Principal Component Analysis (PCA)

Content Prepared By: Chandra Lingam, Cotton Cola Designs LLC

amazon web services

Principal Component Analysis

Dimension Reduction Technique – retain “information” while reducing features

Real World Datasets have several features that contain similar data – Works great!

New PCA features (*Components*) may be important predictors of target – But…how do you map the “importance” to read-world features?

PCA

Works only for numeric data

Data needs to be normalized – features with similar scale

For large dimension datasets, PCA is an option to reduce features and use that for training a model

Number of Components

Typically, various libraries allow you to specify:

Number of Components

Total Variation to Capture as a percentage (for example capture 90% of the “information”) – in this case PCA will figure out required number of components

PCA on SageMaker

Two Modes

Regular - Good for Sparse Data and Moderate sized datasets

Random – Good for very large datasets – uses approximation algorithm

PCA SageMaker – Data Format

Input:

- csv
- recordio-protobuf

Inference:

- csv
- json
- recordio-protobuf

Demo 1 – Random Data Set

PCA with Random Data set

Show that random data set features cannot be reduced much

pca\ExplorePCA\random_data_pca_exploration.ipynb

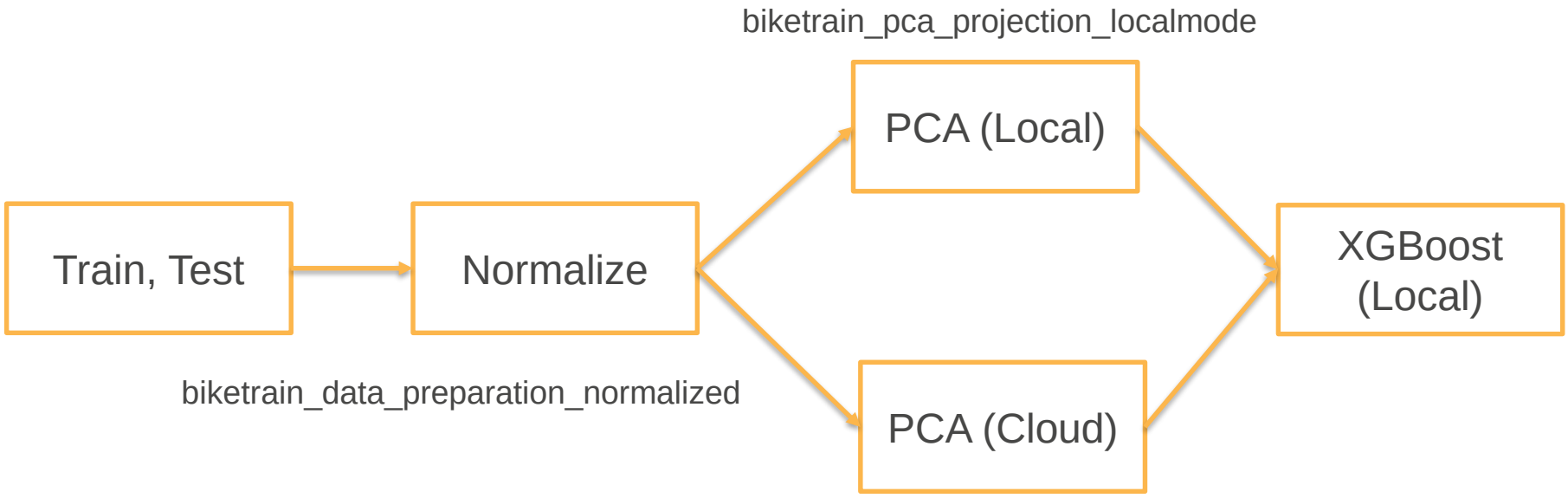
Demo 2 – Correlated Data Set

PCA with correlated data set

PCA can capture substantial amount of information with very few components

pca\ExplorePCA\correlated_data_pca_exploration.ipynb

Demo 3 – Kaggle Bike Train with PCA Components



pca\BikeSharingRegression

Replace: temp, atemp, humidity, windspeed with PCA Components

Factorization Machine (FM)

Content Prepared By: Chandra Lingam, Cotton Cola Designs LLC

amazon web services

Factorization Machine

Factorization Machine algorithm is optimized for handling high dimensional sparse datasets

Supports Regression and Classification

Personalize Content - “predict” ratings/likeness

- Click Prediction for Ad-Placement
- Product recommendation for user
- Movie recommendation
- News/Social Media Feed personalization for users

Factorization Machines

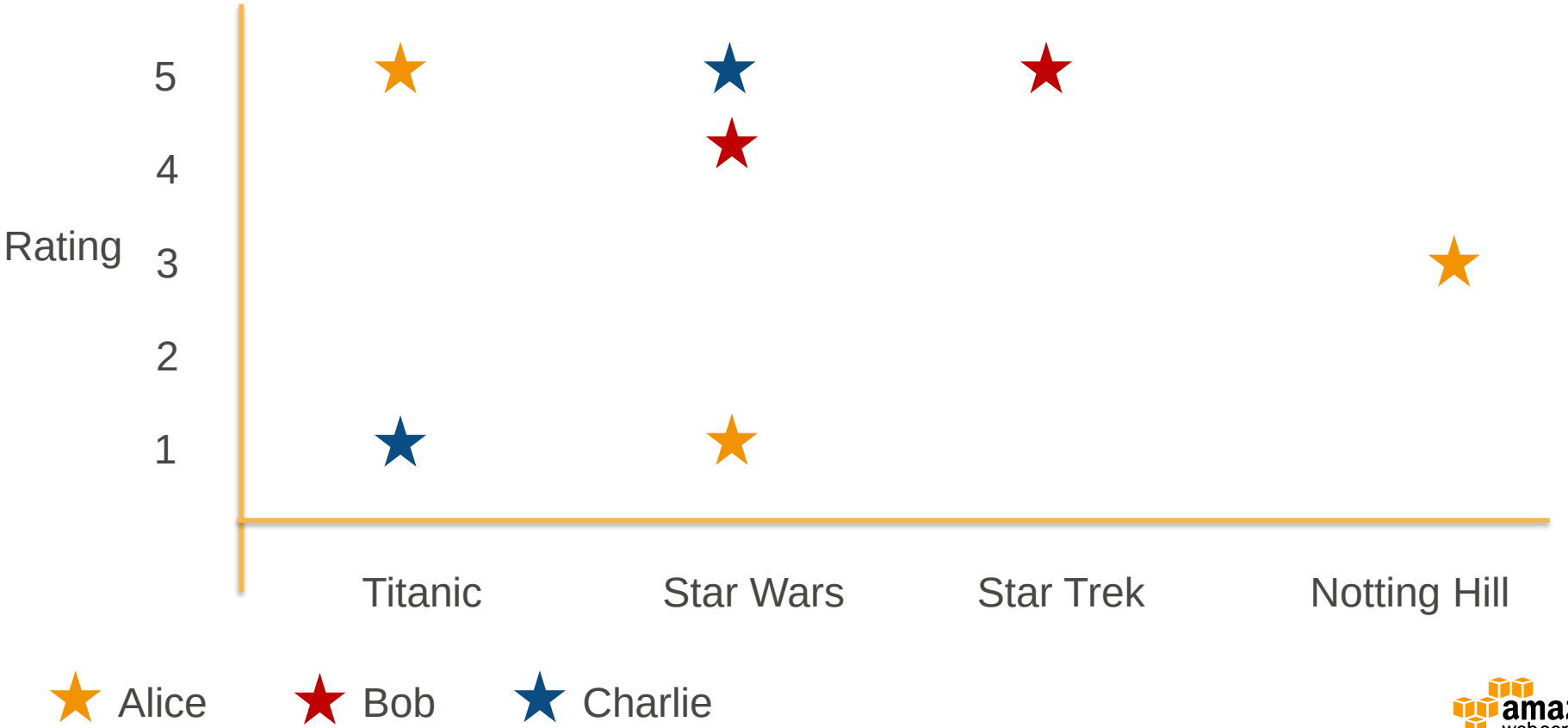
Models all interactions between features using Factorized Parameters

Estimate interactions with very sparse datasets

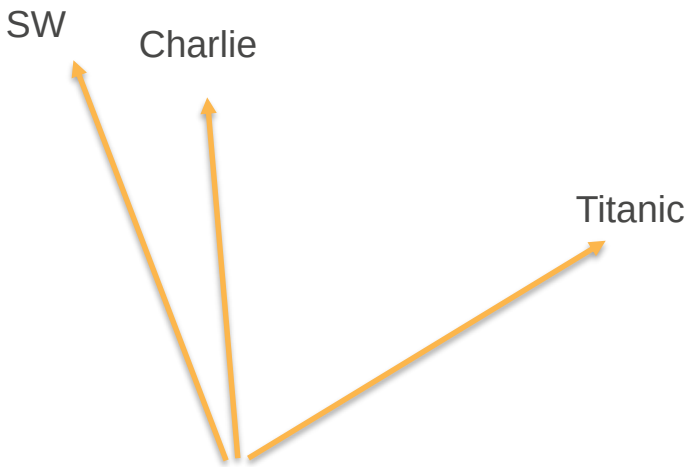
Linear complexity for computing model parameters

