (not deprecated, but not getting many new features)
  - Coordination between data pipelines only rudimentary
- Possible solutions:
  - AWS Step Functions in combination with AWS Glue
  - Introduction of a Workflow tool (e.g. AirFlow)

Self-Service Analytics

Data
Platform

SCOUT 24

Query Challenges

What's Ahead

Unlock the Datalake for Scout24's Toolset and Users with Different Skillsets

Data Analysis for Various User Groups

Provide a Timely and Accurate Update of the Metadata Layer

Data Platform

SCOUT 24

What's Ahead

Unlock the Datalake for Scout24's Toolset and Users with Different Skillsets

OneScout Hive Metastore

Data Analysis for Various User Groups

Provide a Timely and Accurate Update of the Metadata Layer

Data Platform

SCOUT 24

What's Ahead

Unlock the Datalake for Scout24's Toolset and Users with Different Skillsets

OneScout Hive Metastore

Data Analysis for Various User Groups

Personal Analytics Cluster

Provide a Timely and Accurate Update of the Metadata Layer

Data Platform

SCOUT 24

What's Ahead

Unlock the Datalake for Scout24's Toolset and Users with Different Skillsets

OneScout Hive Metastore

Data Analysis for Various User Groups

Personal Analytics Cluster

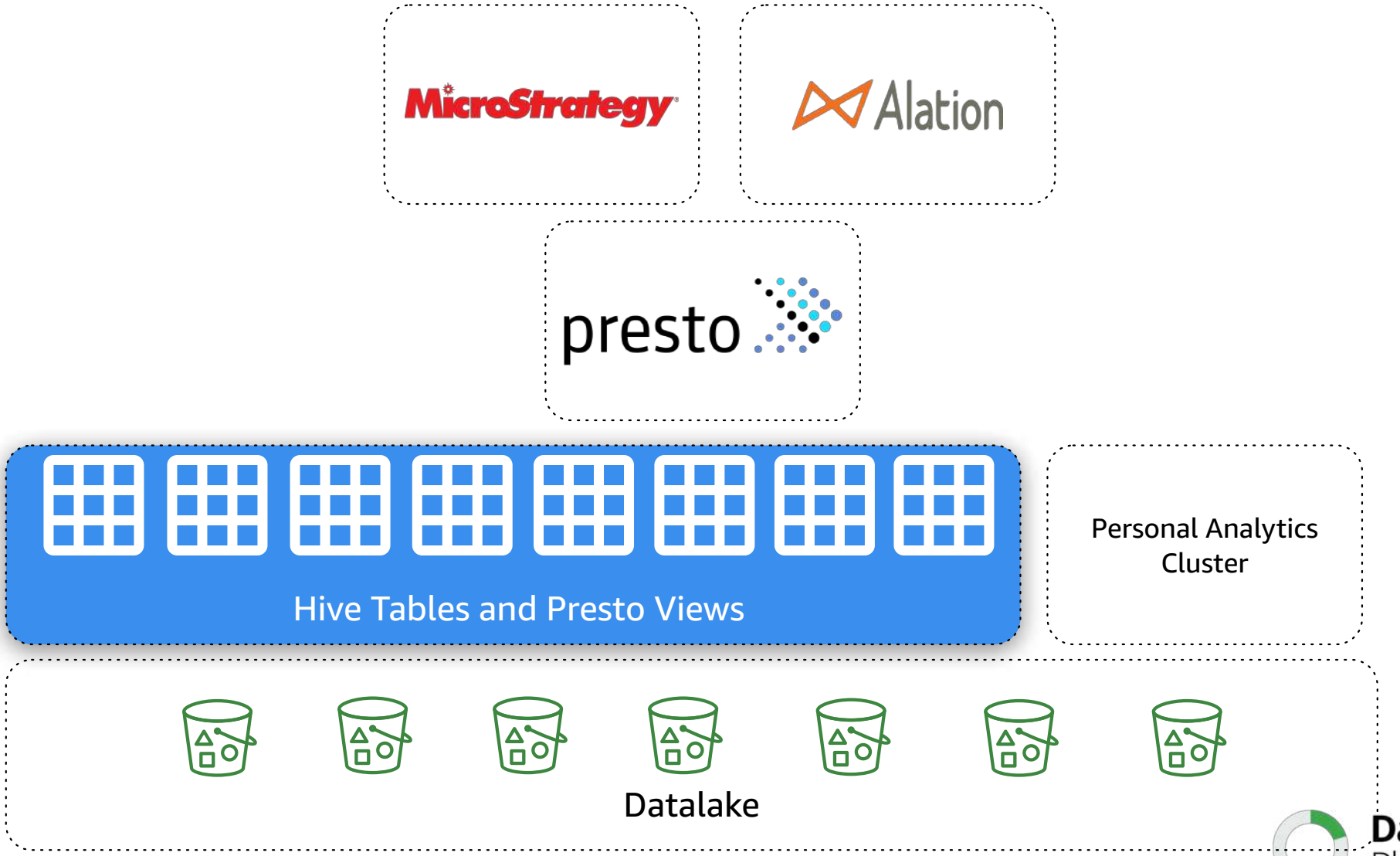
Provide a Timely and Accurate Update of the Metadata Layer

Automatic Hive Partition Detection

Data Platform

SCOUT 24

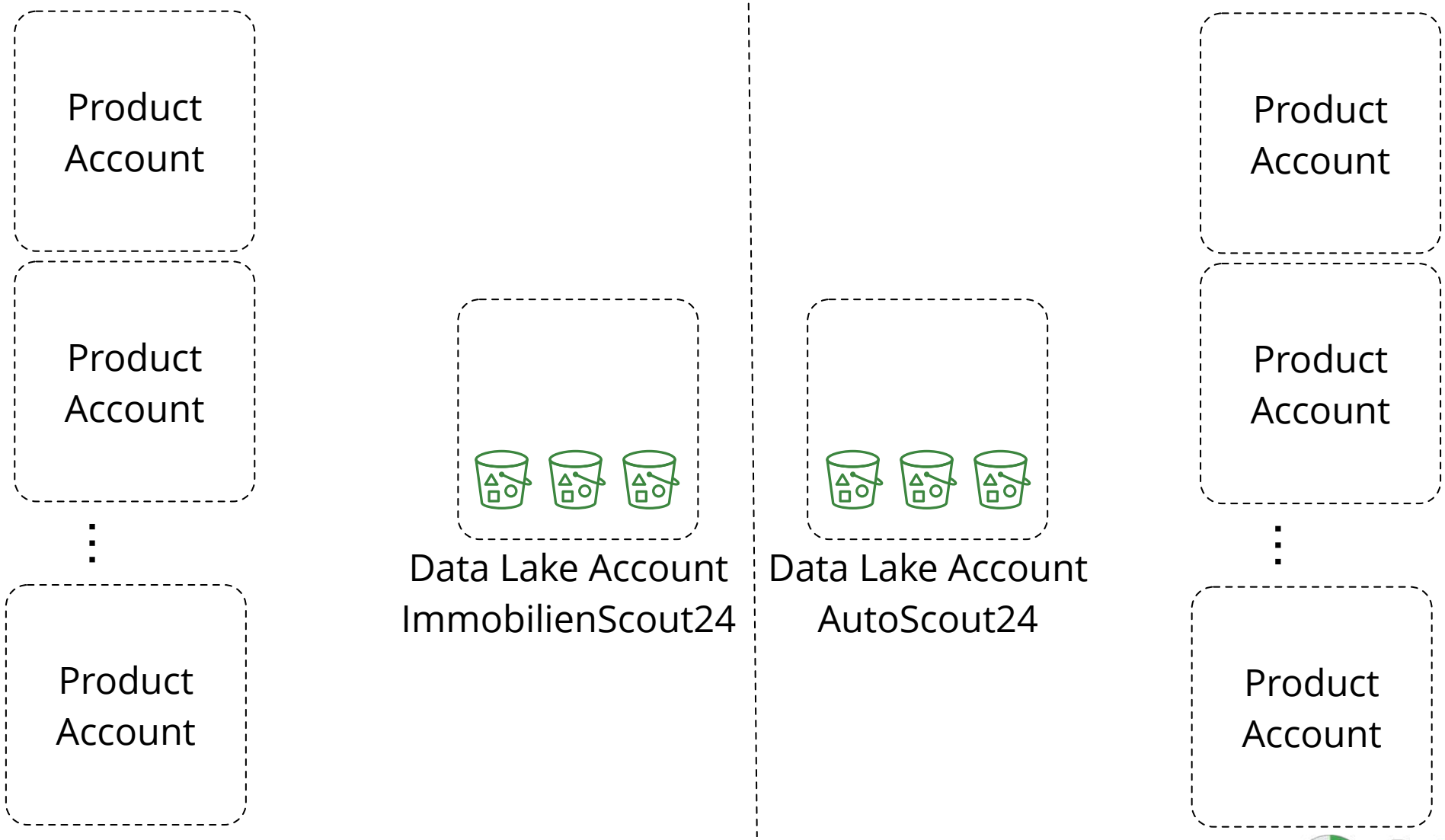
OneScout Hive Metastore – A Schematic View



Data Platform

SCOUT 24

OneScout Hive Metastore – Recap of Ecosystem



Data Platform

SCOUT 24

The Scout24 Hive Metastore Proxy – A Motivation

Goal: One Long-Running Metastore DB every EMR connects to

Ideal Situation

Solution for Multi-Account-Setting

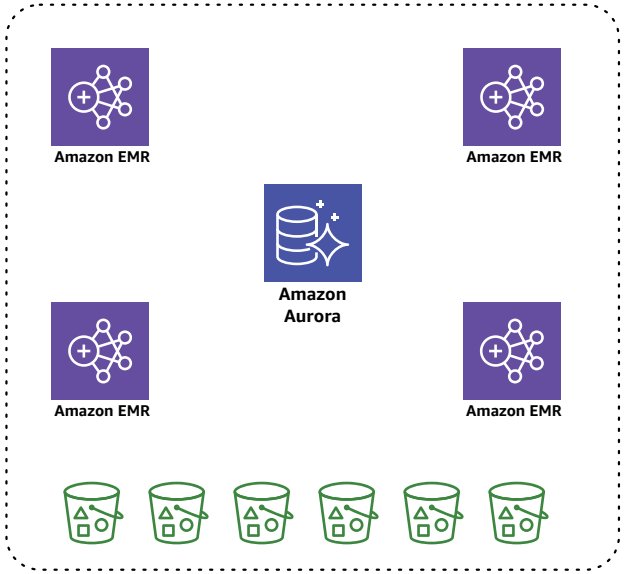
Data Platform

SCOUT 24

The Scout24 Hive Metastore Proxy – A Motivation

Goal: One Long-Running Metastore DB every EMR connects to

Ideal Situation



Central Datalake Account

Solution for Multi-Account-Setting

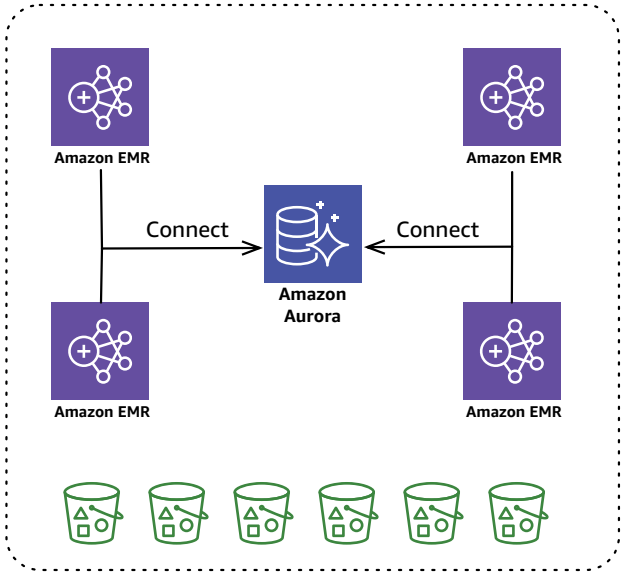
Data Platform

SCOUT 24

The Scout24 Hive Metastore Proxy – A Motivation

Goal: One Long-Running Metastore DB every EMR connects to

Ideal Situation



Central Datalake Account

Solution for Multi-Account-Setting

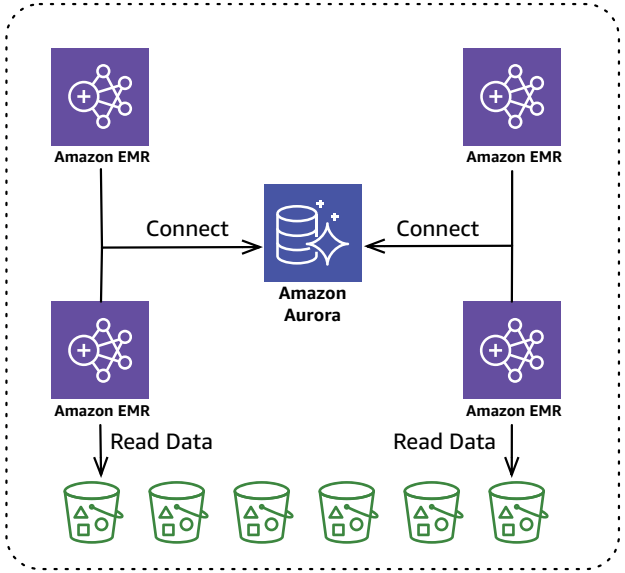
Data Platform

SCOUT 24

The Scout24 Hive Metastore Proxy – A Motivation

Goal: One Long-Running Metastore DB every EMR connects to

Ideal Situation



Central Datalake Account

Solution for Multi-Account-Setting

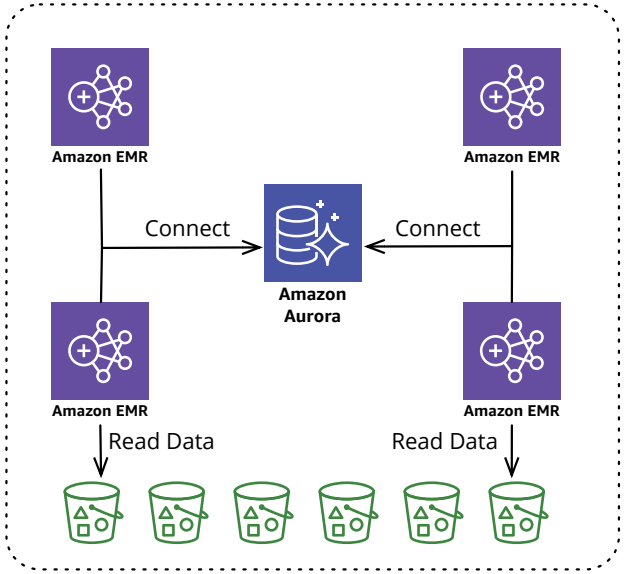
Data Platform

SCOUT 24

The Scout24 Hive Metastore Proxy – A Motivation

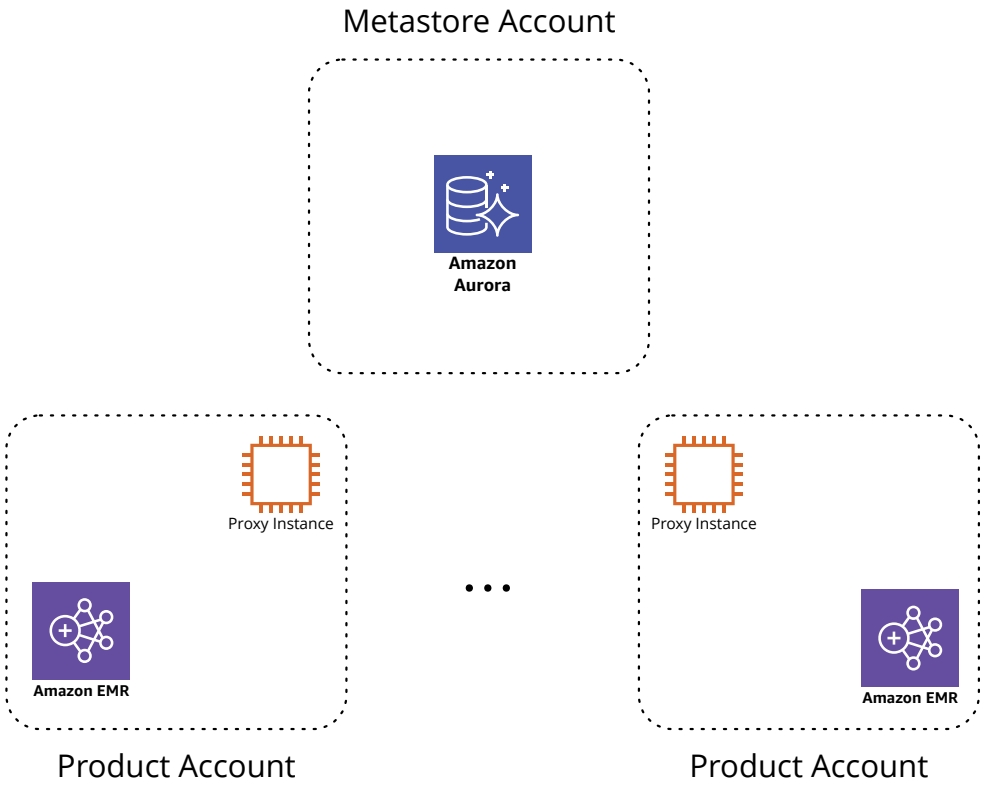
Goal: One Long-Running Metastore DB every EMR connects to