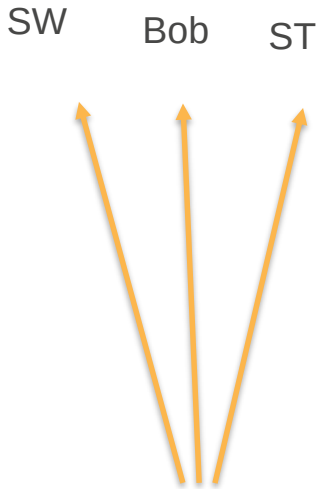
Supports very large datasets

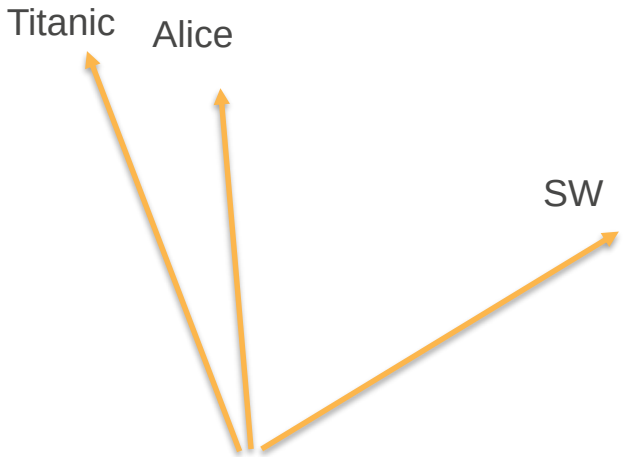
Movie and User



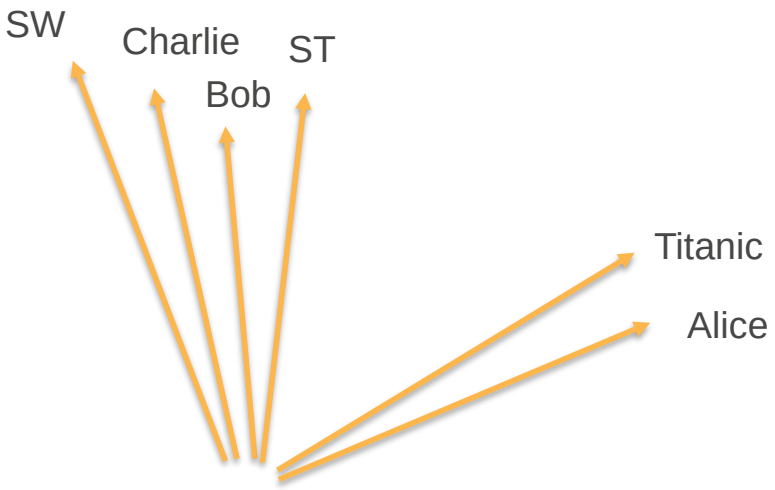
Pair Wise Interaction







Recommendation



Factorization Machine – Data Format

Input:

recordio-protobuf (with Float32 values)

Inference:

json

recordio-protobuf

Demo – Movie Recommendation

Movie Lens Dataset

Predict how a user would a movie

Recommend movies based on user rating, other similar users and other similar movies

fm\movie_data_preparation.ipynb,
fm_cloud_training_template.ipynb,
fm_cloud_prediction_template.ipynb

Demo Movie Recommendation Files

File Name	Purpose
Movies.csv	List of movies [movie id, title, genre]
Ratings.csv	Movies ratings by user [user id, movie id, rating]
Movie_genre.csv	Movies with Genre in separate columns
user_movie_{train|test}.recordio	Sparse RecordIO Train/Test Data – OneHotEncoded [user id, movie id], Rating
user_movie_{train|test}.svm	Sparse SVM Train/Test Data – Easy to read with text editor.
one_hot_enc_movies.svm	List of movie ids and corresponding one hot encoded movie column identifier
one_hot_enc_users.svm	List of user ids and corresponding one hot encoded user column identifier

Useful Resources

Factorization Machines by Steffen Rendle

LibFM Software

Comparison of LibFM Implementations by Alex Rogozhnikov

Collaborative Filtering by Anand Rajaraman

Hyperparameter Tuning

SageMaker

Content Prepared By: Chandra Lingam, Cloud Wave LLC

amazon web services

Hyperparameters

Tunable Parameters of Machine Learning Algorithm

Customize how a model is trained

Improve quality of predictions

Hyperparameter Tuning is Hard

Time consuming Activity

Several hyperparameters and wide range of values

Interdependencies between hyperparameters

Need systematic way to converge to optimal values

SageMaker Automatic Model Tuning

Input:

- Algorithm, Training & Test Data
- Hyperparameters and range of values to search
- Optimization Objective

Tuning:

- Launch multiple training jobs – with different hyperparameters
- Pick the training job and hyperparameters that offer best performance

Search Strategy

Random Search

- Random combination of values from within the range of values configured

Bayesian Search

- Tuning itself is treated as regression problem
- Input features: Hyperparameters
- Target: Optimization Objective

Bayesian Search

Start with some combination of hyperparameter values

Run training jobs with these values

Assess outcome

Use regression to choose next set of hyperparameter values

Repeat

Lab: Tune Factorization Machines

Recommender System to predict how a user would rate movies

TEST RMSE Score degraded from 0.8 to 1.9 with new movie dataset

Factorization Hyperparameters – difficult to understand

Optimize as a black-box

DeepAR Time Series Forecasting

Content Prepared By: Chandra Lingam, Cloud Wave LLC

amazon web services

DeepAR Forecasting

DeepAR is a time series forecasting algorithm

Supervised algorithm based on Recurrent Neural Networks

Typical Use – Forecast:

- Product demand
- Passenger traffic
- Weather trend
- And more

Time component

Almost all data has a time component to it:

ATM transaction, Social Media Post, Doctor appointment, Departure/Arrival Time

Are these considered time series?

Time series

“a series of values of a quantity obtained at successive times, often with equal intervals between them.”

Time Series Forecasting with non-time series algorithms

Do not natively handle Date Time Features

Do not learn from time dependent nature of data – rather it is completely feature driven

DeepAR

Classical forecasting methods (ARIMA or ETS) fit a single model to each time series

DeepAR can fit many similar time series in a single model

DeepAR outperforms ARIMA/ETS on datasets that contain hundreds of related time series

DeepAR Time Series Forecasting

Train model with one or more time series

Use model to extrapolate time series into future

Use model to generate forecasts for a new time series that is similar to the ones it is trained on

prediction_length hyperparameter determines how many points in future needs to be forecast

DeepAR Stationary vs Non-Stationary

Classical Time series forecasting algorithms expect time series to be stationary (mean, standard deviation are constant over time)

- You would need to transform the data to make it stationary

DeepAR does not expect data to be stationary. You can train with data as-is.

- Verify by experiments if stationarity improves model

Training/Test Important Differences

With other algorithms:

- Data is randomly divided into training and test sets
- Training file contains only training data
- Test file contains only test data
- Inference: Model predicts target for new data

DeepAR Training/Test

With DeepAR:

- Time series is split based on time – time order needs to be honored
- Training – Entire dataset except for last prediction_length points
- Test – Entire dataset

DeepAR – Data Format

Input:

- JSON Lines (.json, .json.gz)
- Parquet (.parquet)

Inference:

- JSON

DeepAR Training Input

Field	Description
start	Start Timestamp for the time series Format: YYYY-MM-DD HH:MM:SS
target	Array of floating point or integer values in a time series Supports missing values: “NaN” in JSON Lines, nan in Parquet
dynamic_feat	Optional input features. Floating point or integer values. Array of Arrays. Each inner array represents values of a feature and must be of same length as target Missing values are not supported in the features
cat	Optional category. Category is used for identifying/encoding a time series. Each categorical feature is represented as 0 based integer. If you are using categories, then training set must have timeseries for all categories. DeepAR can forecast only for categories that were trained with.

DeepAR Inference Format

Field	Description
instances	Corresponds to time series that should be forecast
start	Start Timestamp for the time series Format: YYYY-MM-DD HH:MM:SS
target	Array of values. Time series for which to forecast
dynamic_feat	Field should be included only if model was trained with features. You must provide dynamic features for each value in target and for future timepoints Length(dynamic_feat) = length (target) + prediction_length
cat	Field should be included only if model was trained with categories.