Ideal Situation



Central Datalake Account

Solution for Multi-Account-Setting



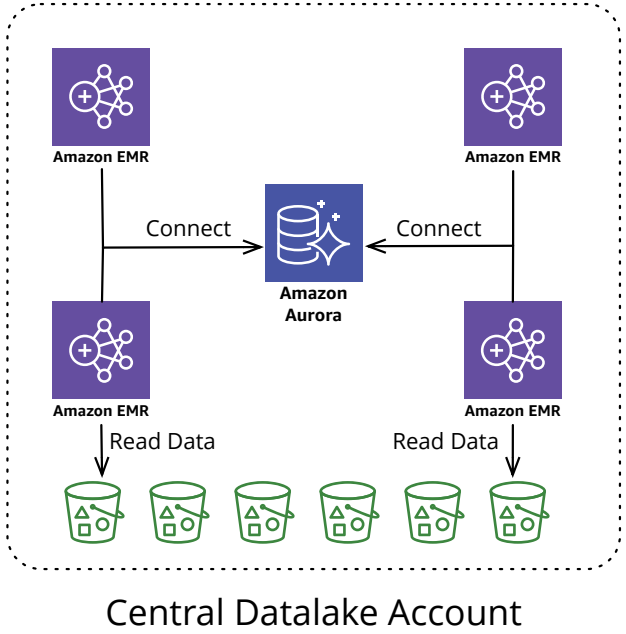
Data Platform

SCOUT 24

The Scout24 Hive Metastore Proxy – A Motivation

Goal: One Long-Running Metastore DB every EMR connects to

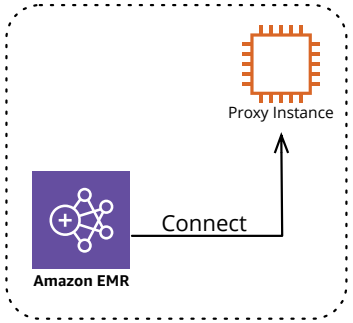
Ideal Situation



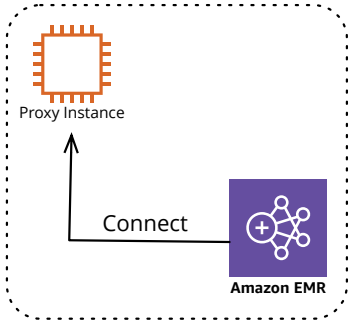
Central Datalake Account

Solution for Multi-Account-Setting

Metastore Account



Product Account



Product Account

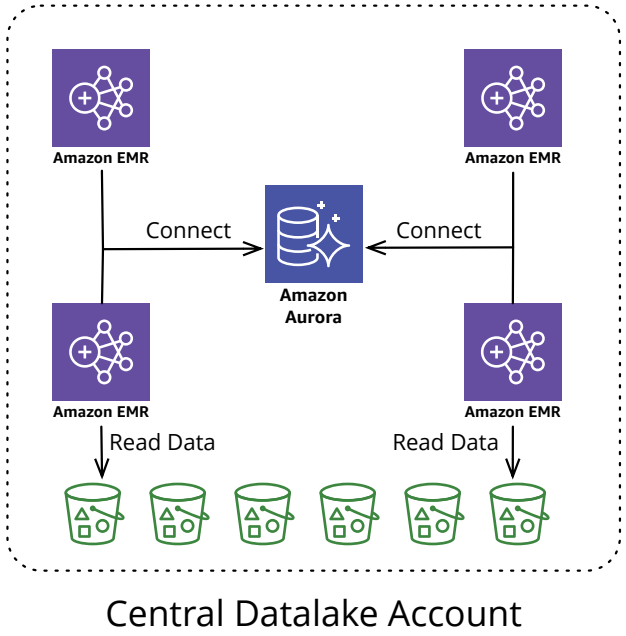
Data Platform

SCOUT 24

The Scout24 Hive Metastore Proxy – A Motivation

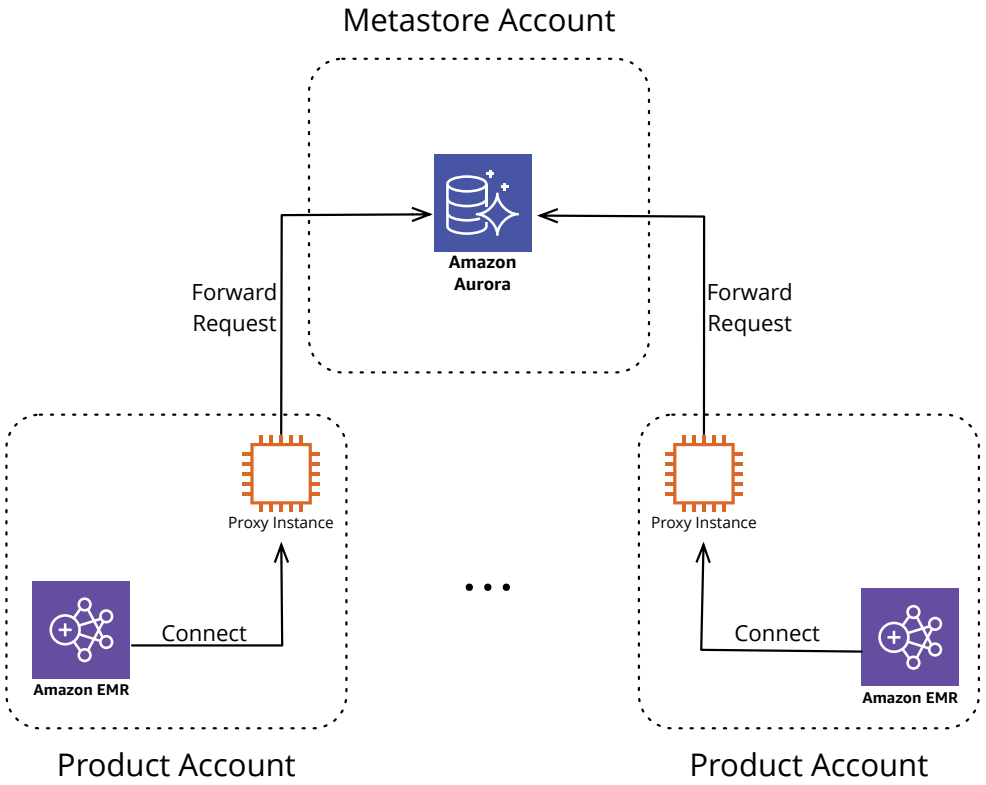
Goal: One Long-Running Metastore DB every EMR connects to

Ideal Situation



Central Datalake Account

Solution for Multi-Account-Setting



Data Platform

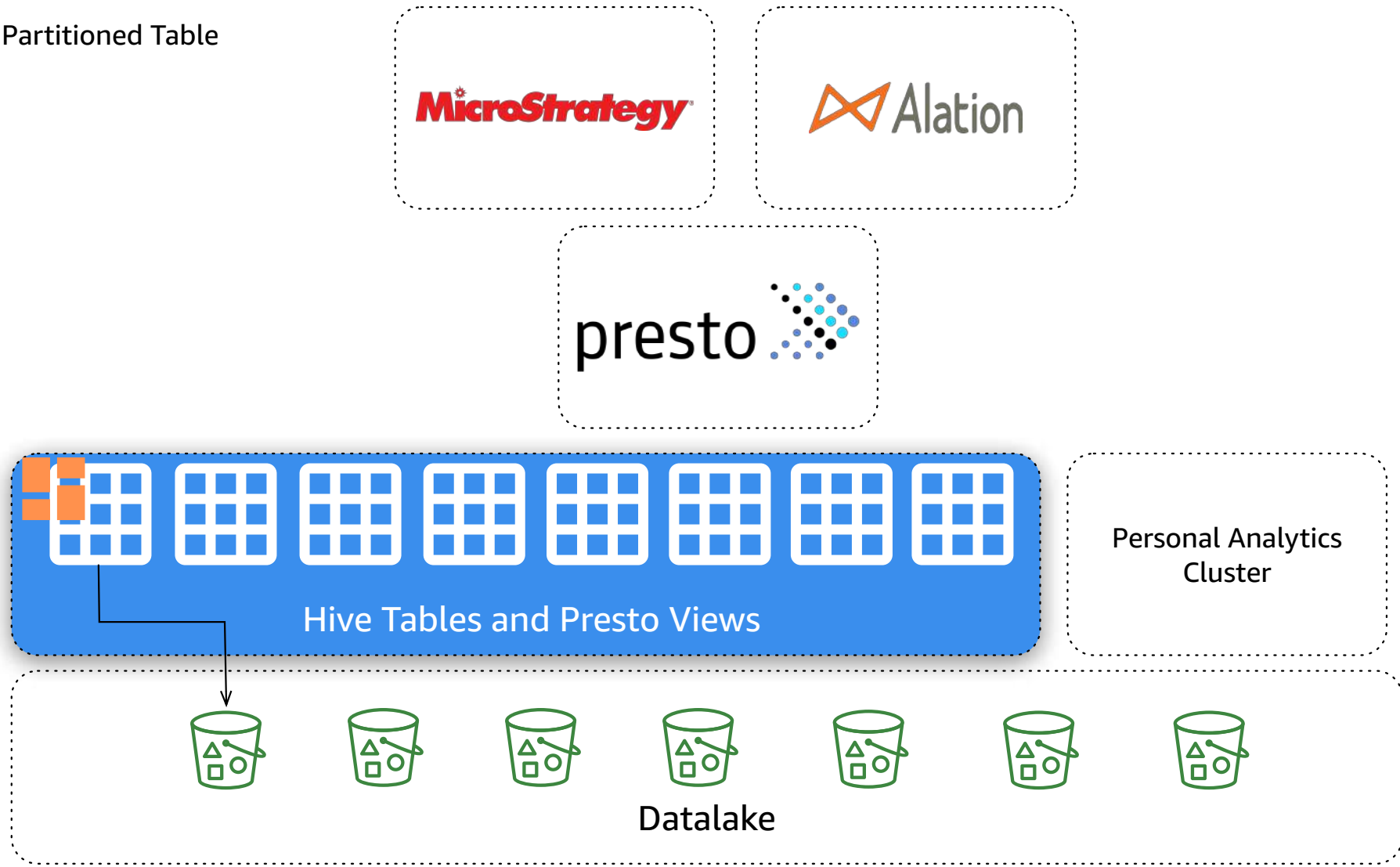
SCOUT 24

Automated Partition Detection – A Motivation

Data Platform

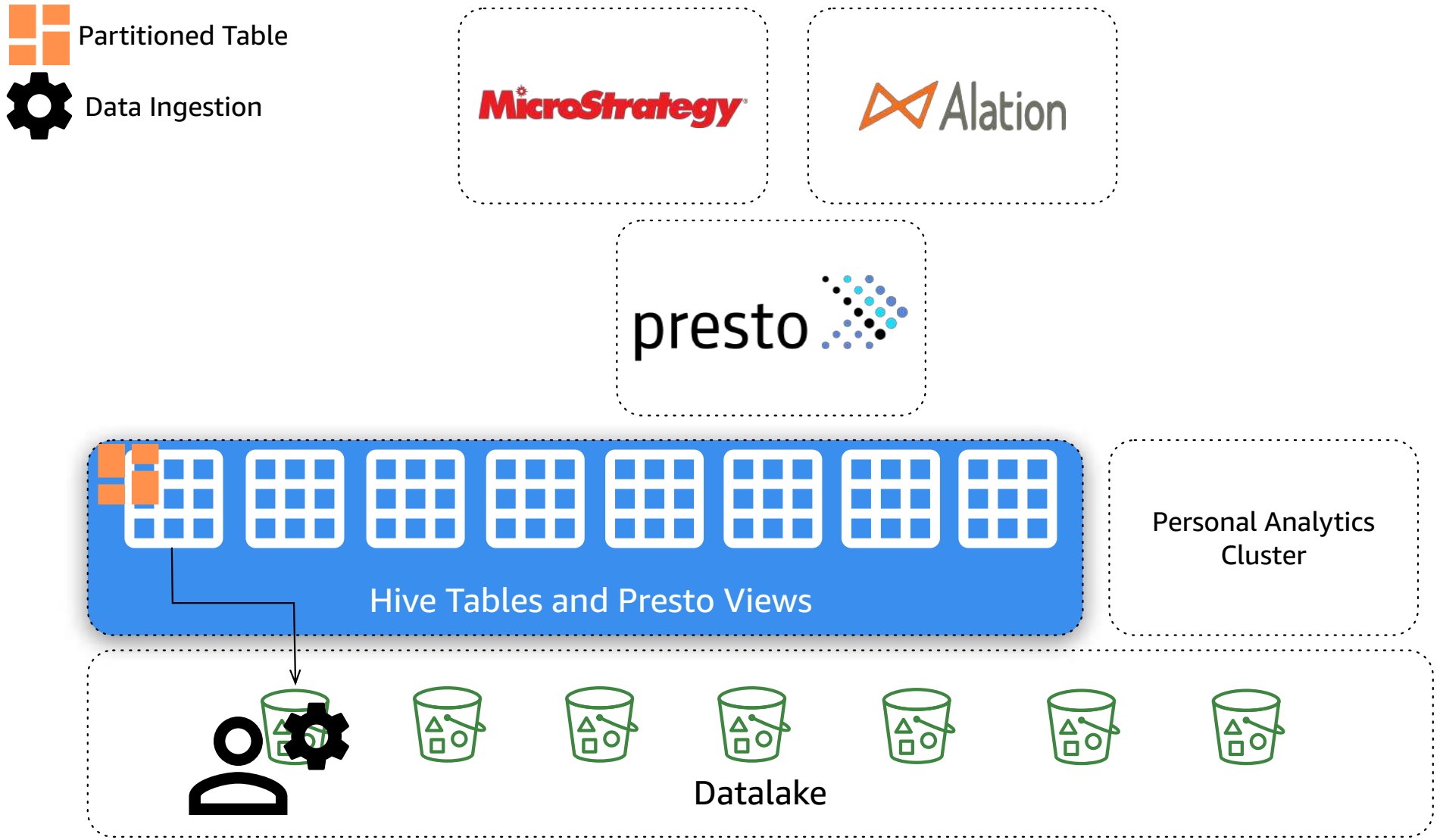
SCOUT 24

Automated Partition Detection – A Motivation



Data Platform SCOUT 24

Automated Partition Detection – A Motivation



Data Platform

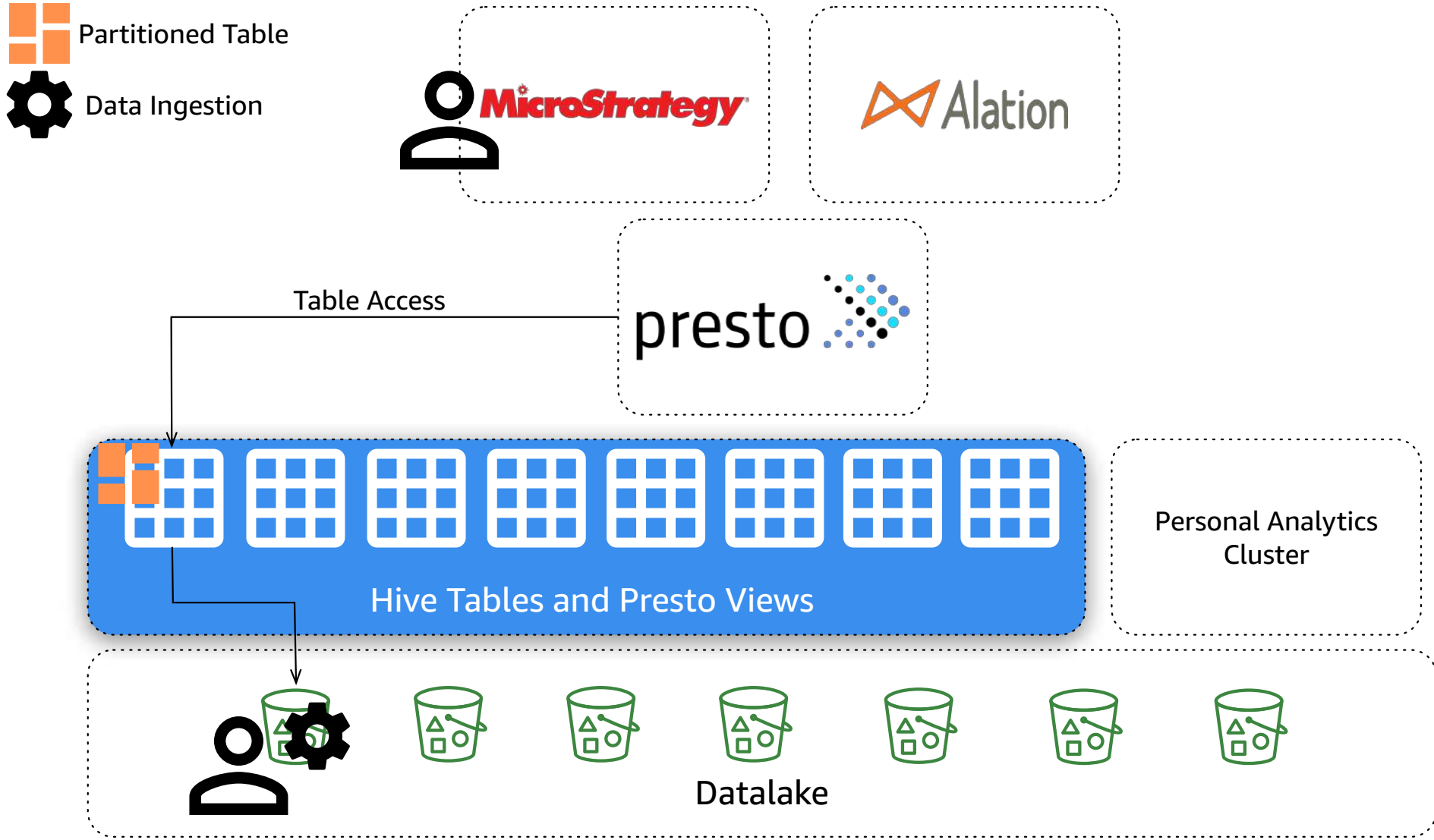
SCOUT 24

Automated Partition Detection – A Motivation

Data Platform

SCOUT 24

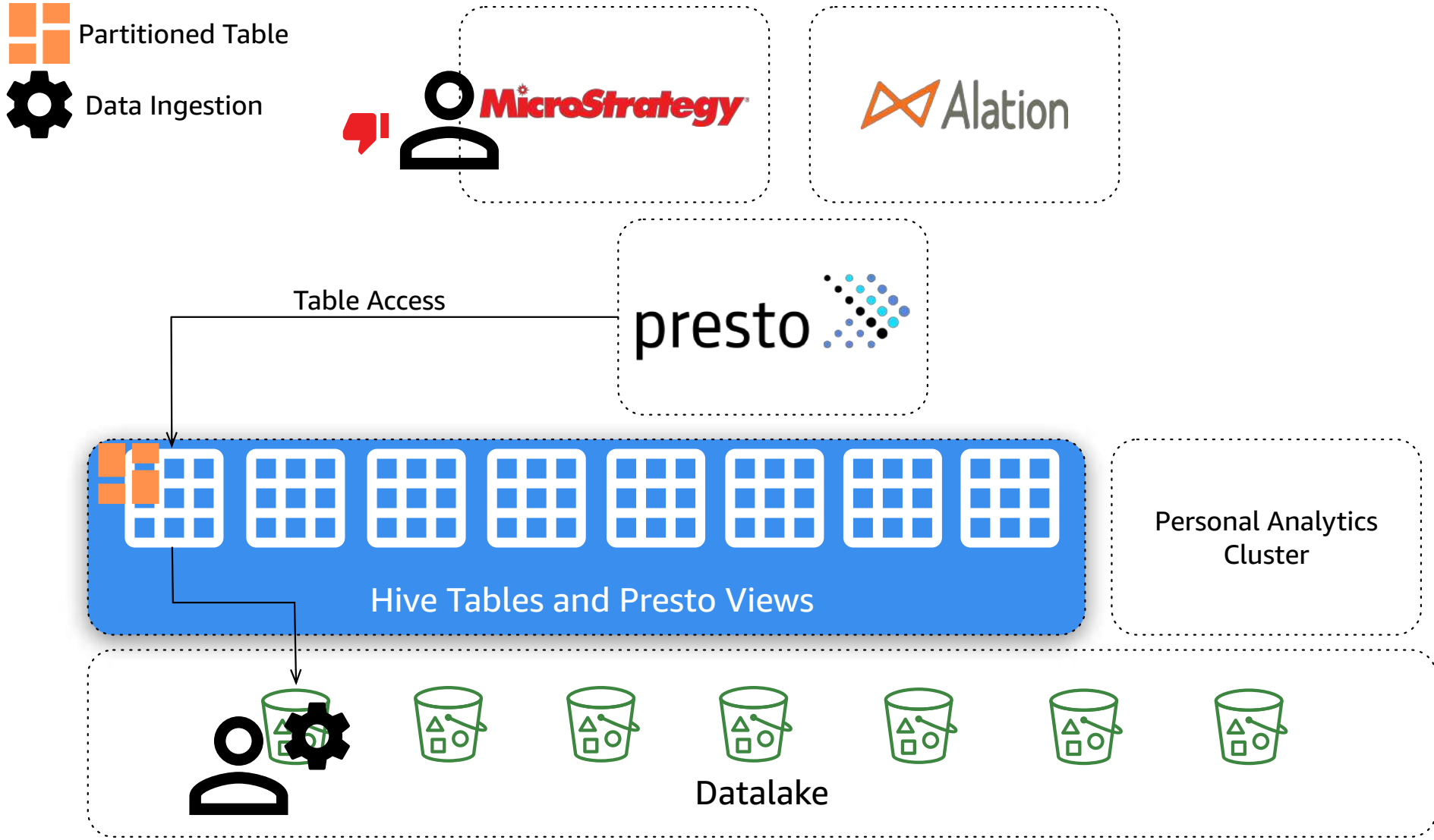
Automated Partition Detection – A Motivation