Demo – Time Series

Time series with Pandas

Detecting missing time steps

Handling missing values

Demo - Overview

- Kaggle Bike Rental as a time series forecasting problem
- How to handle missing values in time series
    - Training Data consists of two years worth of hourly rental
    - Gaps in time series - Training data consists of first 19 days of each month
- How to handle missing features

Demo

Demo 1 - Train only with target time series

Demo 2 - Add Categories

Demo 3 – Add Dynamic Features

Note: Demos focus on how to get things done with DeepAR. Emphasis is not on best Kaggle scores

AWS Provided Examples: Synthetic Data, Electricity Forecast

Demo 3 – Dynamic Features

Requires More Powerful Training instance

Trained with ml.c5.4xlarge instance (16 CPU, 32 GB RAM)

Training with ml.m4.xlarge (free-tier) or ml.m5.xlarge – Out of memory exception (4 CPU, 16 GB RAM)

For Endpoint, you can still use ml.m4.xlarge instance

Total Cost: USD 0.50 (for training job) + USD 0.20 (endpoint – free tier eligible)

amazon web services

DeepAR Hyperparameters

DeepAR Hyperparameters

Parameters with greatest impact

Useful Resources

Beginner's Guide to create a Time Series Forecast by Aarshay Jain

Handling Missing Values in Time Series For Beginners by jingjuewang

Pandas: Working with missing data

DeepAR: How it works

DeepAR: Modeling non-stationary data Q&A

Random Cut Forest (RCF)