Data Platform

SCOUT 24

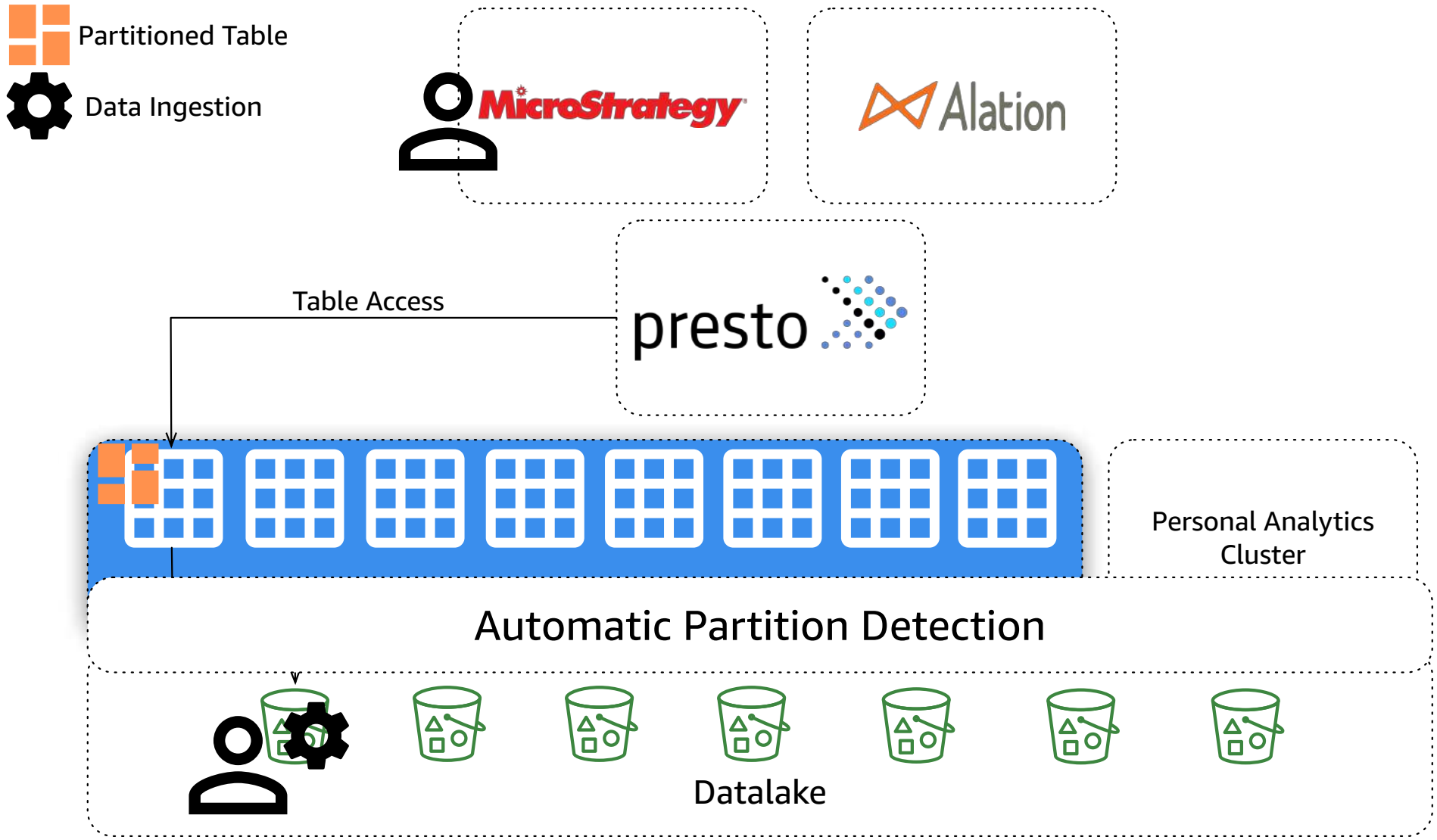
Automated Partition Detection – A Motivation



Data Platform

SCOUT 24

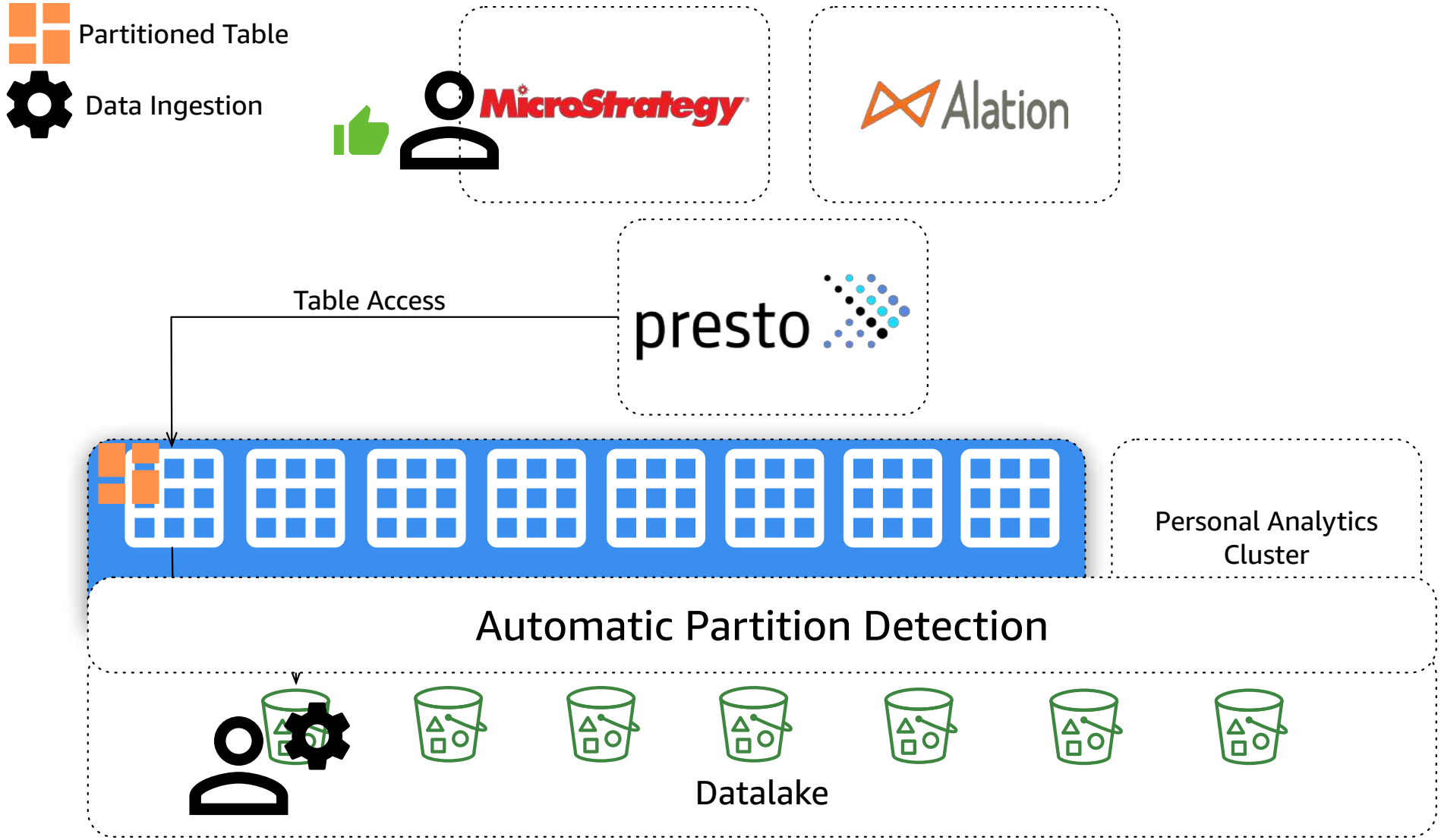
Automated Partition Detection – A Motivation



Data Platform

SCOUT 24

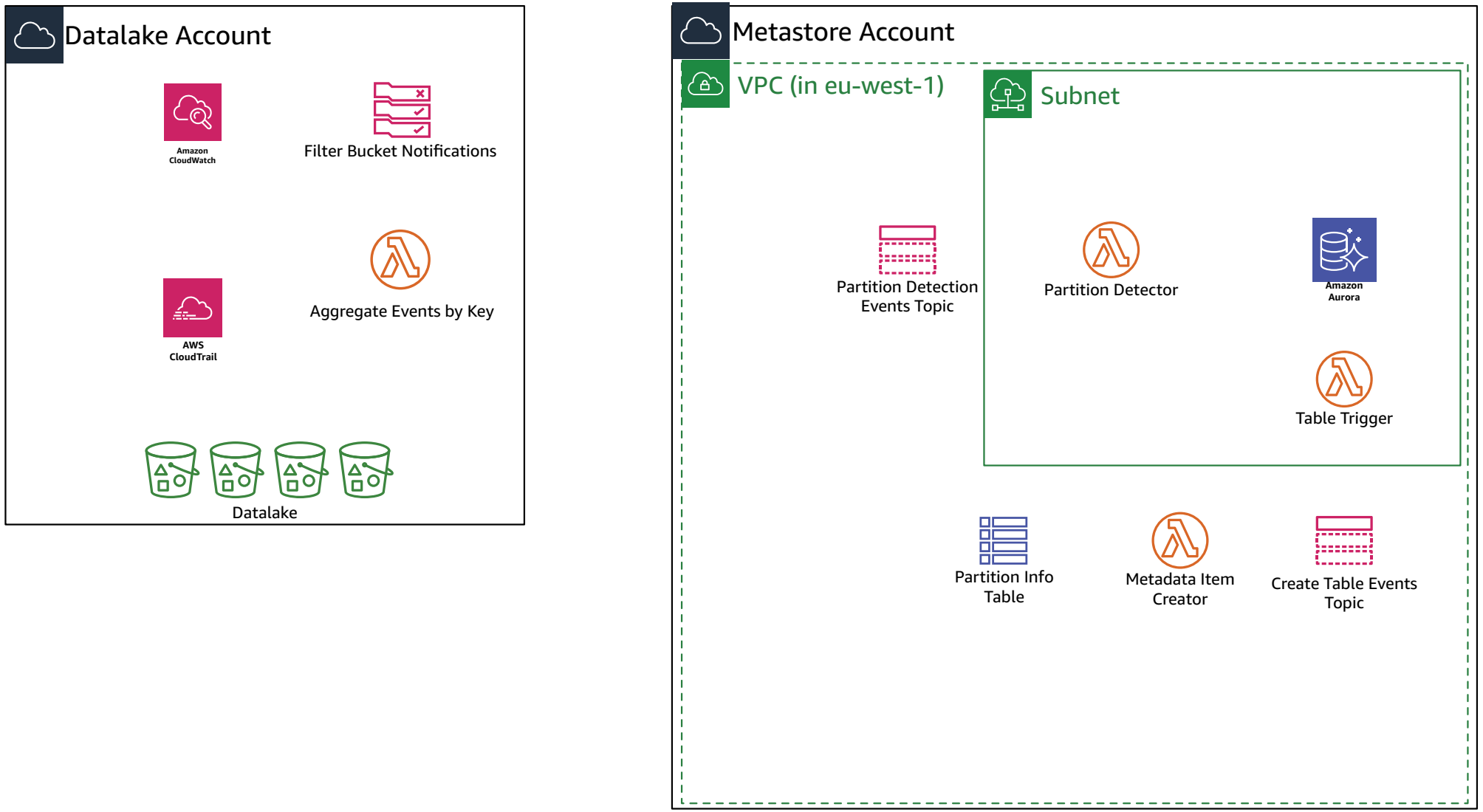
Automated Partition Detection – A Motivation



Data Platform

SCOUT 24

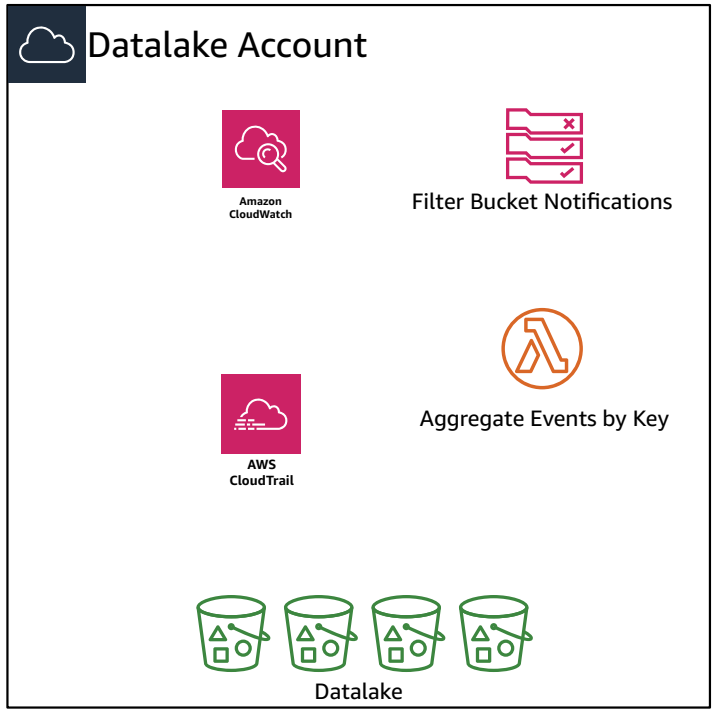
Partition Detection Architecture

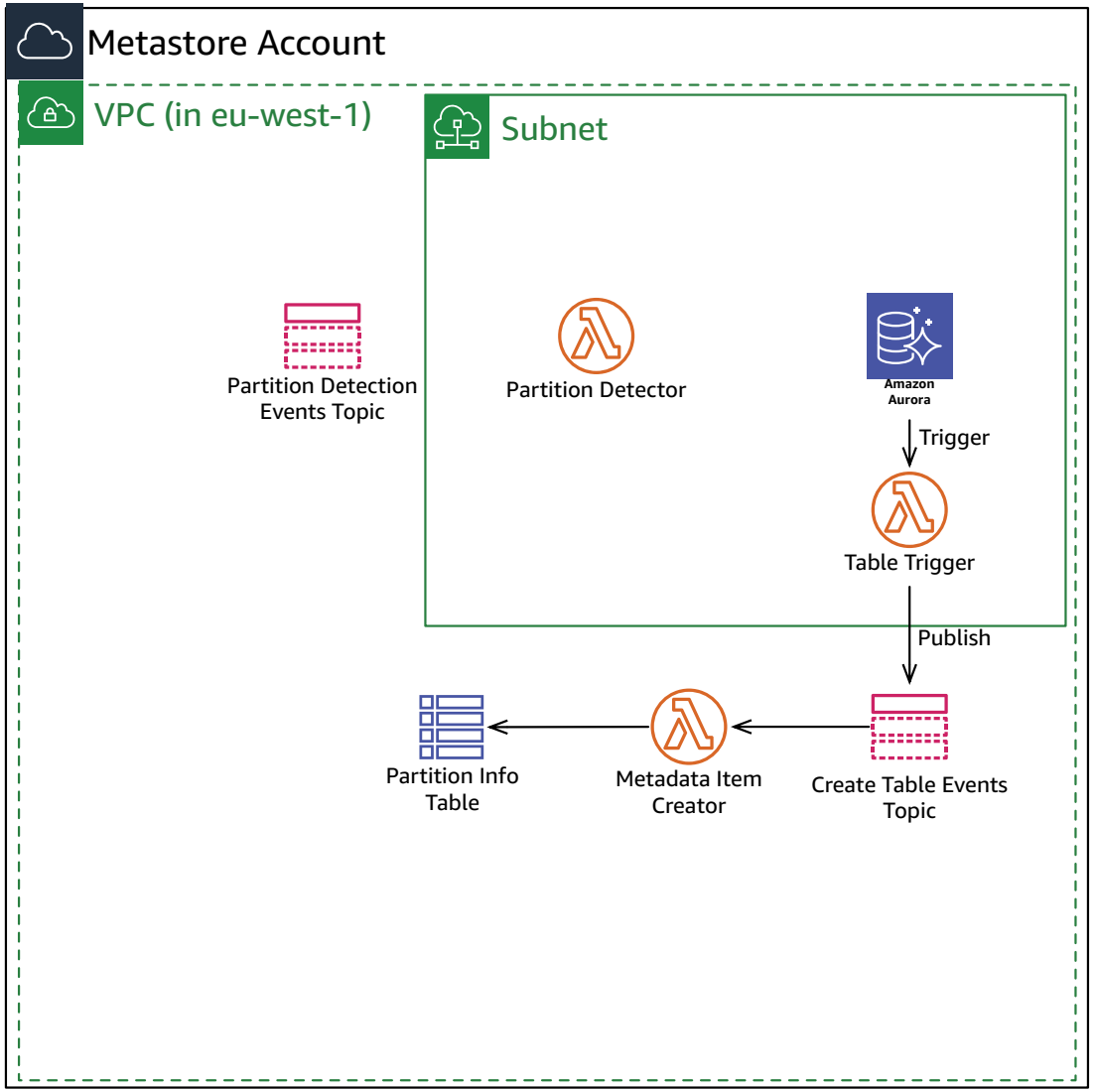


Data Platform

SCOUT 24

Partition Detection Architecture

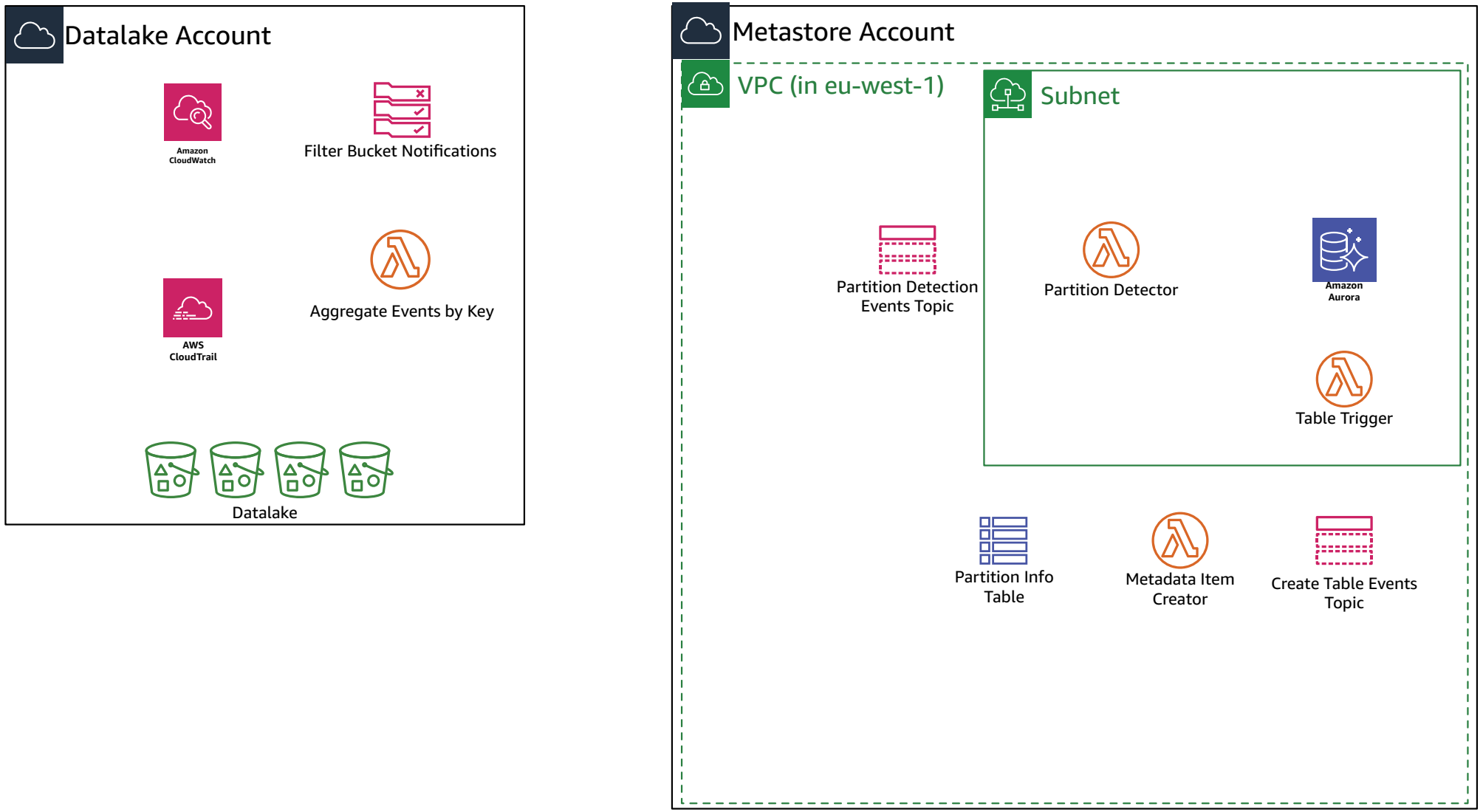




Data Platform

SCOUT 24

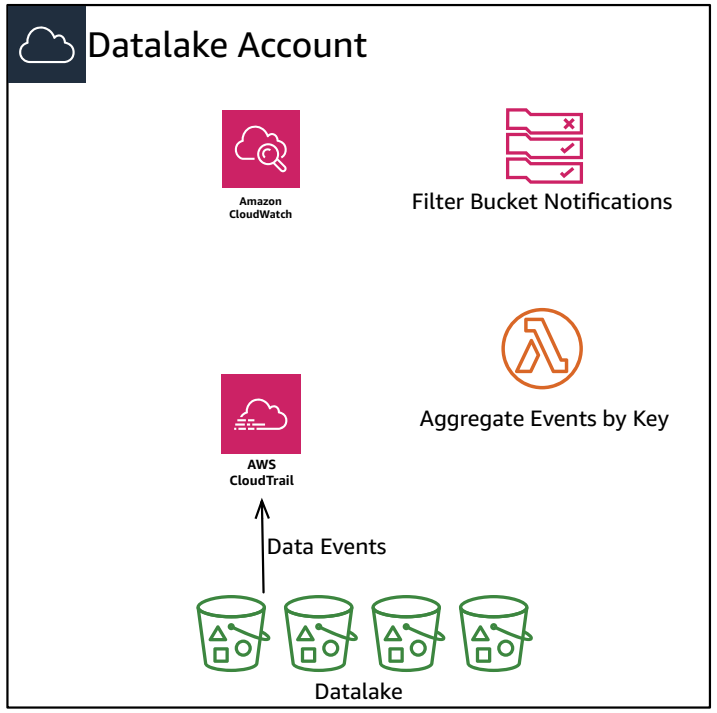
Partition Detection Architecture

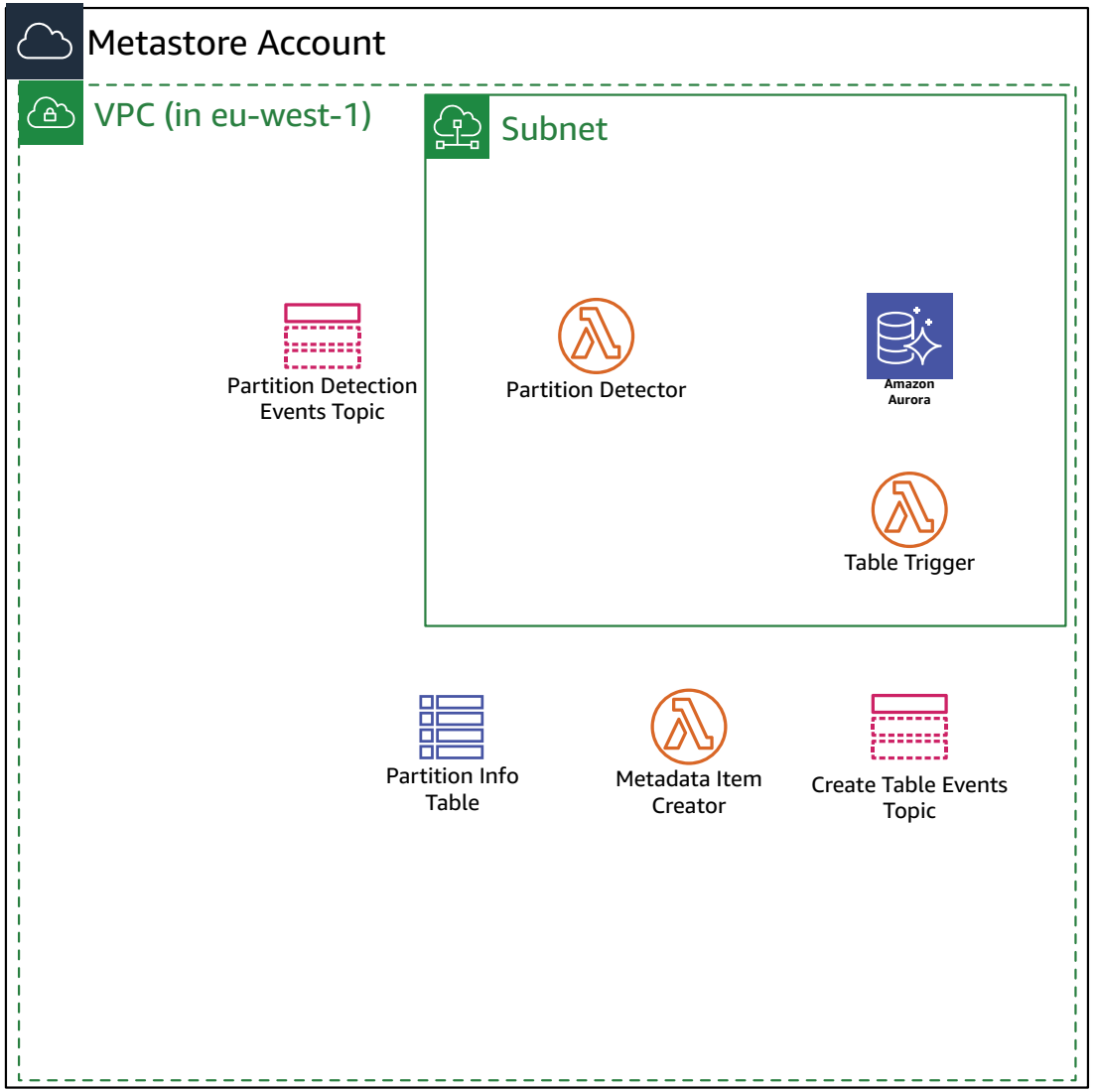


Data Platform

SCOUT 24

Partition Detection Architecture

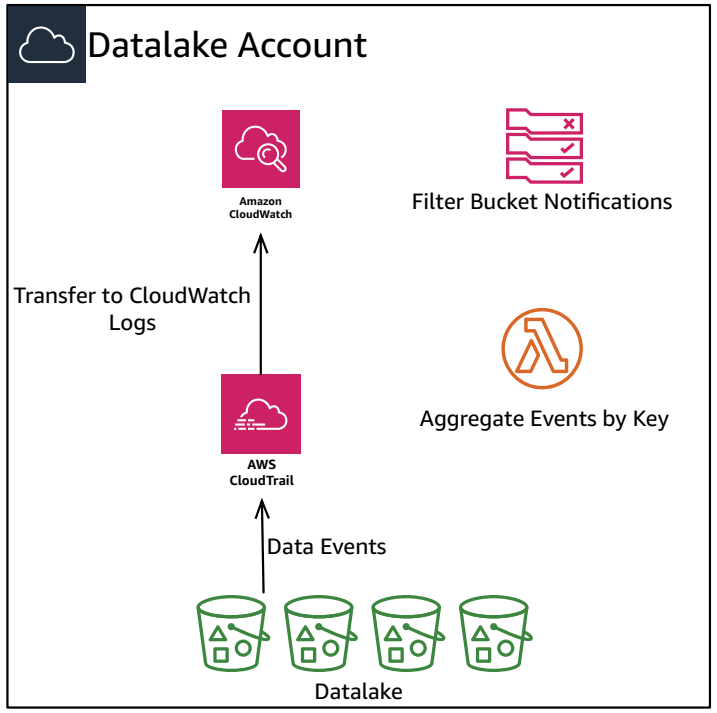


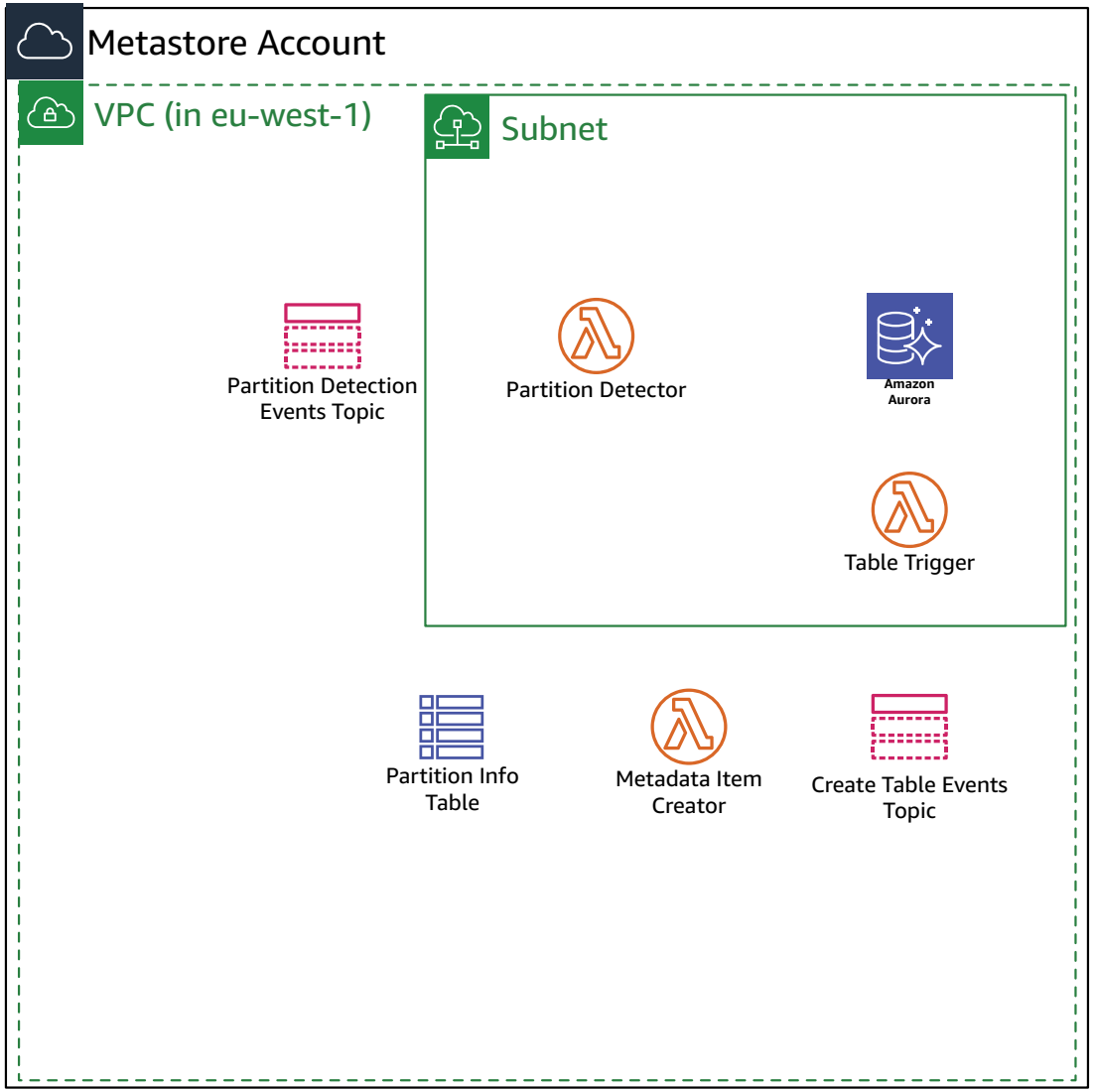


Data Platform

SCOUT 24

Partition Detection Architecture

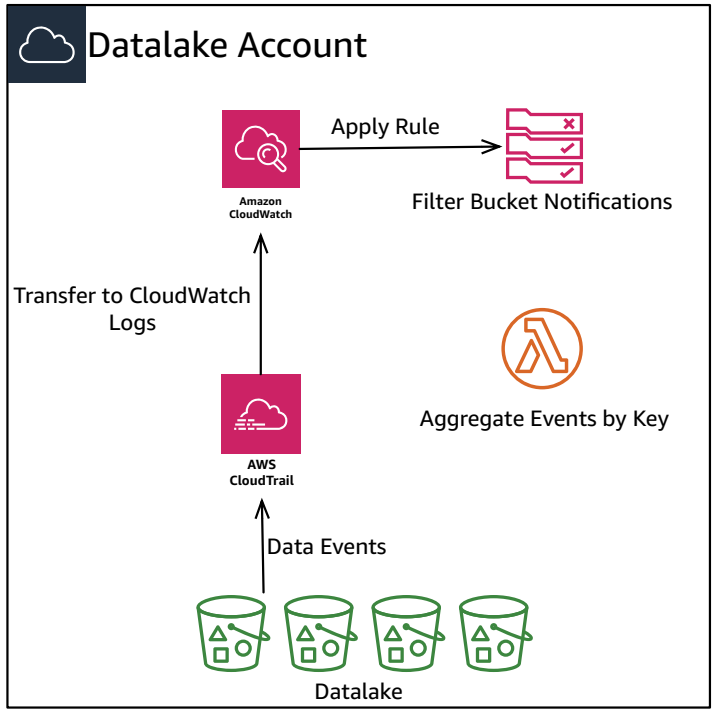


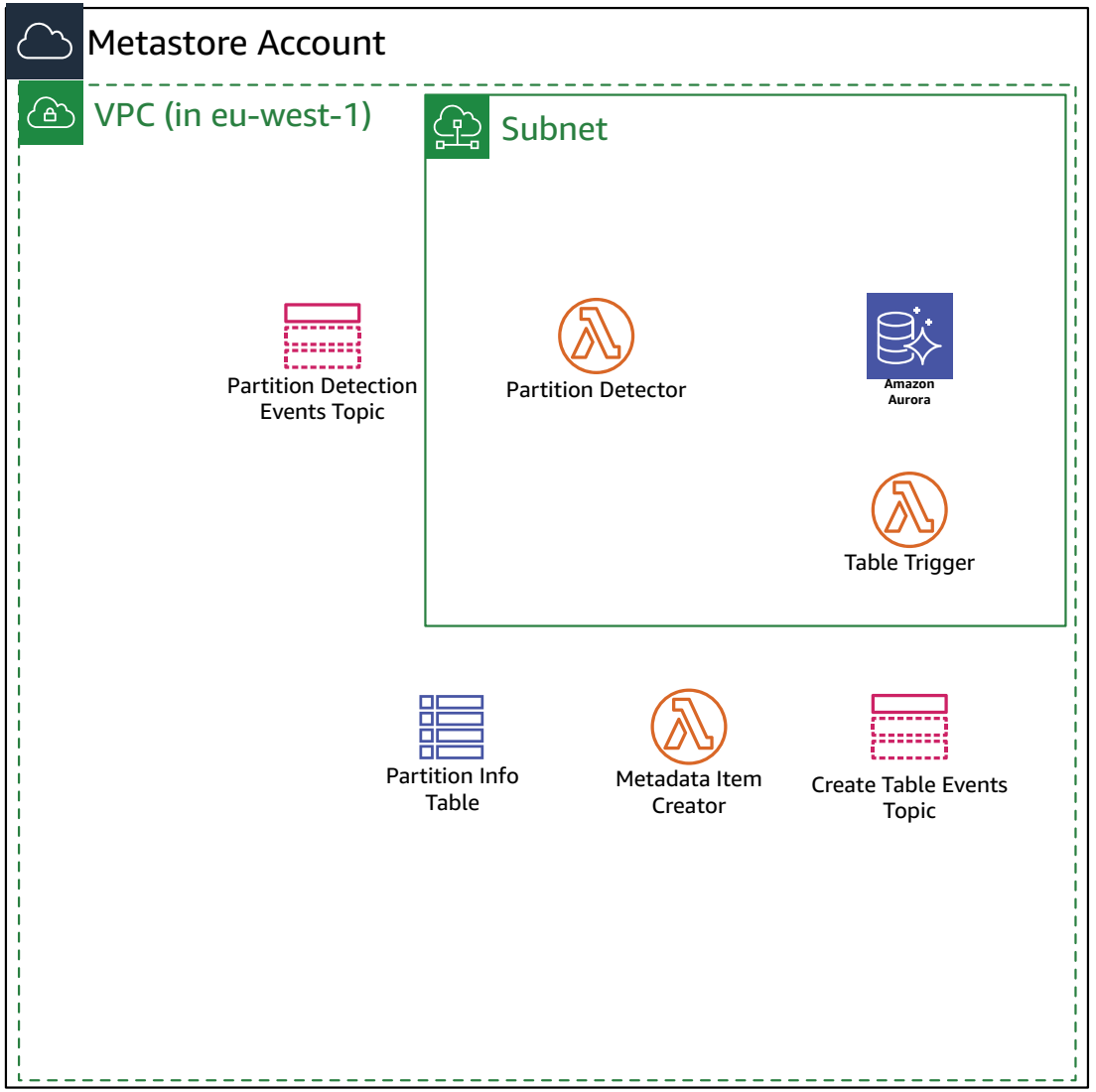


Data Platform

SCOUT 24

Partition Detection Architecture

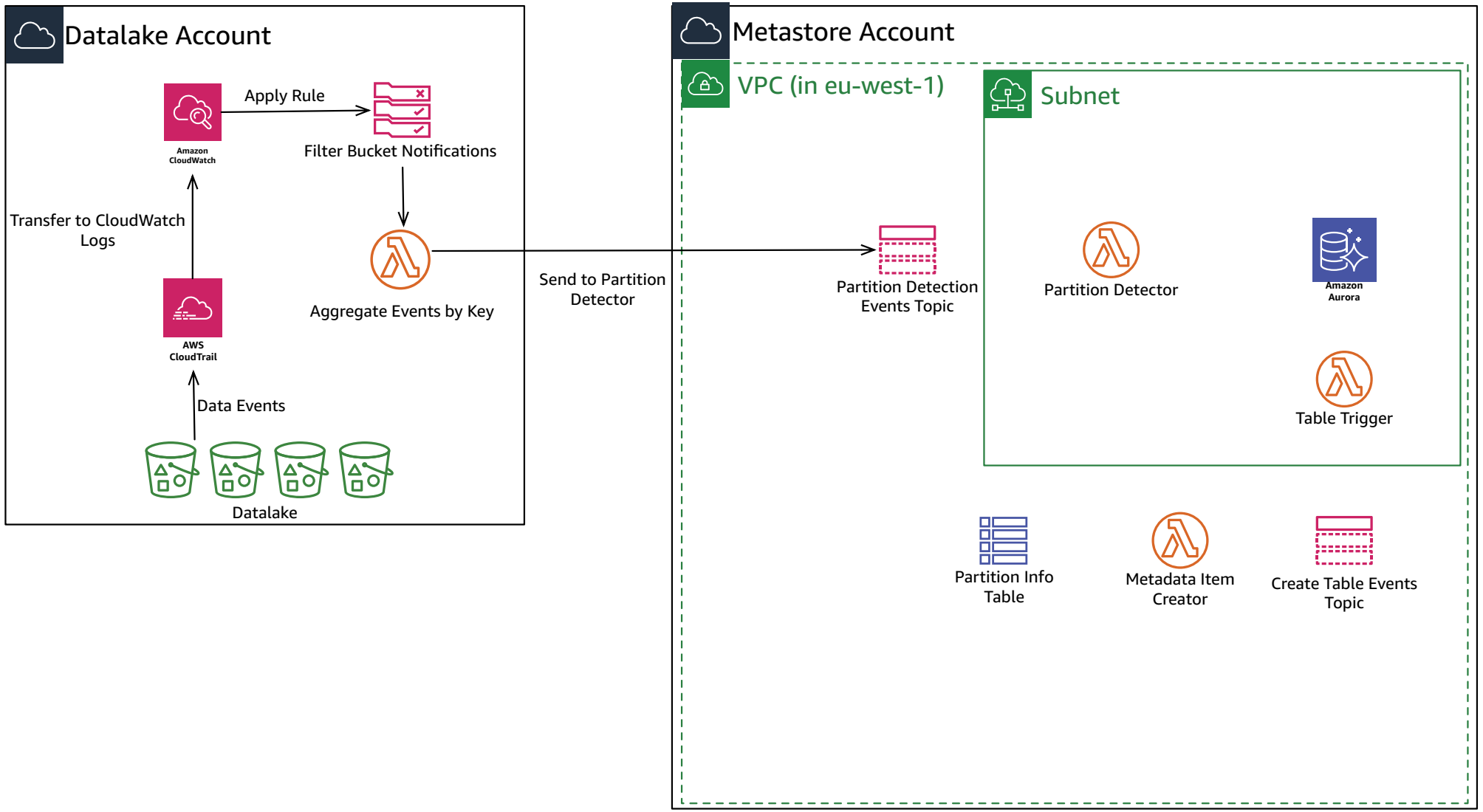