Random Cut Forest

Unsupervised algorithm to detect outliers or anomalous data points

Tree based ensemble method

Support for Timeseries data

Assigns an anomaly score for each data point

RCF Uses

Traffic spike due to rush hour or accident

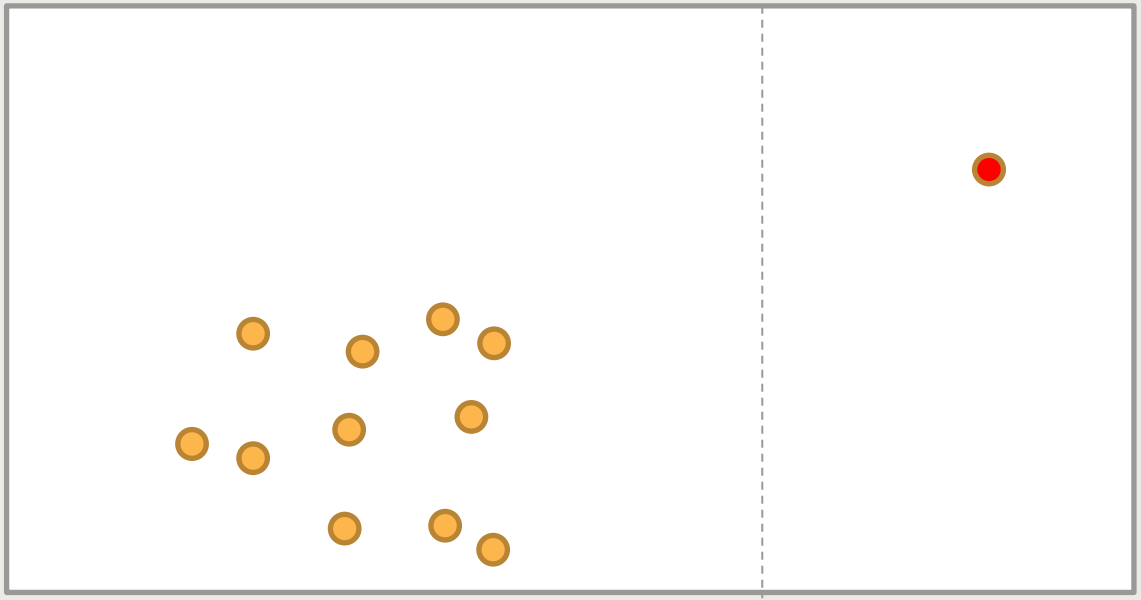
DDoS attack detection

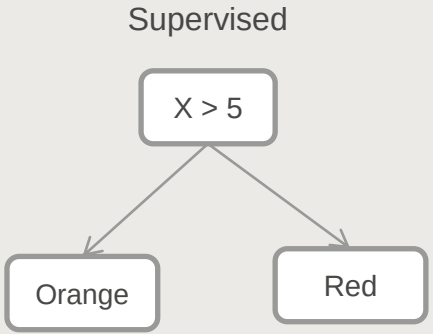
Unauthorized data transfer detection

Intuition

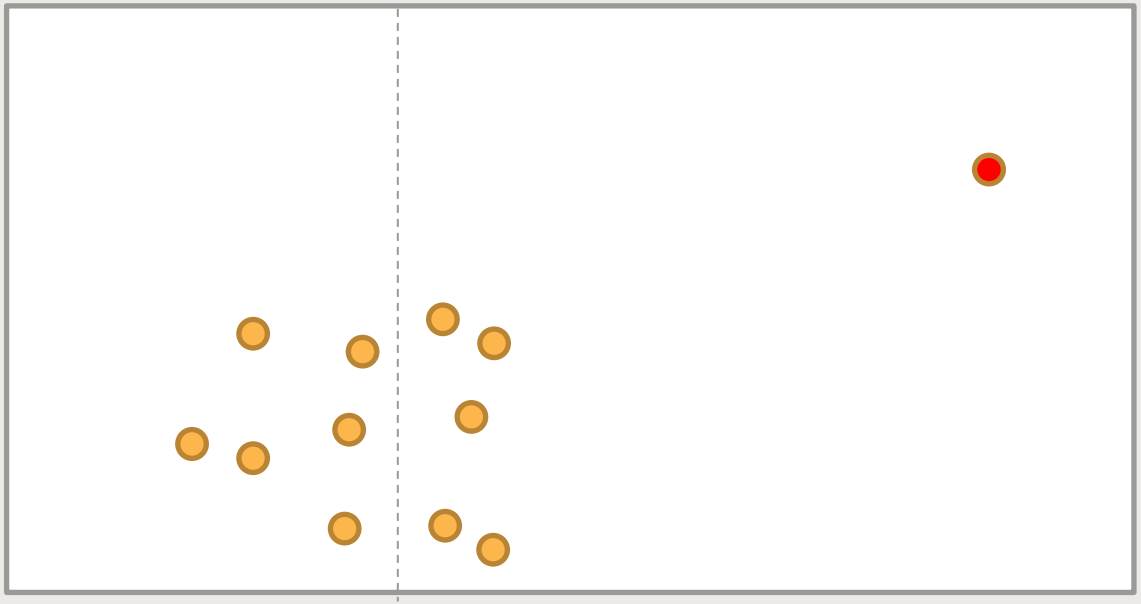
Using Isolation Forest

Tree based Classifier

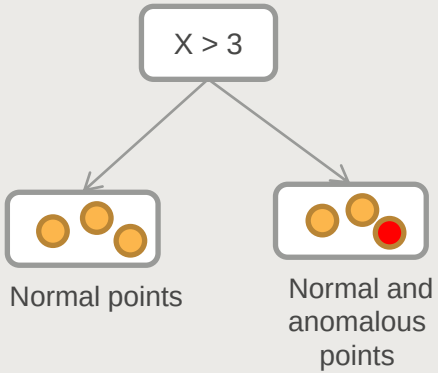




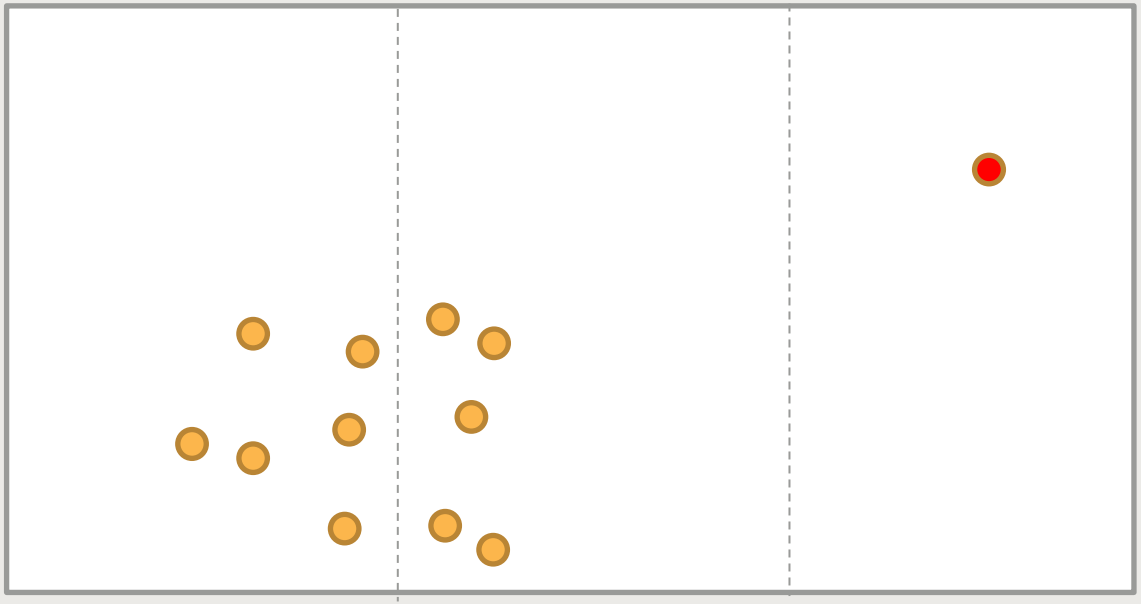
Anomaly Detection – Random Cut



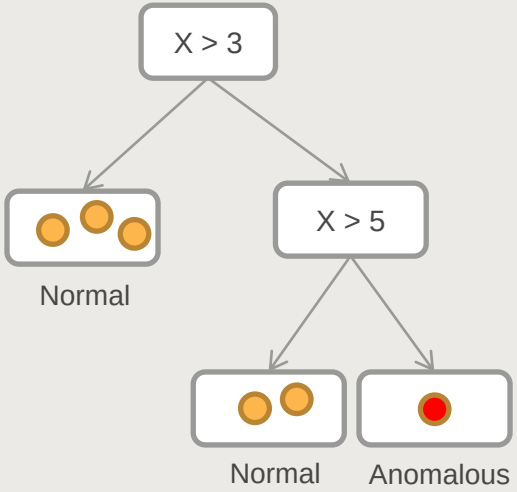
Unsupervised – Explain the data



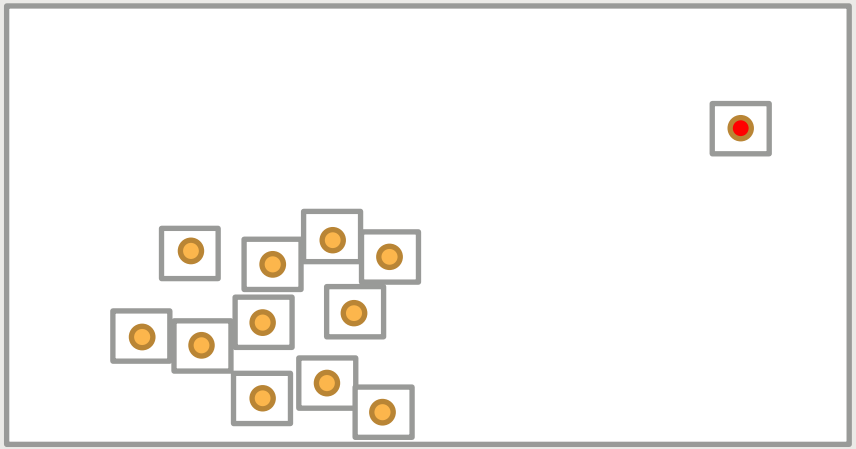
Anomaly Detection – Random Cut



Unsupervised – Explain the data



Anomalous points are closer to root (depth)

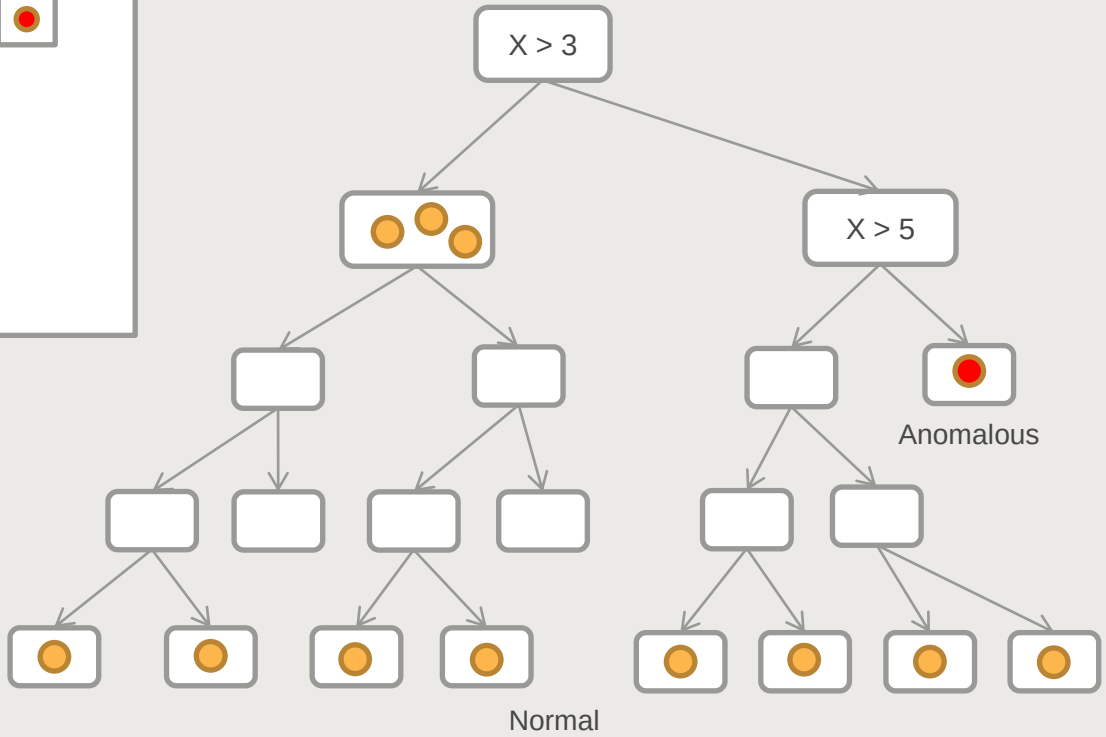


Normal data points require a lot of splits

Anomalous points require fewer splits

Each point is assigned an anomaly score based on depth
Normal points = low score
Anomalous points = high score

Unsupervised – Explain every single point



Anomaly Detection with Isolation Forest

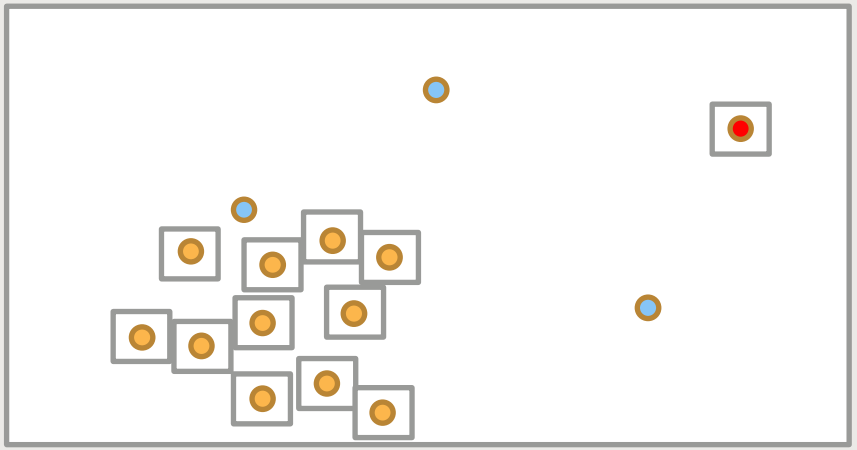
"*if I have a set of data points along the line and I choose an arbitrary split, there is going to be empty spaces between adjacent instances. And the algorithm allocates two additional patterns for that empty space*"

Dr. Thomas Dietterich

Anomaly Detection: Algorithms, Explanations, Applications | Microsoft Research

https://youtu.be/12Xq9OLdQwQ (40:00)

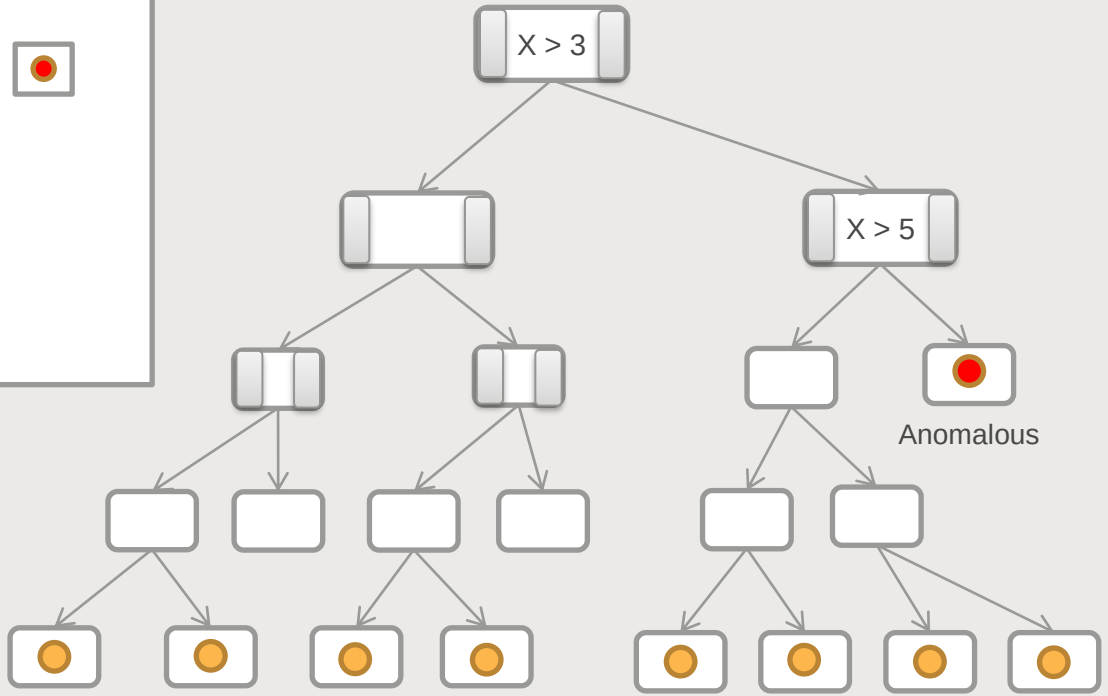
What about new data points?



Tree covers the gaps!