Data Platform

SCOUT 24

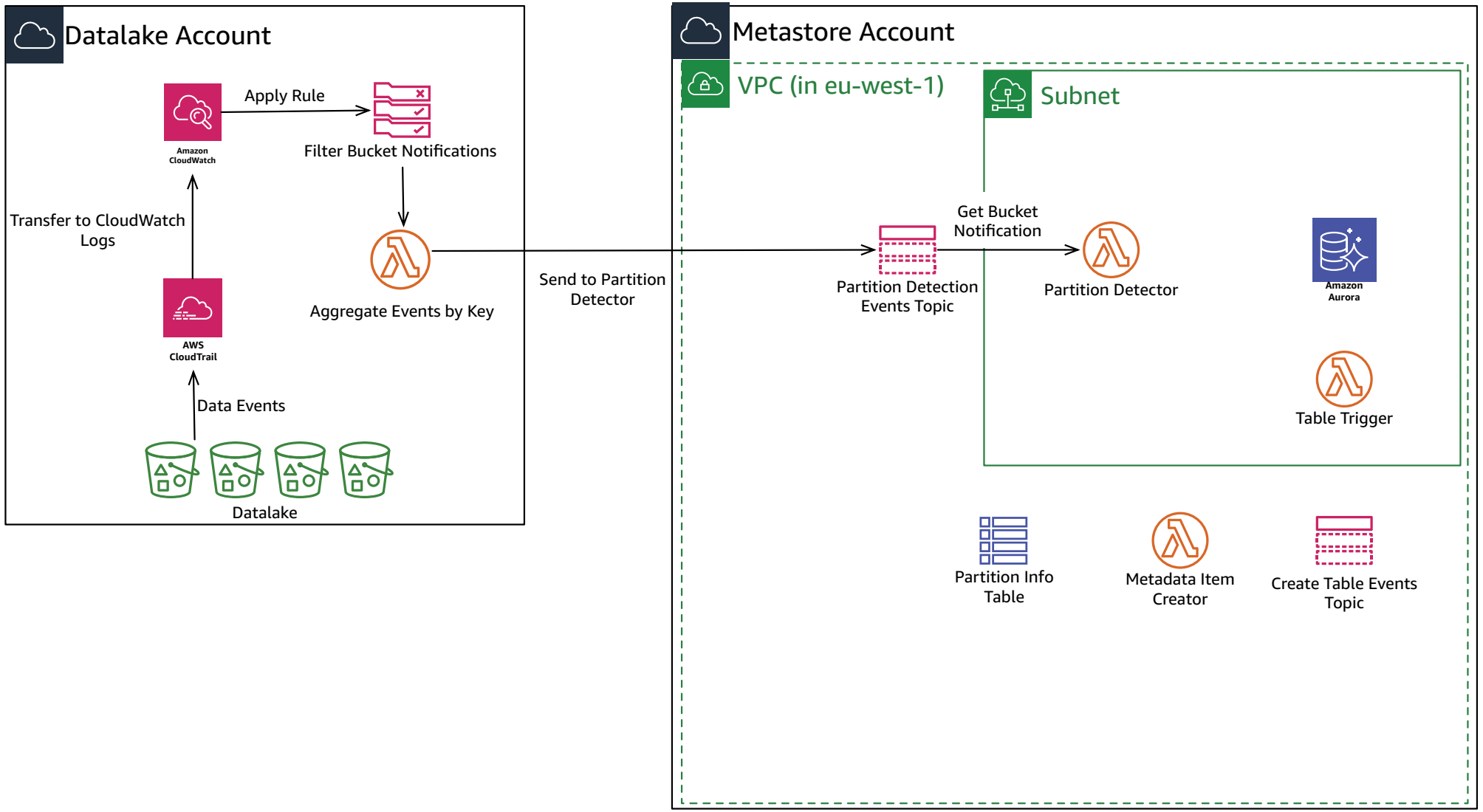
Partition Detection Architecture



Data Platform

SCOUT 24

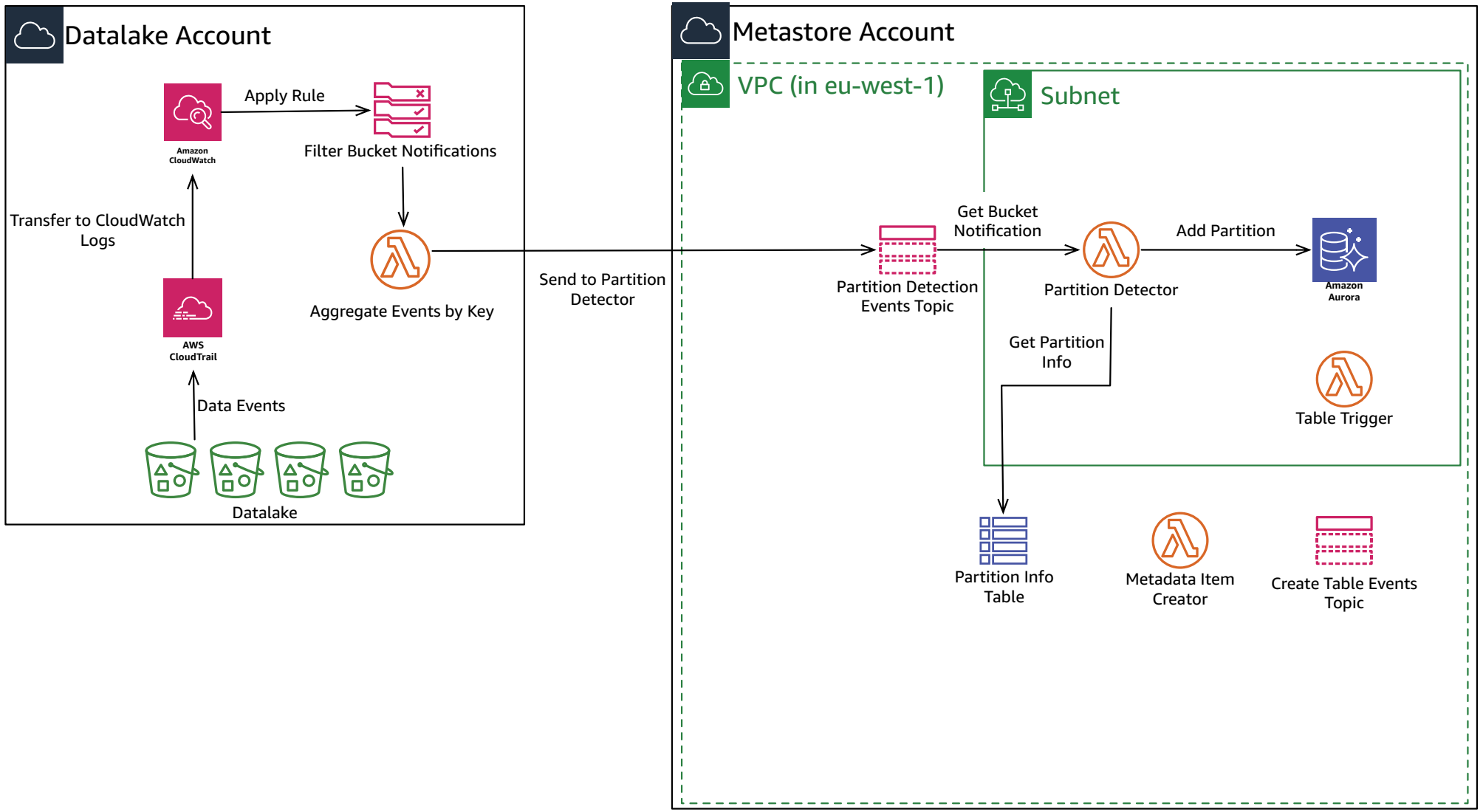
Partition Detection Architecture



Data Platform

SCOUT 24

Partition Detection Architecture



Data Platform

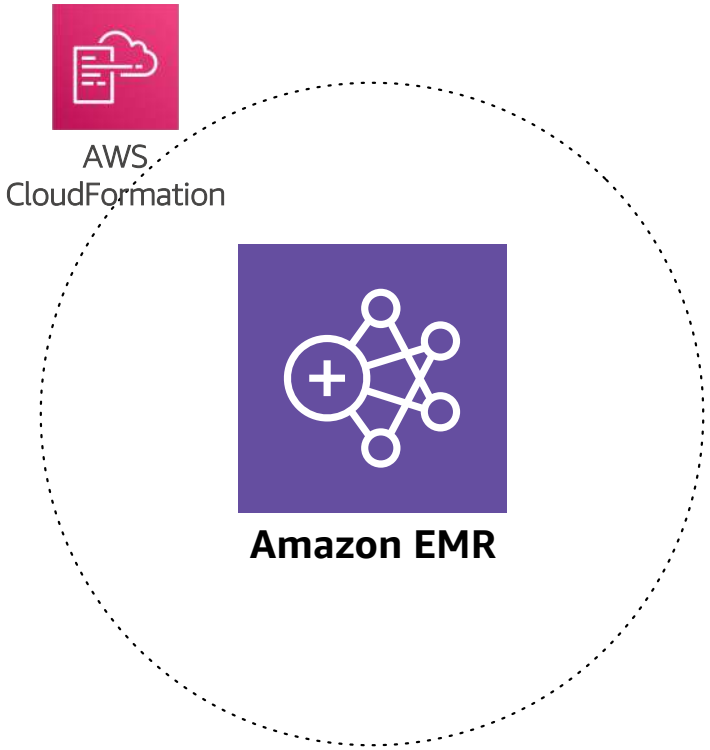
SCOUT 24

Personal Analytics Cluster

Data Platform

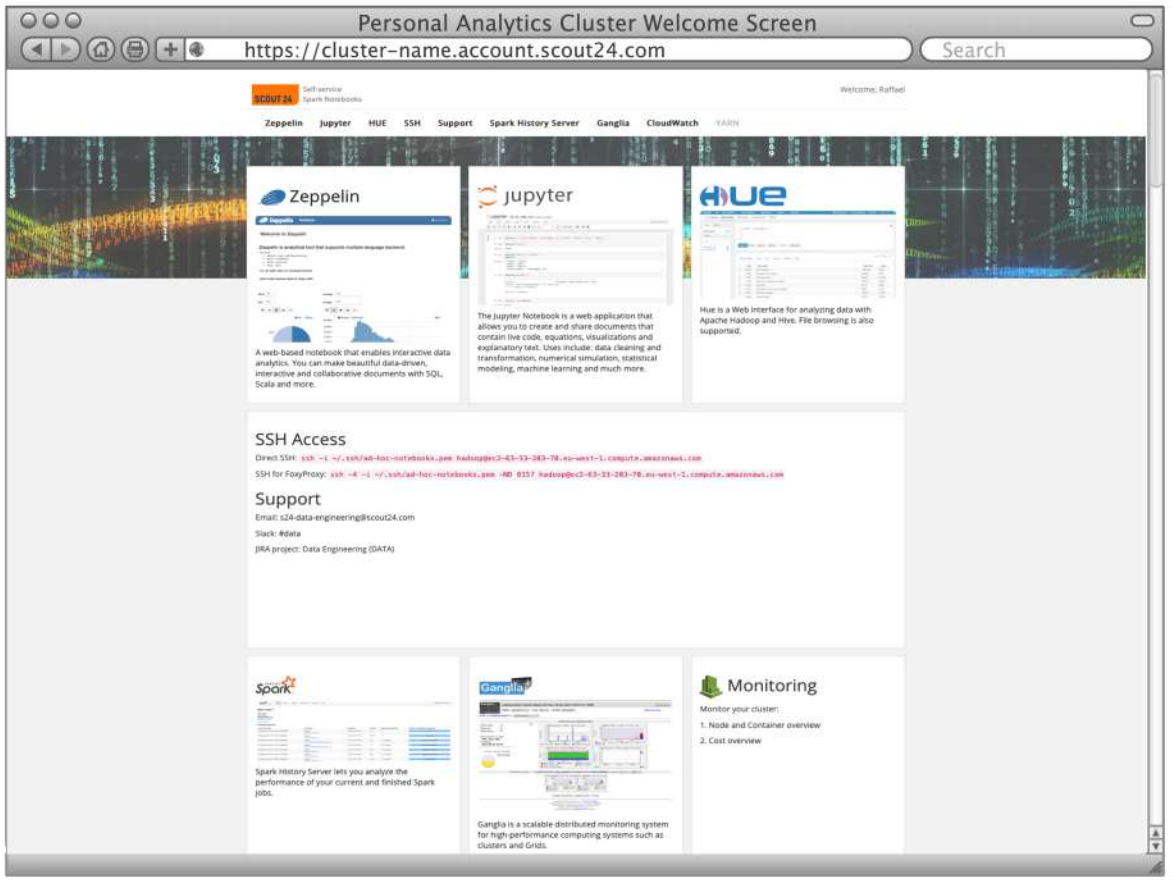
SCOUT 24

The Personal Analytics Cluster – An Overview



Data Platform SCOUT 24

The Personal Analytics Cluster – An Overview



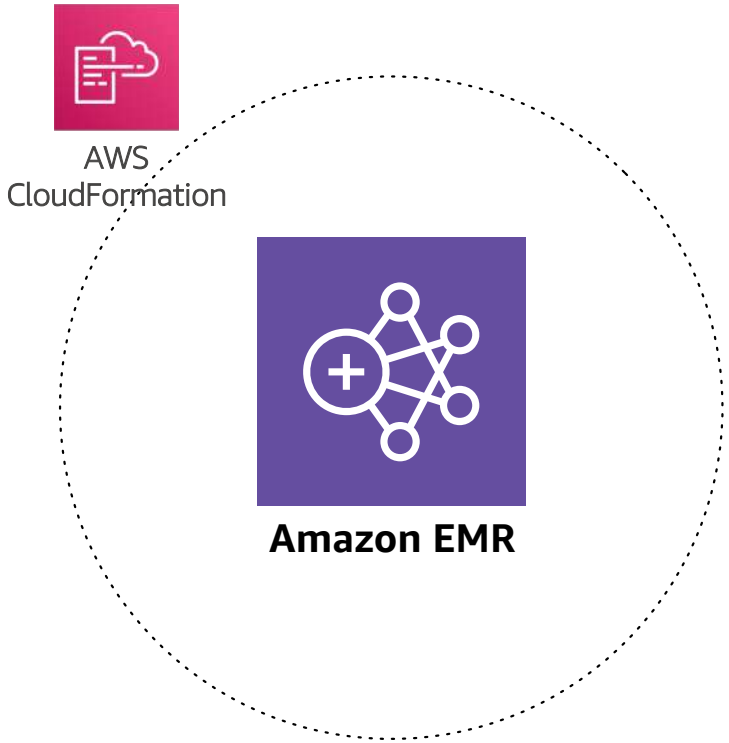
Personal Analytics Cluster

- Easy Access via Web Interface
- Zeppelin and Jupyter Notebook Restore
- OneClick Deployment
- Managed Scaling and Shutdown
- Support for Pre-baked AMIs and Configs

Data Platform

SCOUT 24

The Personal Analytics Cluster – An Overview



Provisioned Infrastructure

Daily: 20

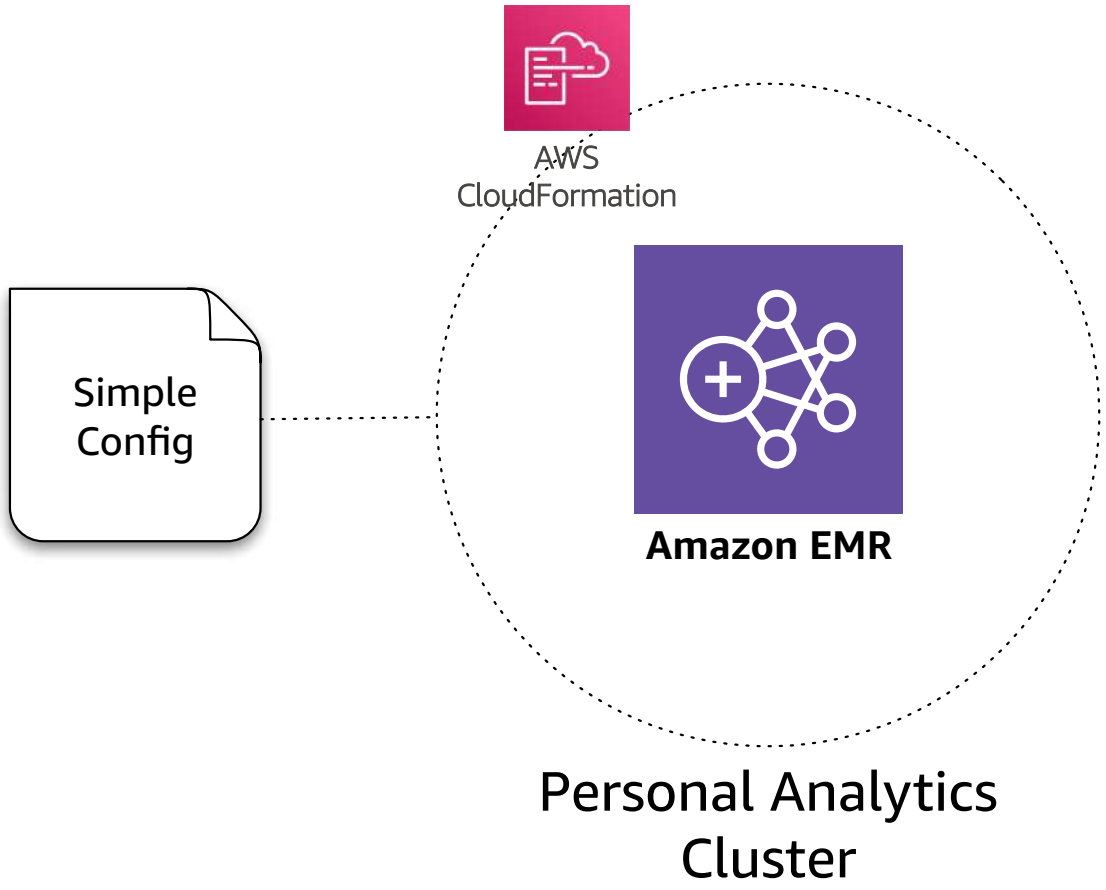
Weekly: 128

Monthly: 389

Data Platform

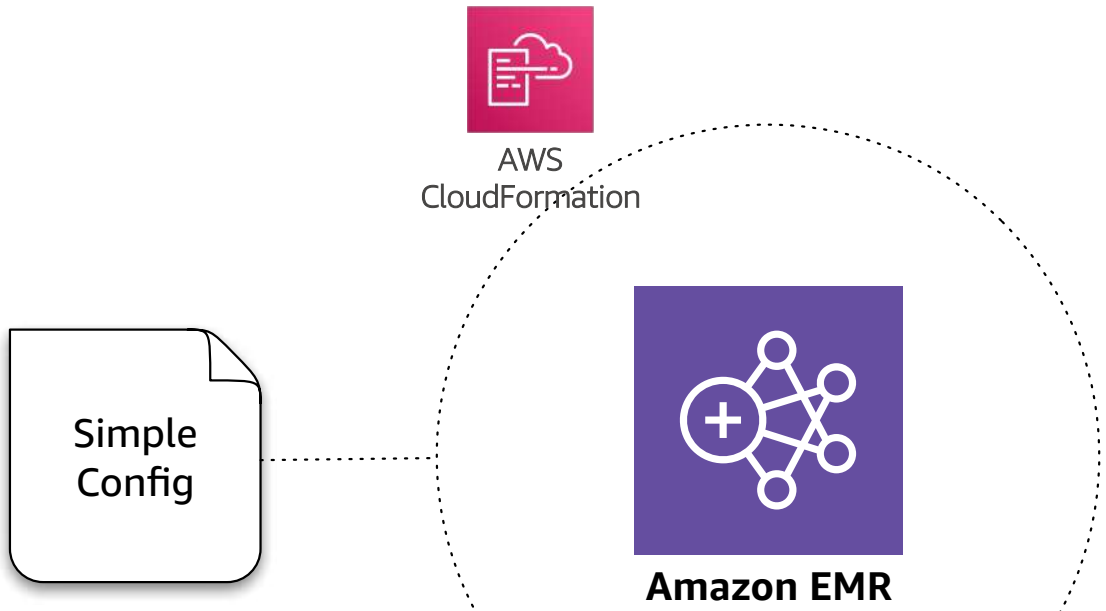
SCOUT 24

The Personal Analytics Cluster – An Overview



Data Platform SCOUT 24

The Personal Analytics Cluster – An Overview



- Easy Access via Web Interface
- Zeppelin and Jupyter Notebook Restore
- OneClick Deployment
- Managed Scaling and Shutdown
- Support for Pre-baked AMIs and Configs

Data Platform

SCOUT 24

The Personal Analytics Cluster – Configuration Setup

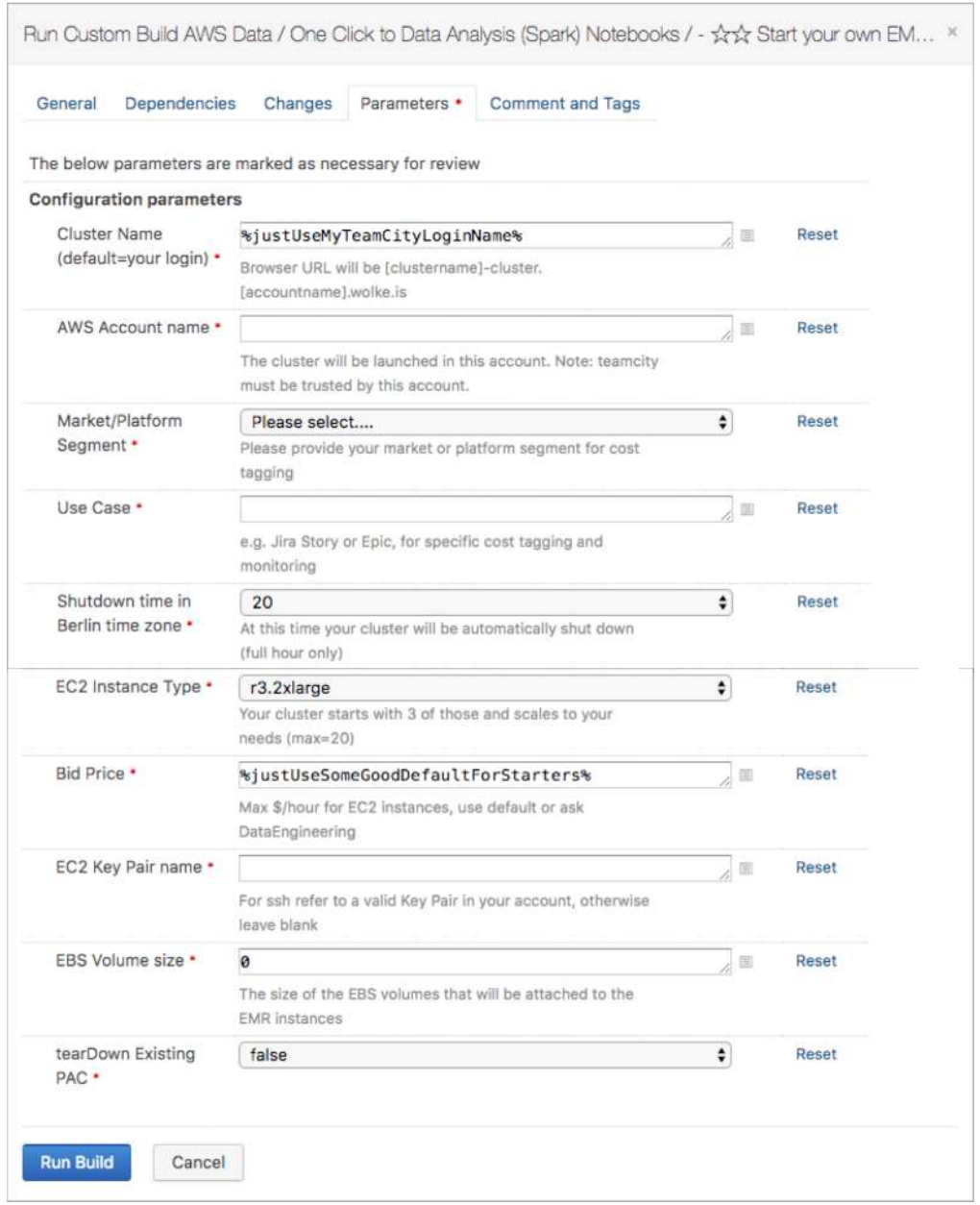
- Cluster Name (default is usually username)
- AWS Account Name
- Market segment
- Use Case
- Shutdown time
- Instance Type
- Bid Price
- Key Pair
- EBS Volume Size
- Tear Down Existing Cluster (optional)

The Personal Analytics Cluster – Configuration Setup

```
$ ./deploy-cluster --cluster-name <> \
        --account-name <> \
        --market-segment <> \
        --usecase <> \
        --shutdown-time <> \
        --instance-type <> \
        --bid-price <> \
        --key-pair <>  \
        --ebs-volume-size <> \
        --teardown-existing <>
```

Data Platform

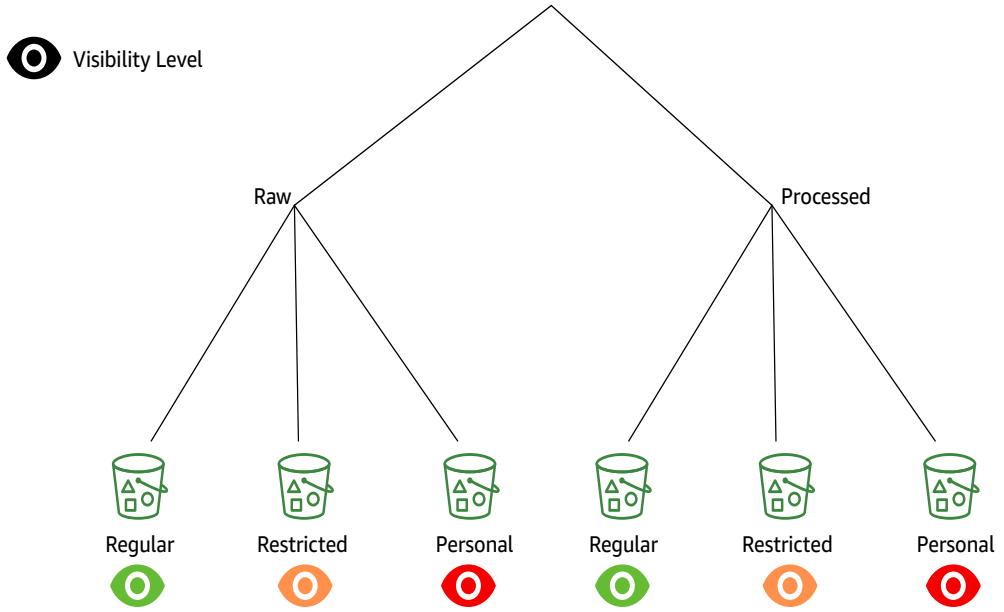
SCOUT 24

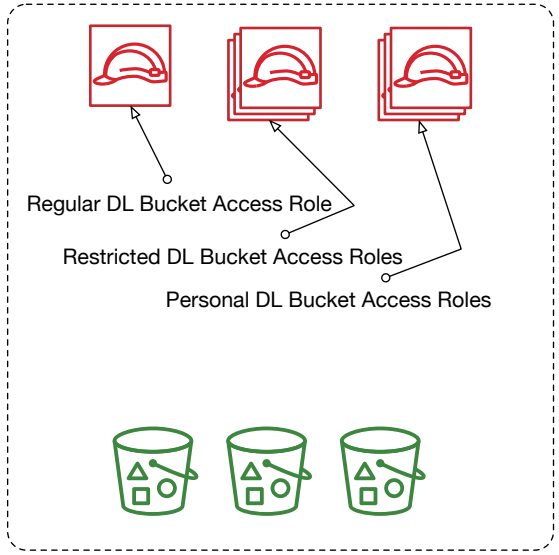


Data Platform

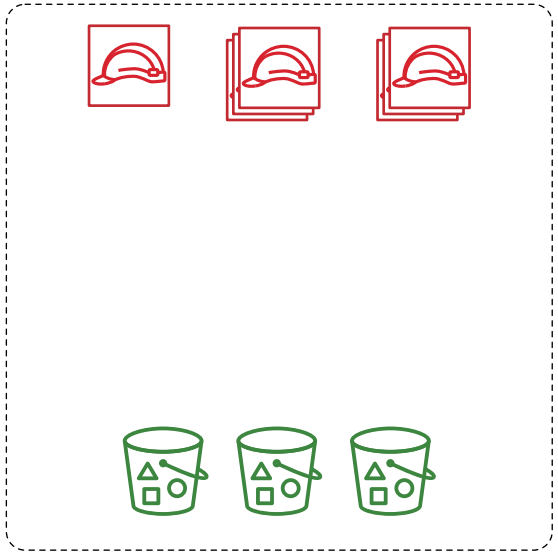
SCOUT 24

The Personal Analytics Cluster – Data Lake Access





Data Lake Account
ImmobilienScout24



Data Lake Account
AutoScout24

Data Platform

SCOUT 24

The Personal Analytics Cluster – Data Lake Access

- Managed with AWS EMR FS Security Configuration
- Integrates with EMR Applications internally; Spark, Presto, etc
- Easy to define
- Maps a defined IAM role to S3 bucket being accessed

Data Platform

SCOUT 24

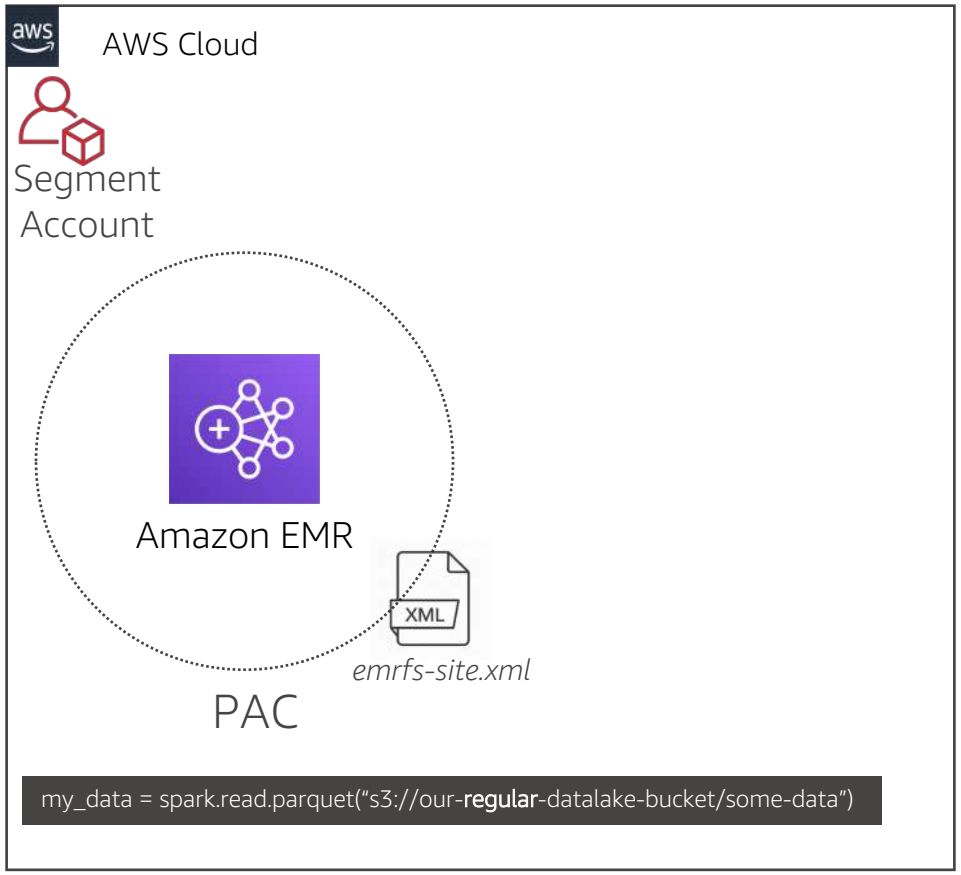
The Personal Analytics Cluster – Data Lake Access