We need to define a threshold to classify if a score is anomaly or normal

Additional patterns to cover for adjacent gaps



Useful resources – Isolation Forest

These resources helped me gain insight into Random Cut Forest (you don’t have to watch it; this is an acknowledgement of people who helped me)

- Elena Sharova: Unsupervised Anomaly Detection with Isolation Forest | PyData 2018 - https://youtu.be/5p8B2Ikcw-k
- Jan van der Vegt: A walk through the isolation forest | PyData 2019 https://youtu.be/RyFQXQf4w4w
- Dr. Thomas Dietterich - Anomaly Detection: Algorithms, Explanations, Applications | Microsoft Research 2018 https://youtu.be/12Xq9OLdQwQ

Random Cut Forest

- Build several trees (forest)
- Each tree is a given several random sample of instances drawn from the training dataset
- RCF uses reservoir sampling to draw random samples from large dataset
    - Works efficiently when size of the data set is too large to fit in memory
    - Or when we don’t know the training set size
- Final Anomaly score = Average of anomaly scores of all trees

Random Cut Forest Prediction

RCF predicts an anomaly score for the data point

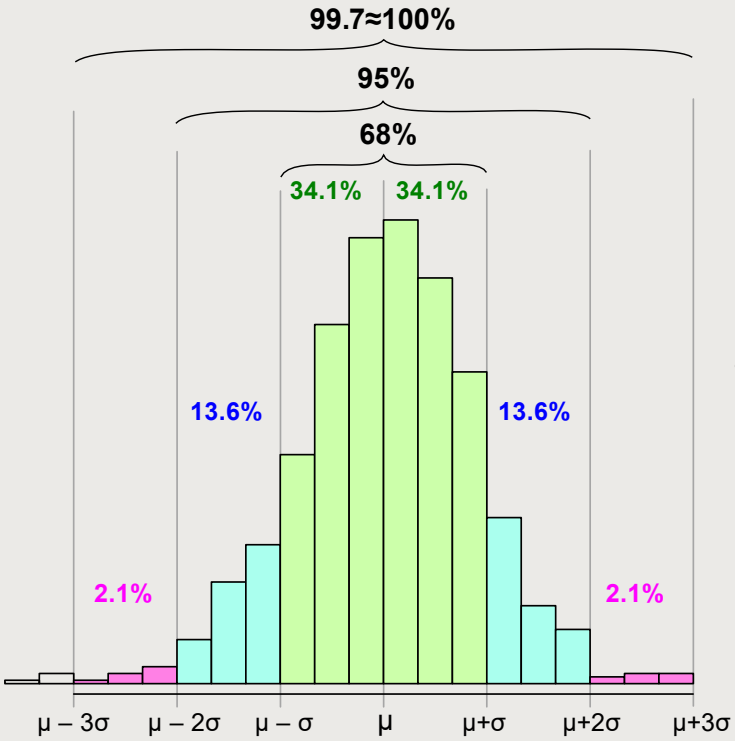
- Score varies inversely with depth
- Low Score is considered “normal”
- High Score indicates “anomaly”

Definition of Low and High score depends on application

Common practice: Scores beyond three standard deviations from mean score are considered anomalous

Reference: https://docs.aws.amazon.com/sagemaker/latest/dg/randomcutforest.html

Distribution (68-95-99.7 rule)



"For approximately normal dataset, 99.7% of the datapoints fall within three standard deviations from mean"

By Melikamp - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=65001875
https://en.wikipedia.org/wiki/68%E2%80%9395%E2%80%9399.7_rule

RCF Supported Data Formats

- Training, Test channels - CSV, RecordIO
- Test – Optional. First column in each row represents the anomaly label
    - “1” – anomalous data point
    - “0” – normal data point
    - RCF computes accuracy, precision, recall, F1-score for test data
- Inference format: JSON, CSV, RecordIO

RCF Hyperparameters

Hyperparameter	Description
feature_dim	Number of features in the data set. SageMaker RCF Estimator automatically computes this
eval_metrics	Test data evaluation metrics. Default: accuracy, precision, recall, f1 score
num_trees	Number of trees in the forest
num_samples_per_tree	Number of random samples given to each tree from the training set

num_trees, num_samples_per_tree are tunable parameters using automatic hyperparameter tuning

Lab – Taxi Passenger (AWS Example)

Analyze anomalies in NY Taxi usage timeseries data

Optimization Techniques: Shingling, number of trees, sample size, cutoff for anomaly score

Measuring performance: Labeled Test Data (binary classification)

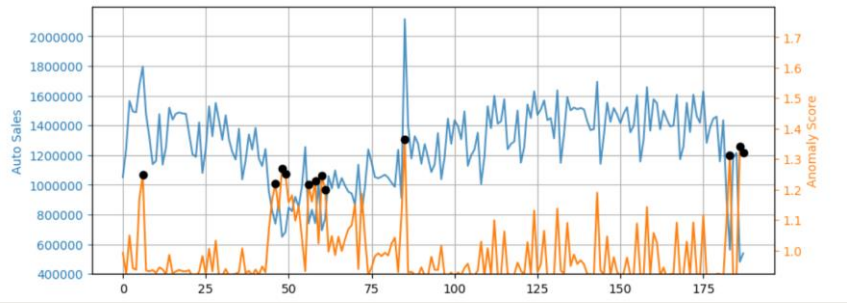
https://github.com/awslabs/amazon-sagemaker-examples/blob/master/introduction_to_amazon_algorithms/random_cut_forest/random_cut_forest.ipynb

Lab – Auto Sales Analysis

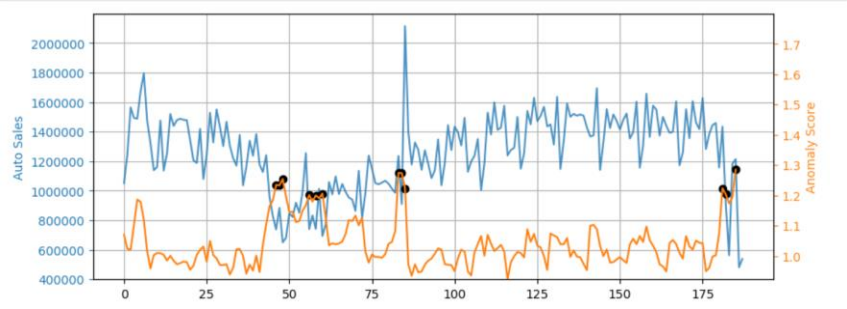
- Analyze 15 years of monthly auto sales in the USA
- Verify how RCF Score varies for change in volume:
  - Housing Crisis
  - Recovery
  - COVID
- Data Source:
  https://www.goodcarbadcar.net/usa-auto-industry-total-sales-figures/,
  http://www.bea.gov/
- Shingle sizes: 1-month, 3-months, 12-months

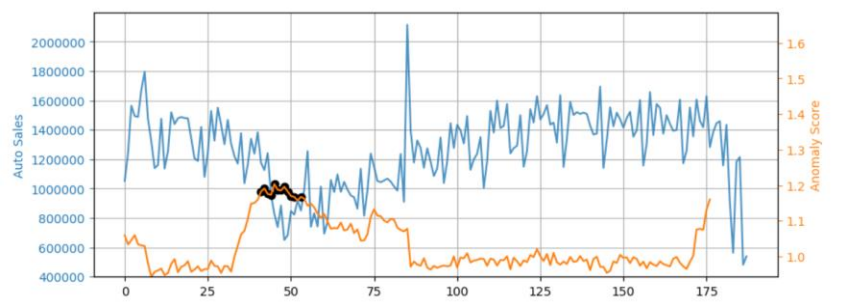
Auto Sales – RCF Anomaly Scores

Shingle size = 1

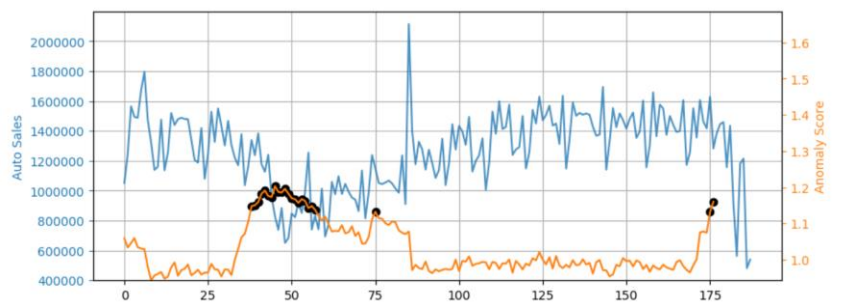


Shingle size = 3





Shingle size = 12. Cutoff = 2 SD



Shingle size = 12. Cutoff = 1.5 SD

Chandra Lingam

57,000+ Students

For AWS self-paced video courses, visit:

https://www.cloudwavetraining.com/

Cloud Wave LLC

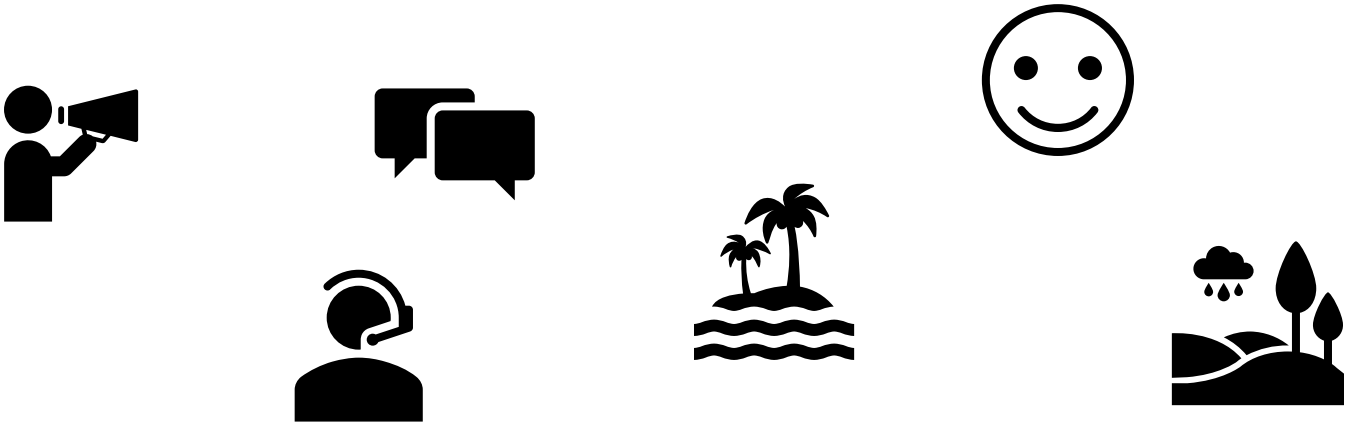
Amazon Artificial Intelligence

Analyze Voice, Text, Video, Image

Chandra Lingam

Cloud Wave LLC

Natural Language, Image, Video Analysis



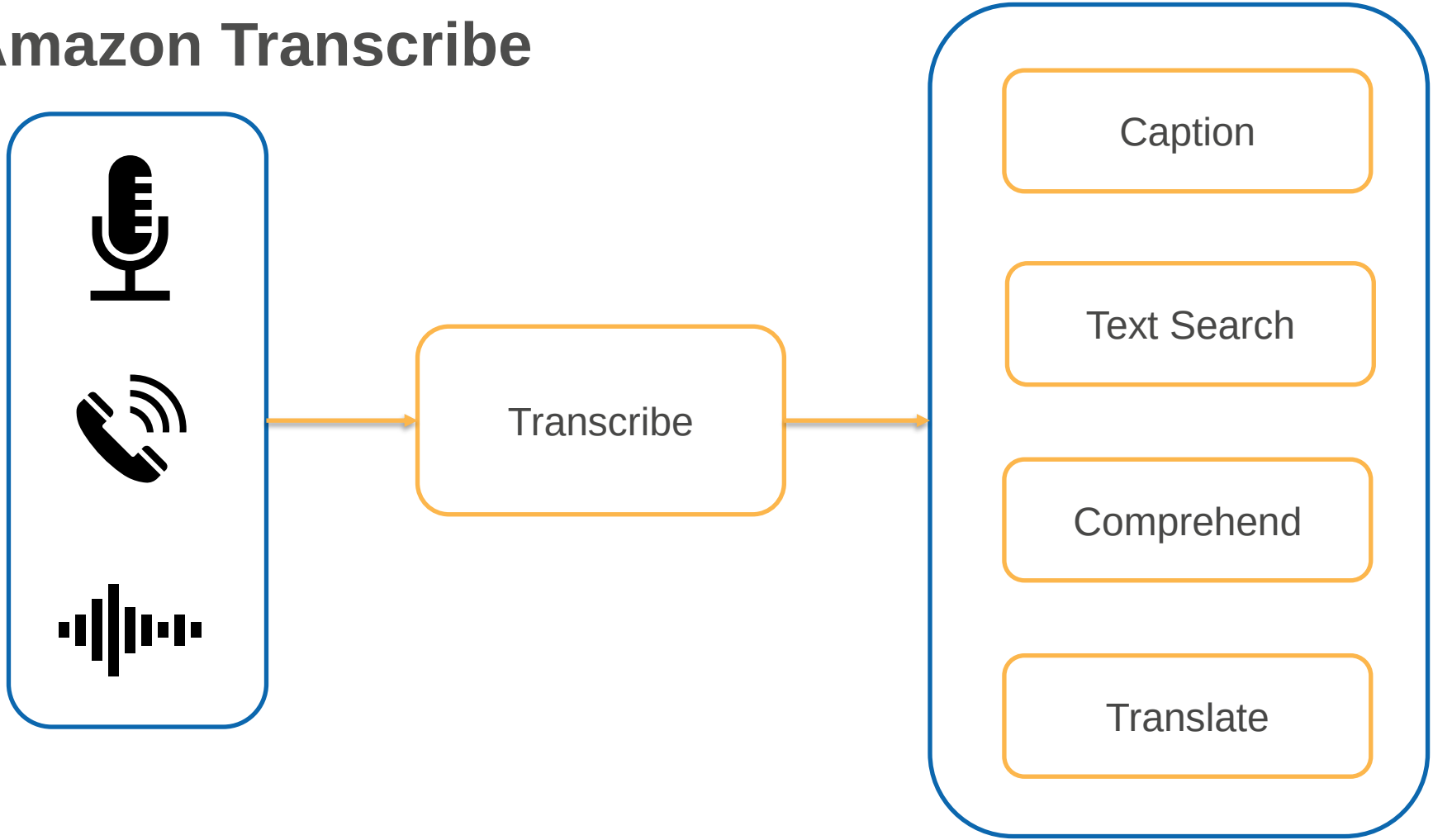
Service	Purpose	Use
Transcribe	Voice to Text	The first step in analyzing voice is to convert to text and then use text based algorithms
Translate	Text to Text - Language translation	Understand text/comment in other languages, Localization of your products to international markets
Comprehend	Analyze text to understand intent	Sentiment analysis (Positive, Negative, Neutral, Mixed), classification, data extraction, organize text by topics
Polly	Text to Voice	Speech enabled products
Lex	Conversation using voice and text	Alexa style interaction. Build your own self-service applications that can understand what customer is asking for using voice or text
Rekognition	Image and Video analysis	Detect objects, person, emotions, text, inappropriate content and track people
Textract	Extract text and data from any document	Convert Scanned documents, extract fields and forms,
DeepLens	Deep Learning enabled Video Camera for developers	Learning tool for building vision enabled applications

Amazon Transcribe

“Audio data is virtually impossible for computers to search and analyze. Therefore, recorded speech needs to be converted to text before it can be used in applications.”

Reference: https://aws.amazon.com/transcribe/

Amazon Transcribe



Transcribe Features

Automatically adds punctuation and formatting

Batch and Real-time transcription

Timestamp for each word – easily locate in original media

Custom Vocabulary Support

Detect and Tag multiple Speakers

Lab – Sample Audio File

“XGBoost routinely finds a top place in machine learning competitions. So, let's take a quick tour of XGBoost and compare it with a linear model and other tree-based methods. This spreadsheet contains the bike rental count prediction data set provided by Kaggle.”

Transcribe Issues

Actual

“XGBoost routinely finds a top place in machine learning competitions. So, let's take a quick tour of XGBoost and compare it with a linear model and other tree-based methods. This spreadsheet contains the bike rental count prediction data set provided by Kaggle.”

Transcribed

“exhibition is vertically finds the top place in machine learning competitions. So let's take a quick tour off extra booze and compare it with a lady moral and other tree bees methods. This spreadsheet contains the bike rental contradiction, data said, perturbed by cargo.”

Custom Words

“These [custom vocabulary] are generally domain-specific words and phrases, words that Amazon Transcribe isn’t recognizing, or proper nouns.”

Reference:

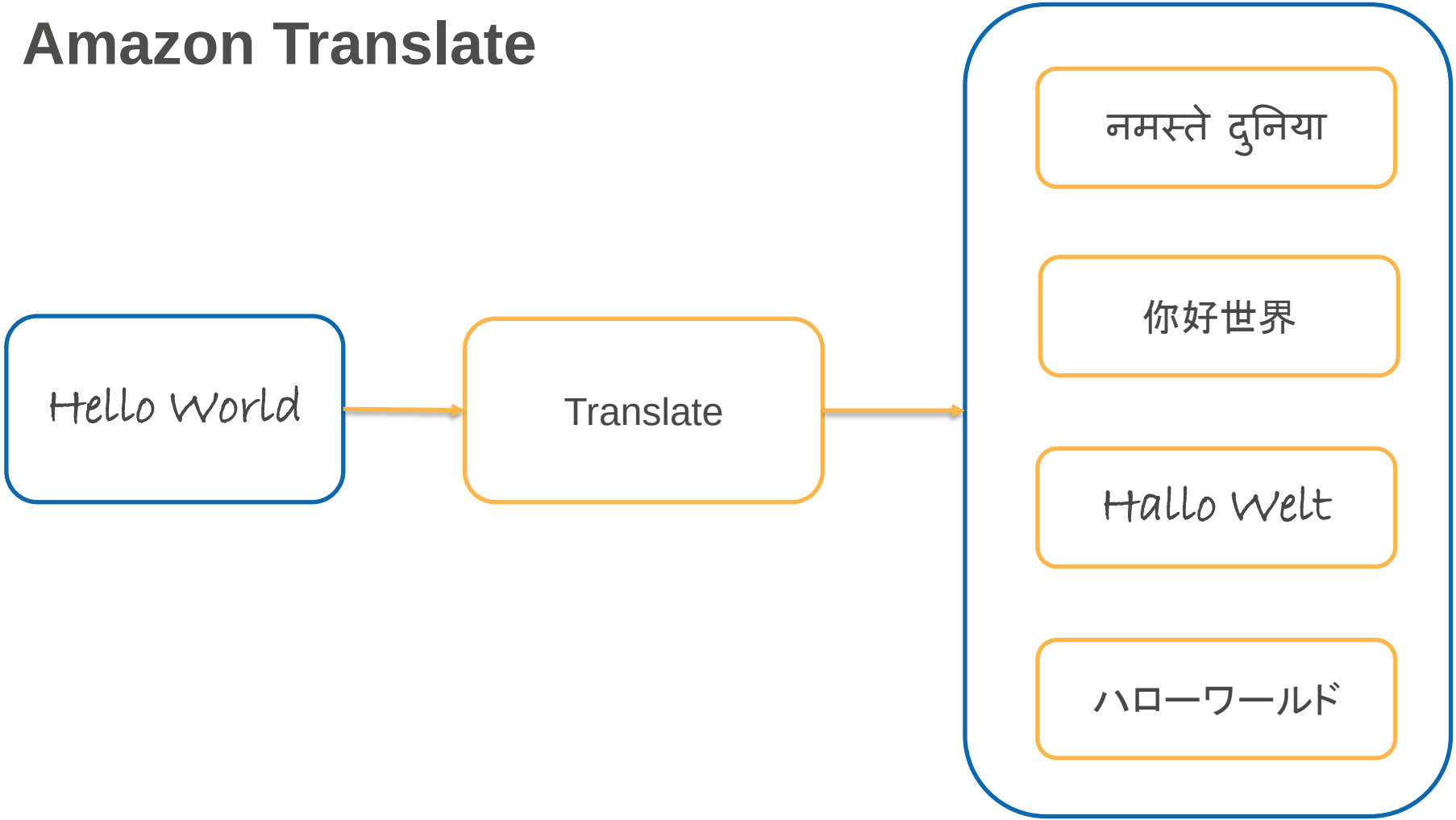
https://docs.aws.amazon.com/transcribe/latest/dg/how-vocabulary.html?shortFooter=true