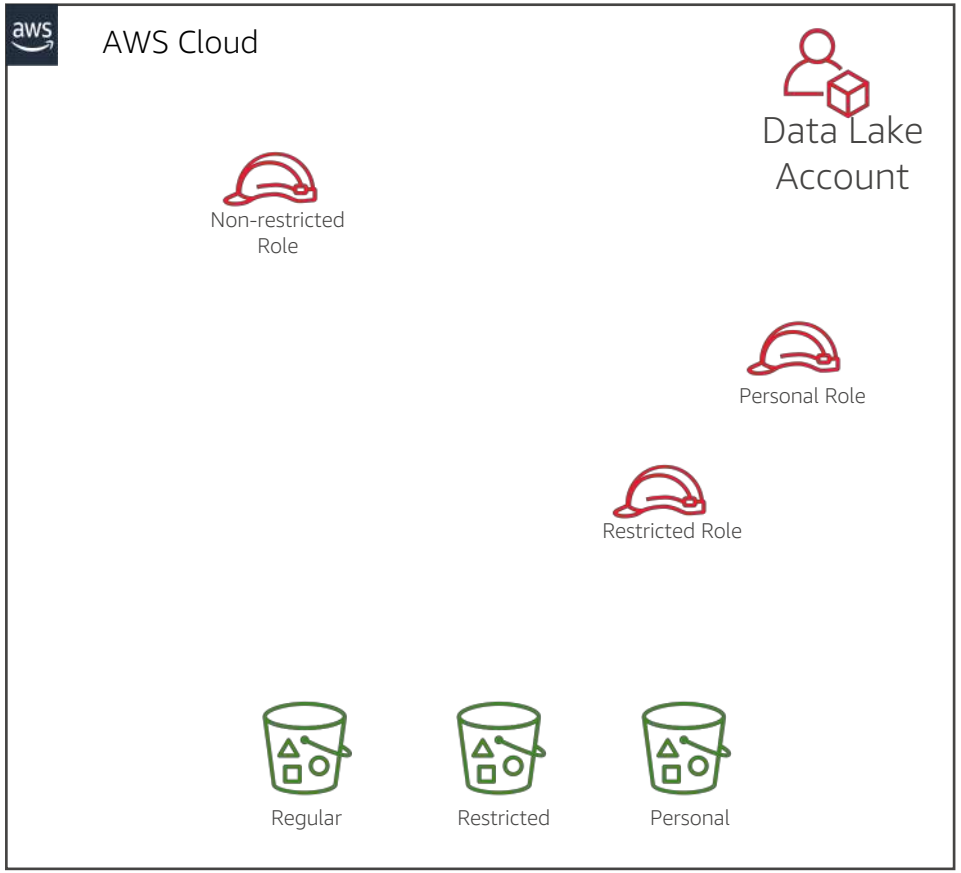
Scenario One:
Non-restricted Data Lake Access

Data Platform

SCOUT 24

The Personal Analytics Cluster – Data Lake Access

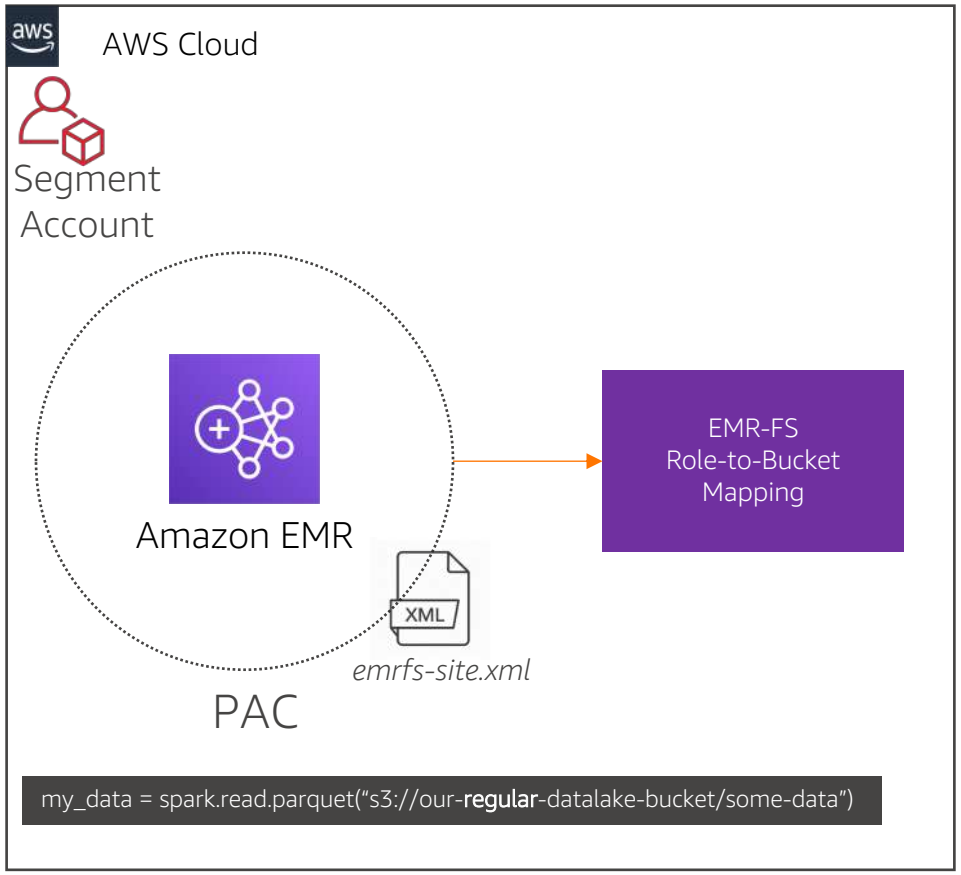


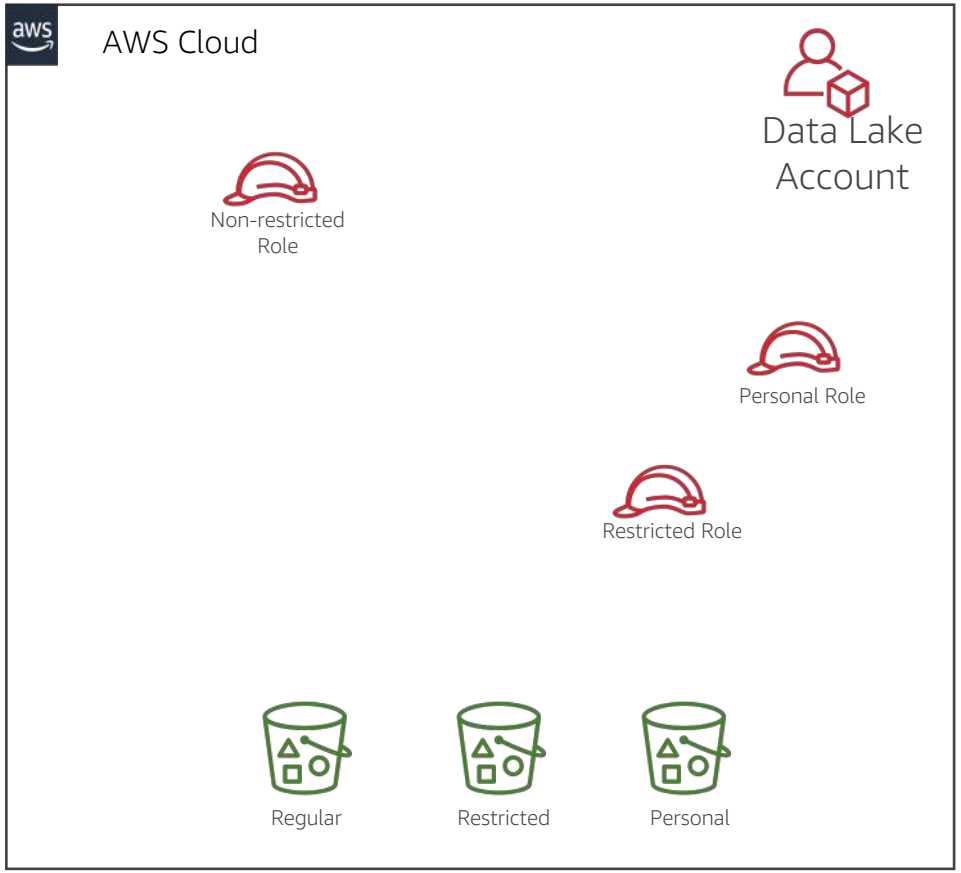


Data Platform

SCOUT 24

The Personal Analytics Cluster – Data Lake Access

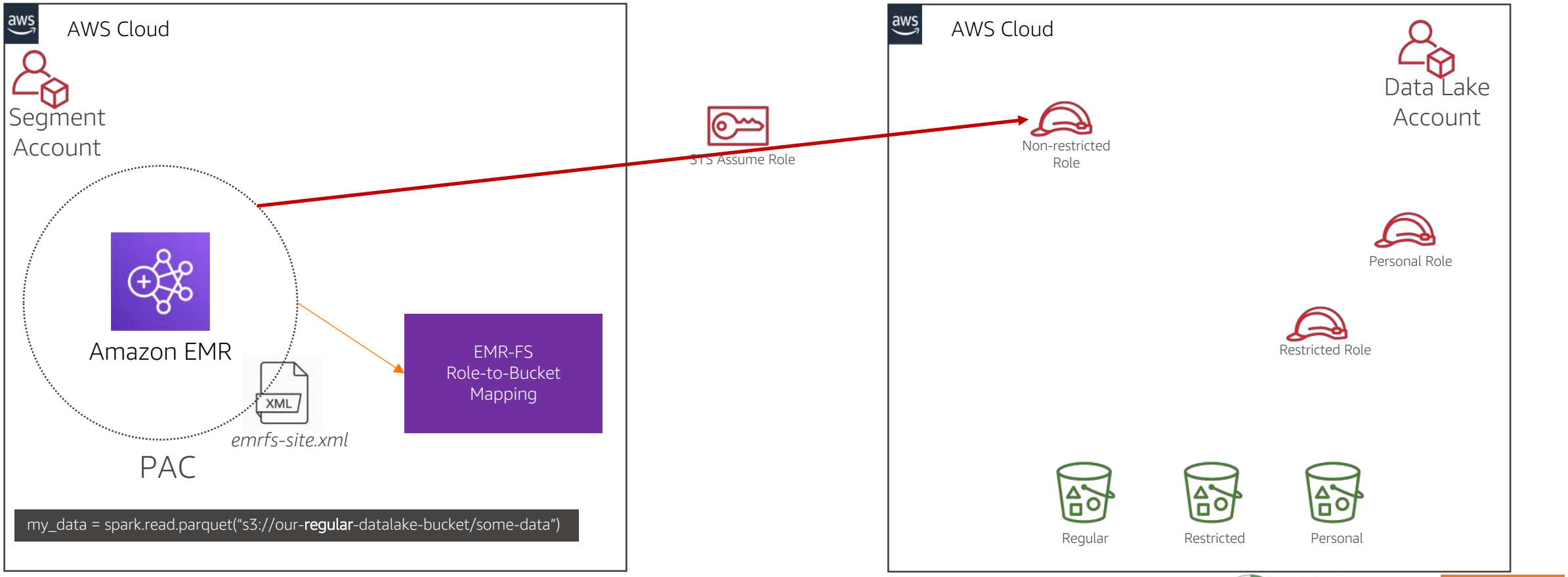




Data Platform

SCOUT 24

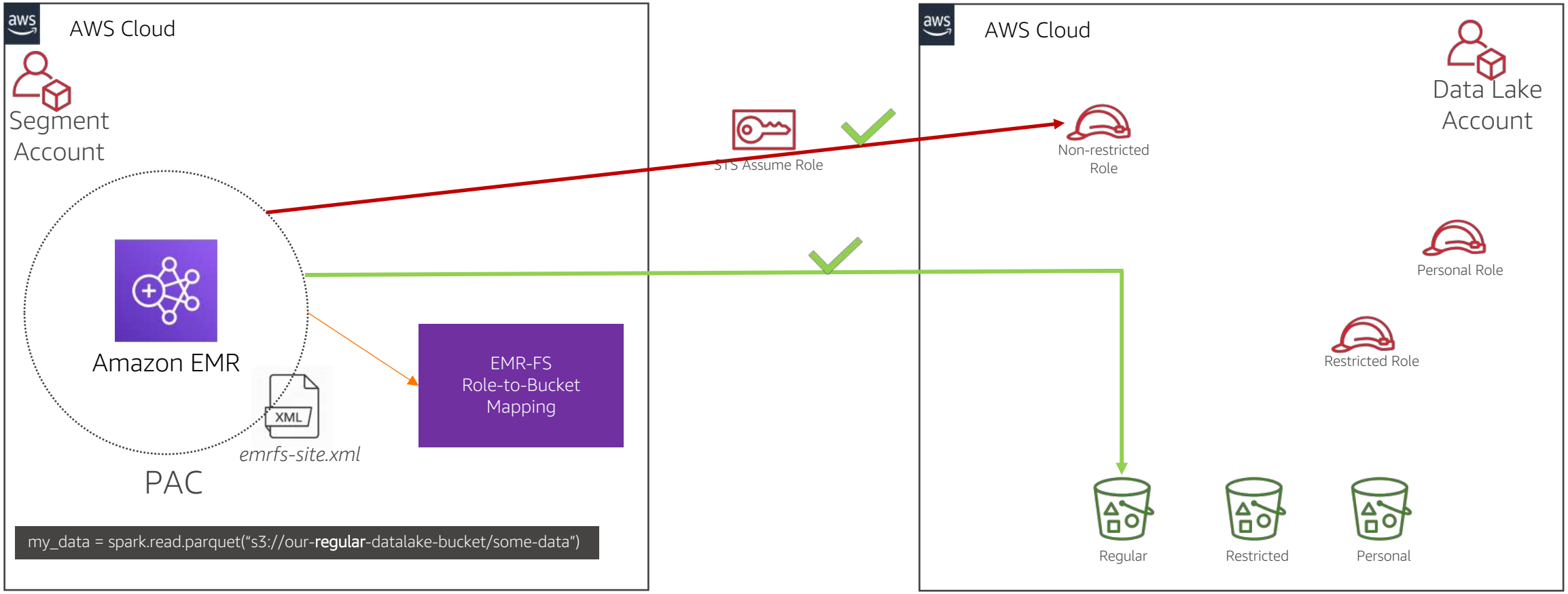
The Personal Analytics Cluster – Data Lake Access



Data Platform

SCOUT 24

The Personal Analytics Cluster – Data Lake Access



Data Platform

SCOUT 24

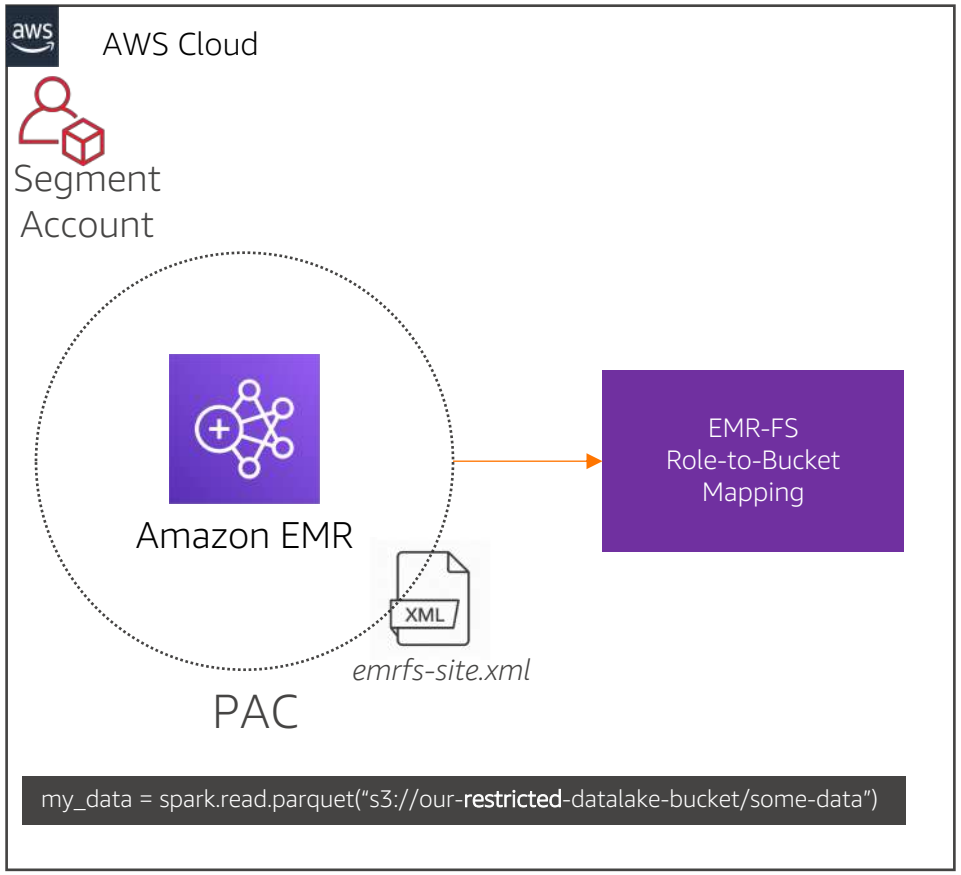
The Personal Analytics Cluster – Data Lake Access

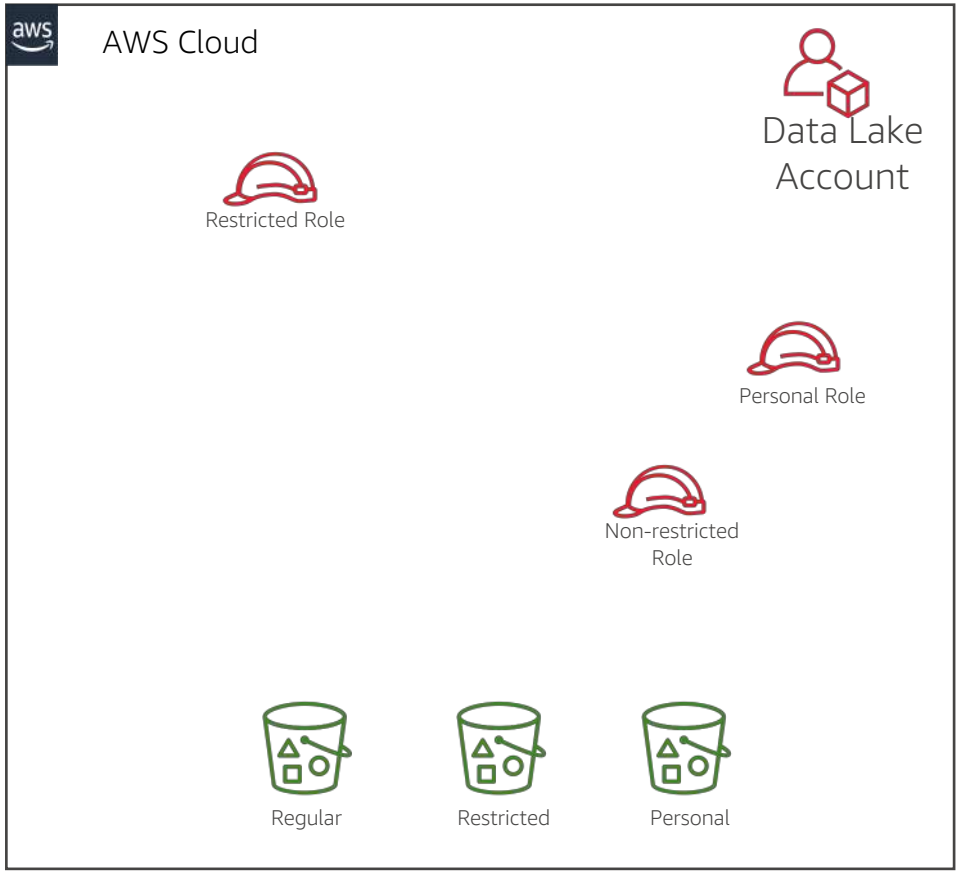
Scenario Two:
Restricted/Personal Data Lake Access

Data
Platform

SCOUT 24

The Personal Analytics Cluster – Data Lake Access

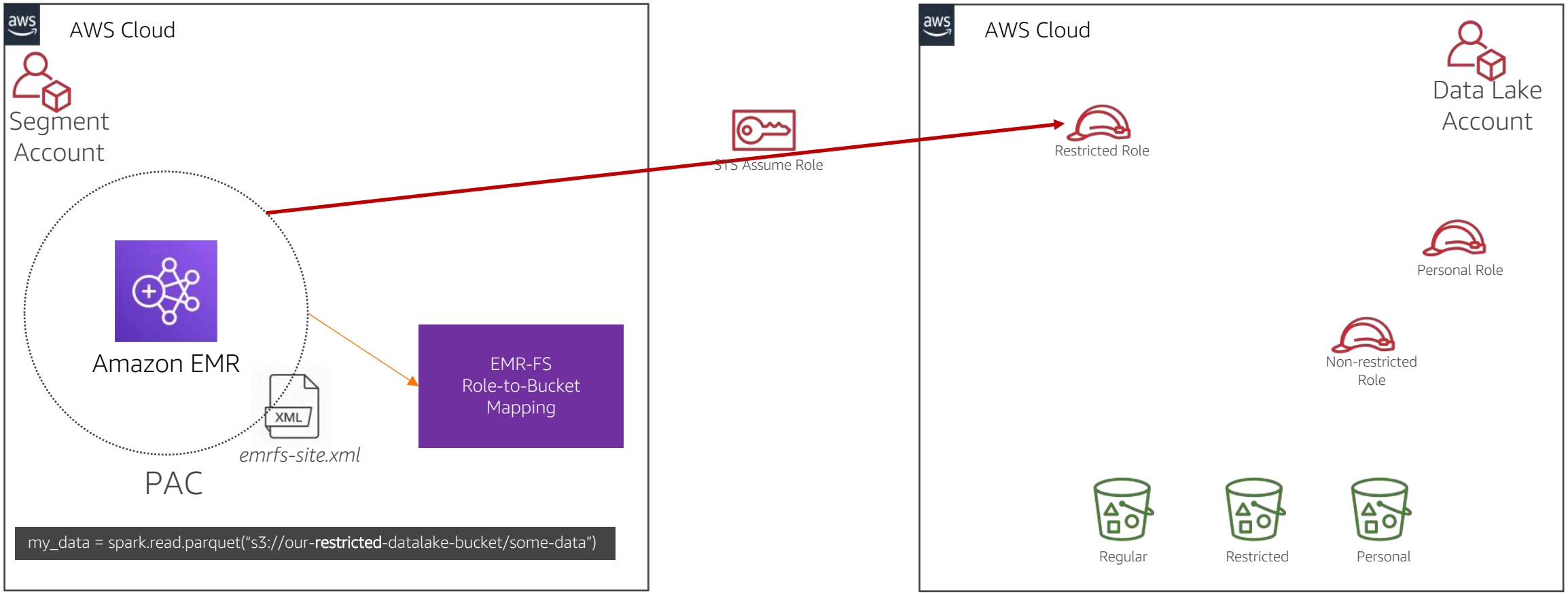




Data Platform

SCOUT 24

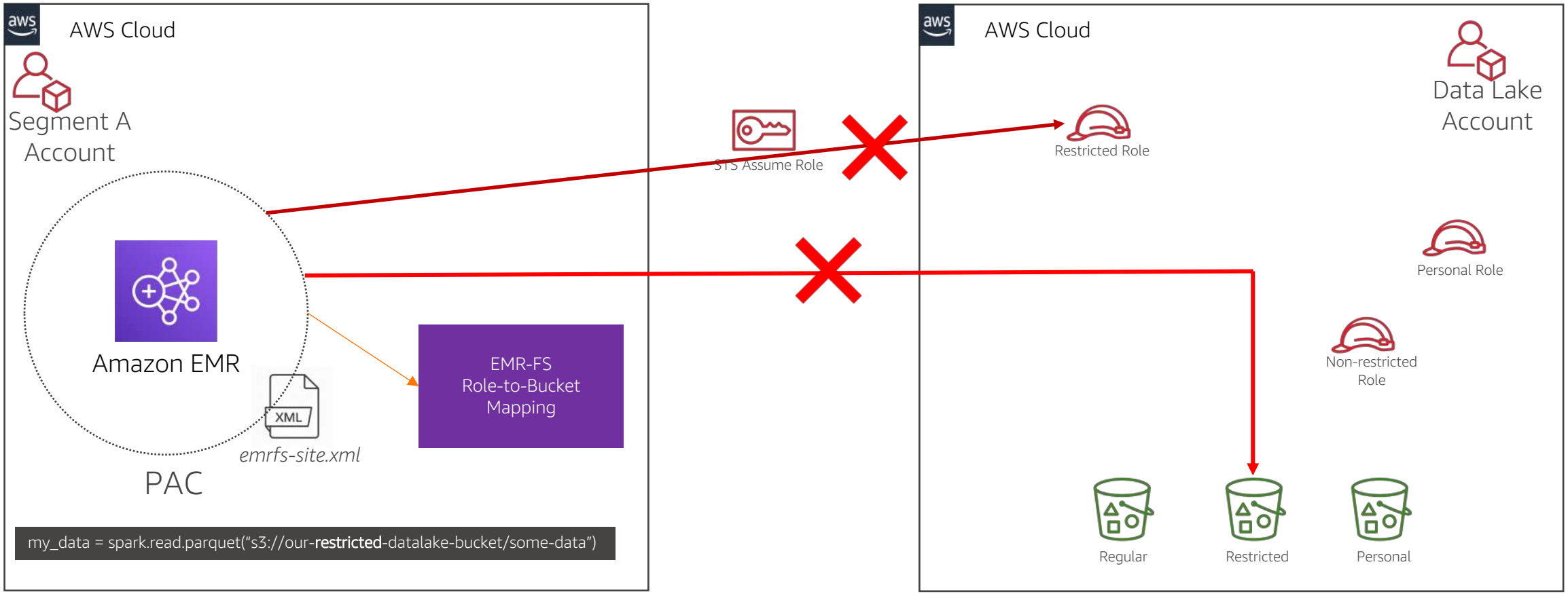
The Personal Analytics Cluster – Data Lake Access



Data Platform

SCOUT 24

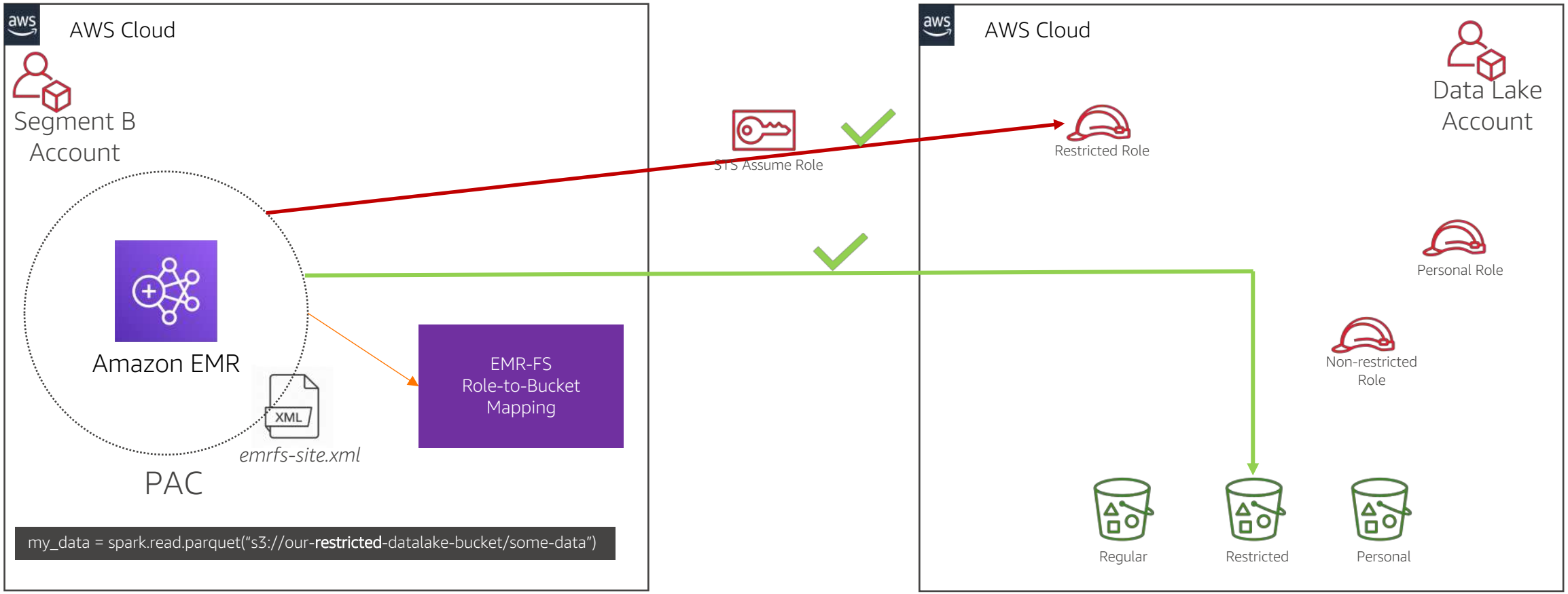
The Personal Analytics Cluster – Data Lake Access



Data Platform

SCOUT 24

The Personal Analytics Cluster – Data Lake Access



Data Platform

SCOUT 24

Data Platform SQL Access

Our Journey to Presto

Data Platform

SCOUT 24

Our Journey to Presto

On EMR

- Cross Account Support (OneScout Hive Metastore)
- Leverages Datalake Access Roles (EMRFS)
- Scheduled Scaling Configurations
- Fits our GDPR Concept (multiple isolated Clusters)

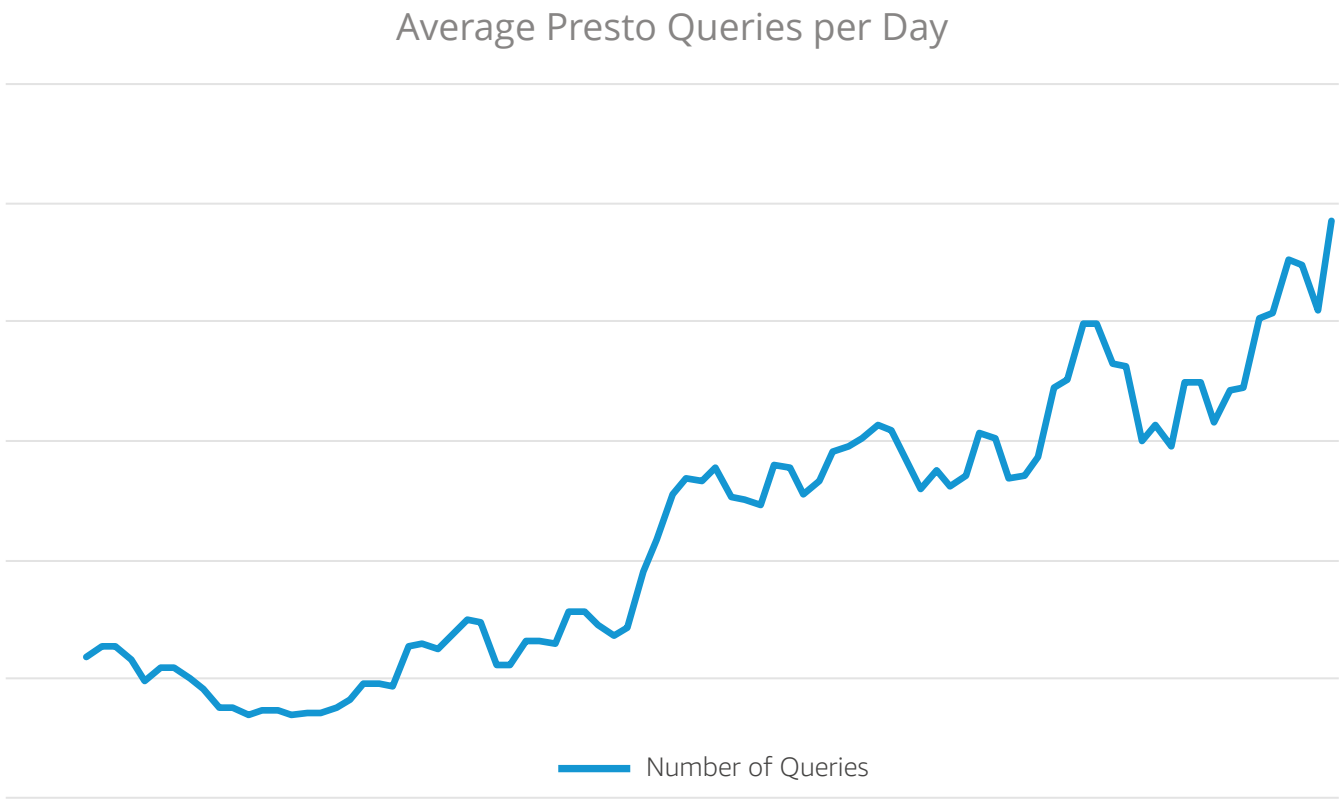
Data Platform

SCOUT 24

Our Journey to Presto

360

Queries per Day



Data Platform

SCOUT 24

We hope to throw out

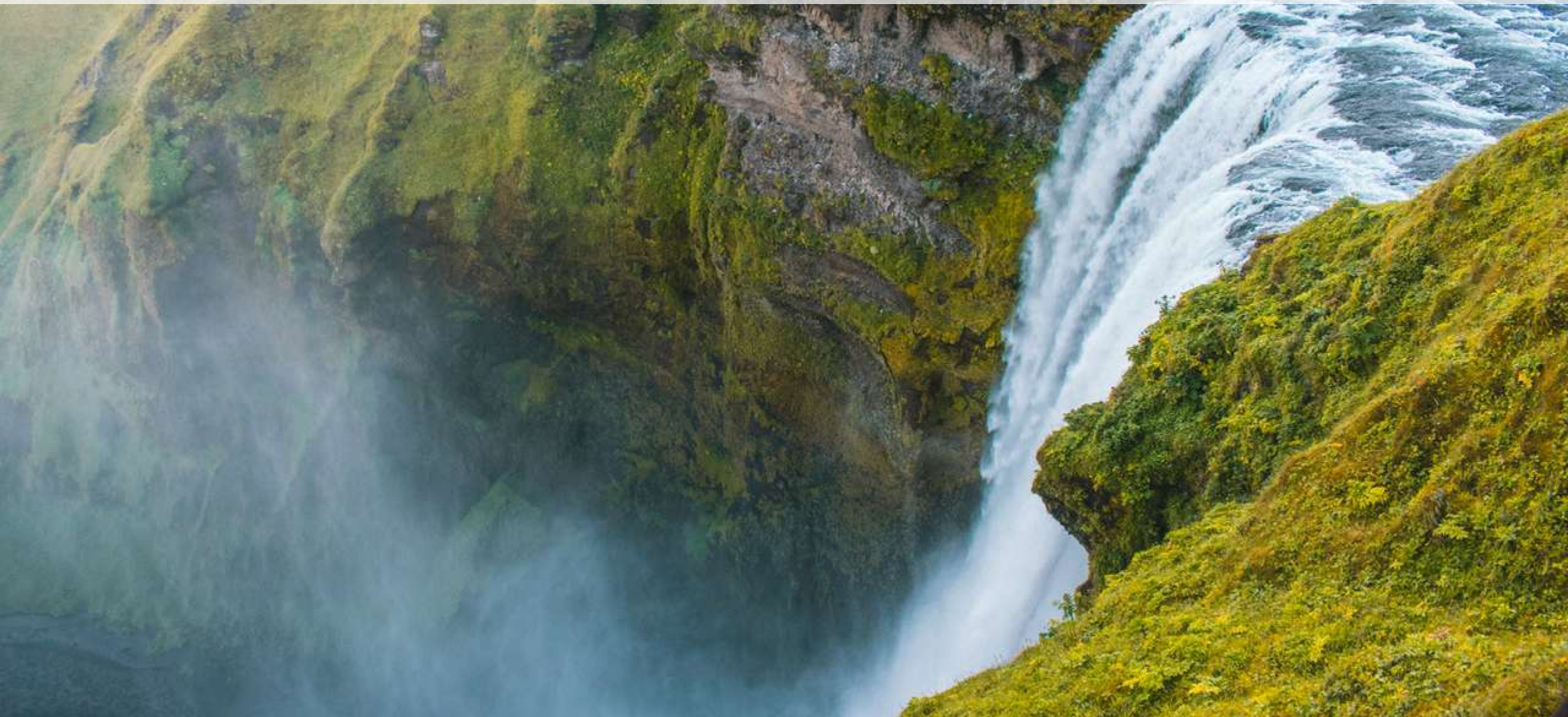
most of the custom components we build.

Our data warehouse was a bottle neck

Data Platform

SCOUT 24

Our data platform holds nothing back



Thank you for your attention!

Christian Dietze

Senior Data Engineer

Scout24 Group

Olalekan Elesin

Data Engineer

Scout24 Group

Raffael Dzikowski

Senior Data Engineer

Scout24 Group