Custom Words

List [of words]

Table [of words with phonetics, output hints]

Amazon Translate



Amazon Translate

Understand reviews, text, comment in other languages

Localization of your products and services

Custom Terminology Support

Real-time and Batch translation

Amazon Translate

"*Neural network based - takes into account the entire context of the source sentence as well as translation it has generated so far, to create a more accurate and fluent translation*"

Reference:

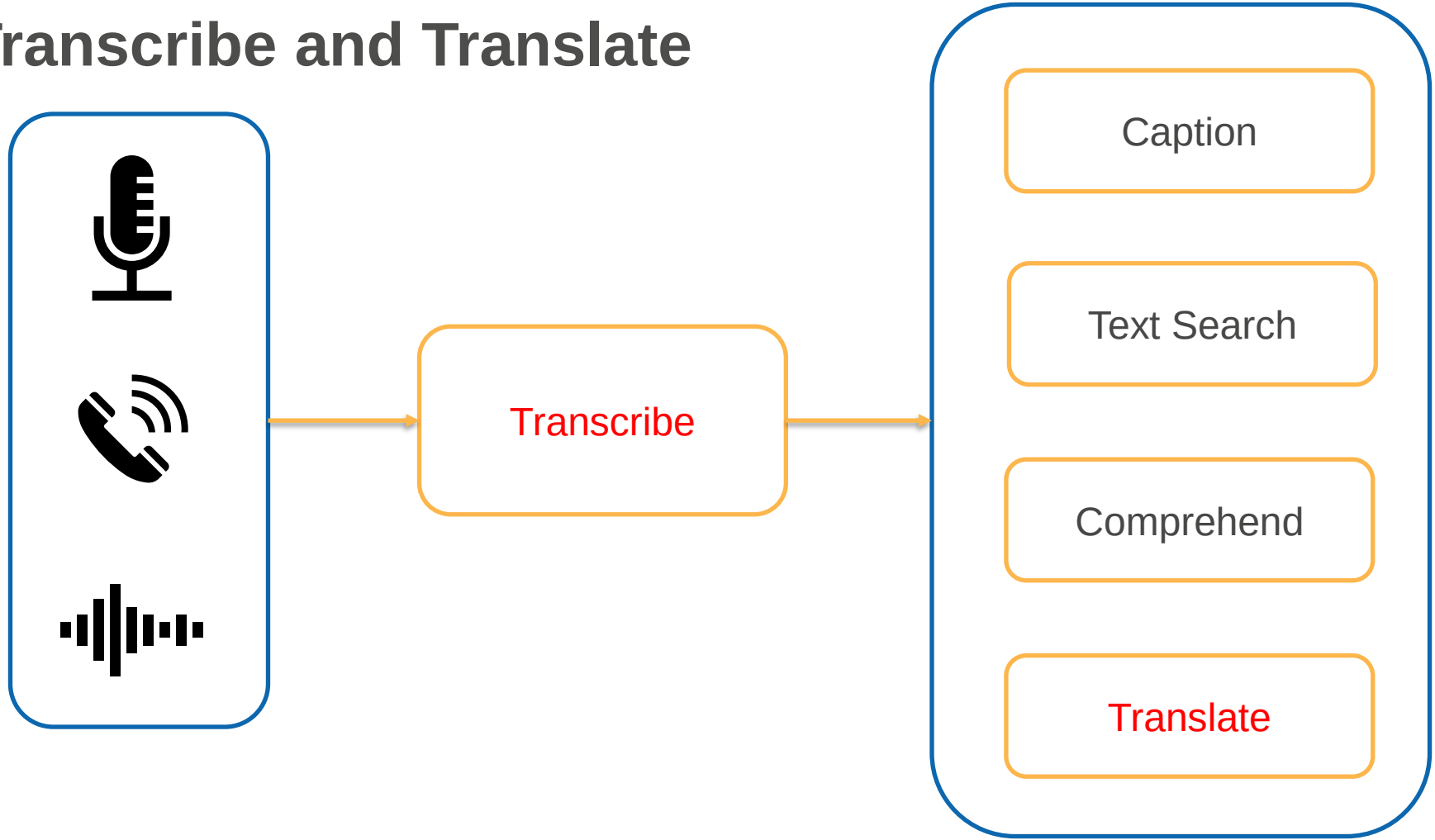
https://aws.amazon.com/translate/details/

Lab – Translate [English to Hindi, Spanish]

Sourcel Text

“XGBoost routinely finds a top place in machine learning competitions. So, let's take a quick tour of XGBoost and compare it with a linear model and other tree-based methods. This spreadsheet contains the bike rental count prediction data set provided by Kaggle.”

Transcribe and Translate

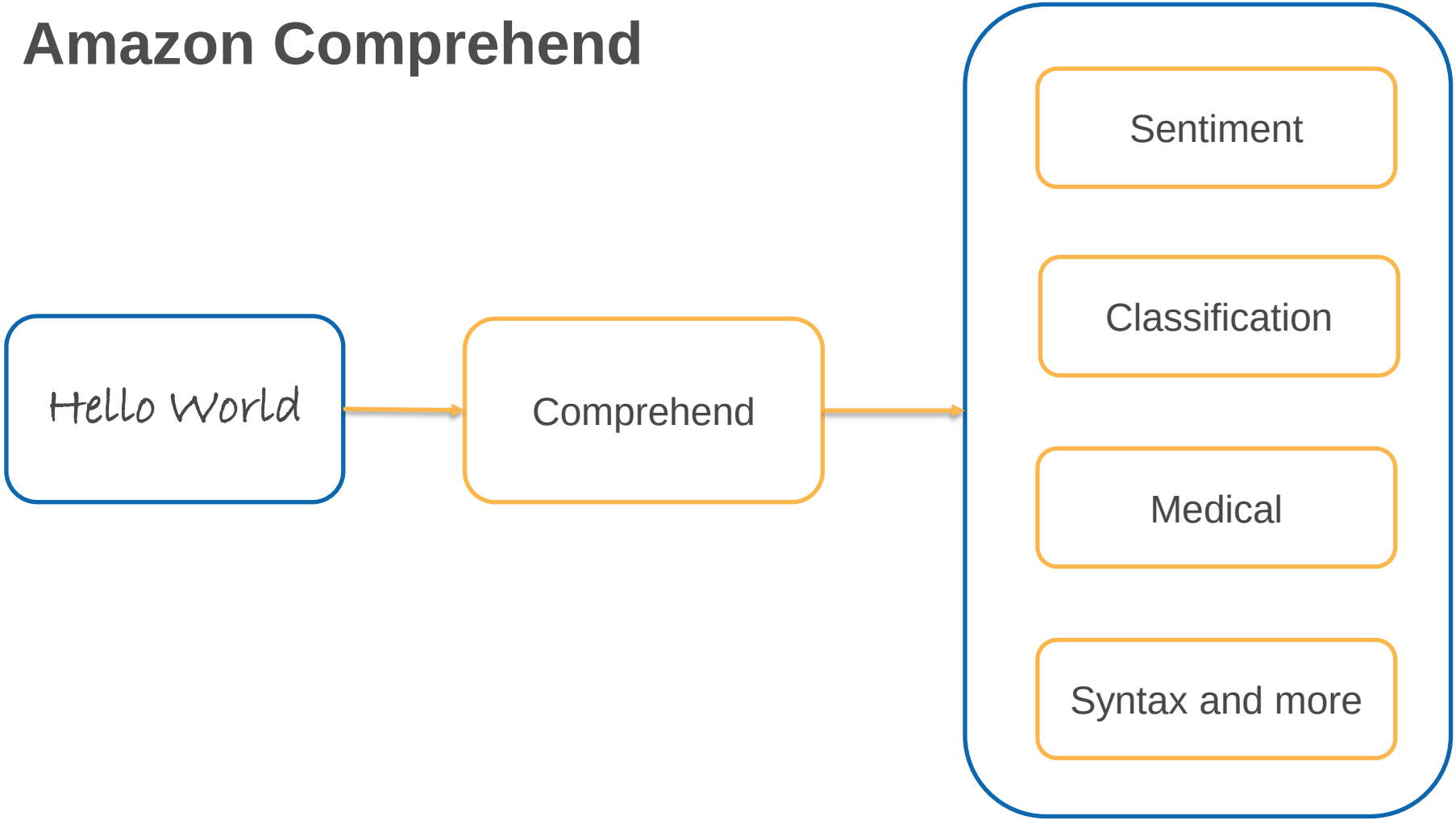


Lab 2 – Translation with Errors

Source Text

“exhibition is vertically finds the top place in machine learning competitions. So let's take a quick tour off extra booze and compare it with a lady moral and other tree bees methods. This spreadsheet contains the bike rental contradiction, data said, perturbed by cargo.”

Amazon Comprehend



Amazon Comprehend Features

Keyphrase Extraction

Sentiment Analysis

Syntax Analysis

Entity Recognition

Medical Information Extraction

Custom Entities

Classification

Topic Modeling

Comprehend Medical

“Amazon Comprehend Medical is a service that detects useful information in unstructured clinical text”

“As much as 75% of all health record data is found in unstructured text: Physician’s note, discharge summaries, test results, case notes and so on.”

Reference:

https://docs.aws.amazon.com/comprehend/latest/dg/comprehend-med.html

Classification Using Comprehend

Text based custom classification

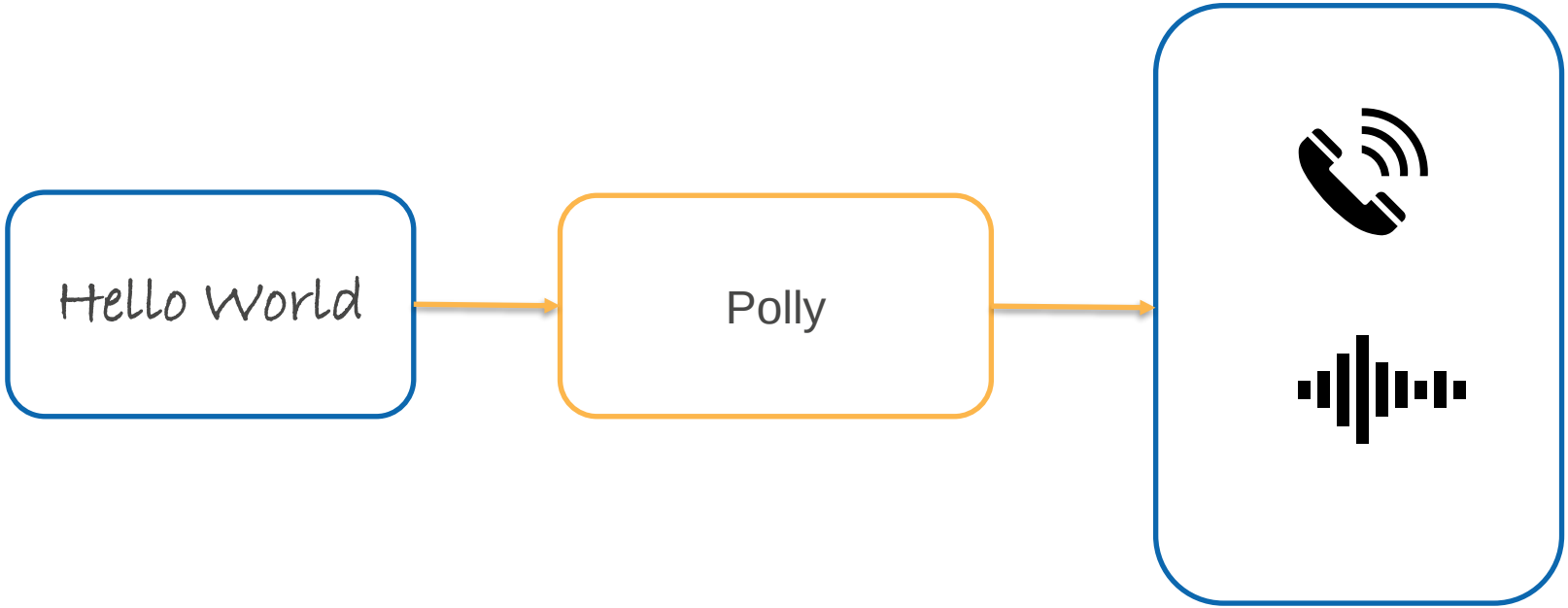
- Moderating Comments and Reviews
- Organizing Documents by Category

Lab – Classification using Comprehend

Objective: Identify tweets that require follow-up

Twitter Classification Dataset [AWS provided labeled dataset]

Amazon Polly



Amazon Polly - Text To Speech Service (TTS)

Speech Enabled Products

Real-time or batch mode

Wide selection of voices, languages

Customize with SSML (Speech Synthesis Markup Language)

Polly SSML

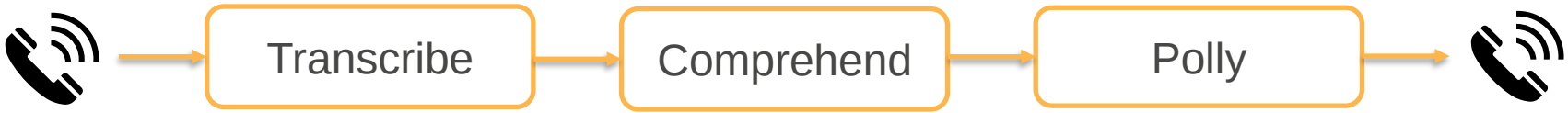
Control how Polly generates speech from text

Adjust pause, speech rate, pitch

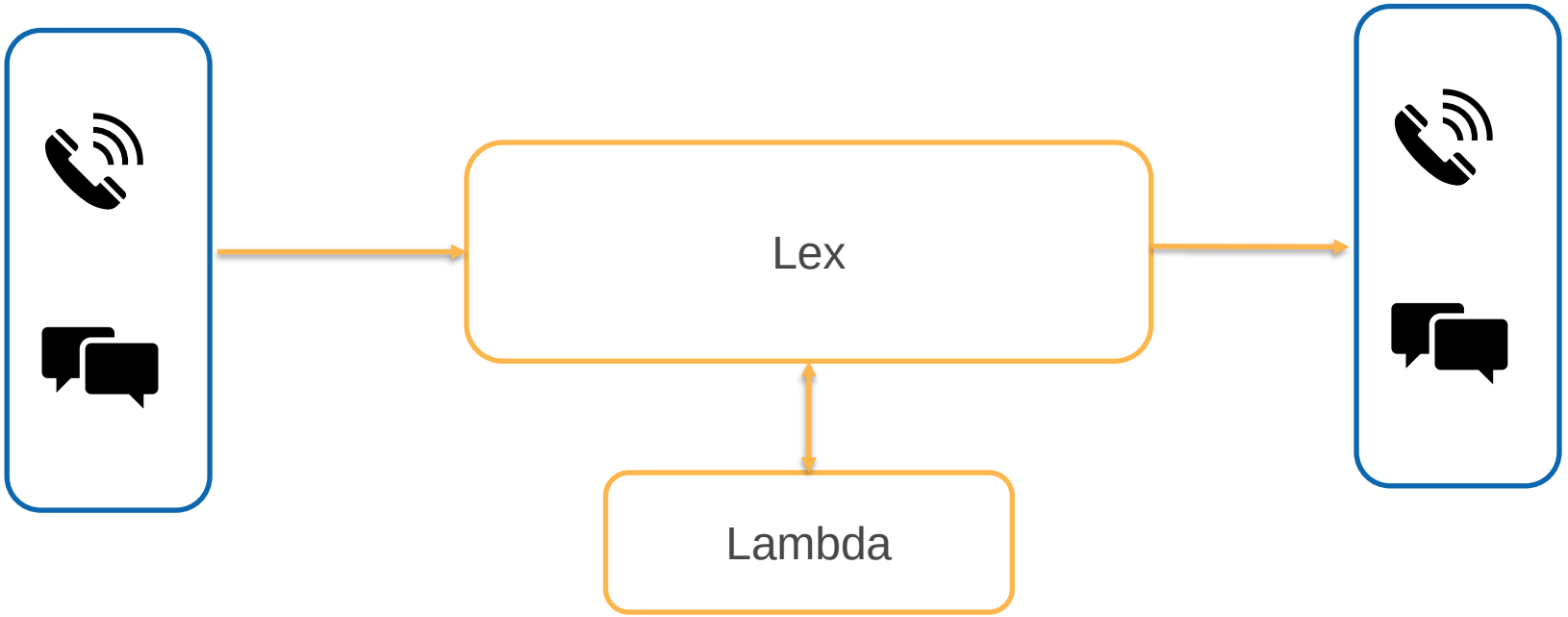
Emphasis of sounds, Phonetic pronunciation

Newscaster

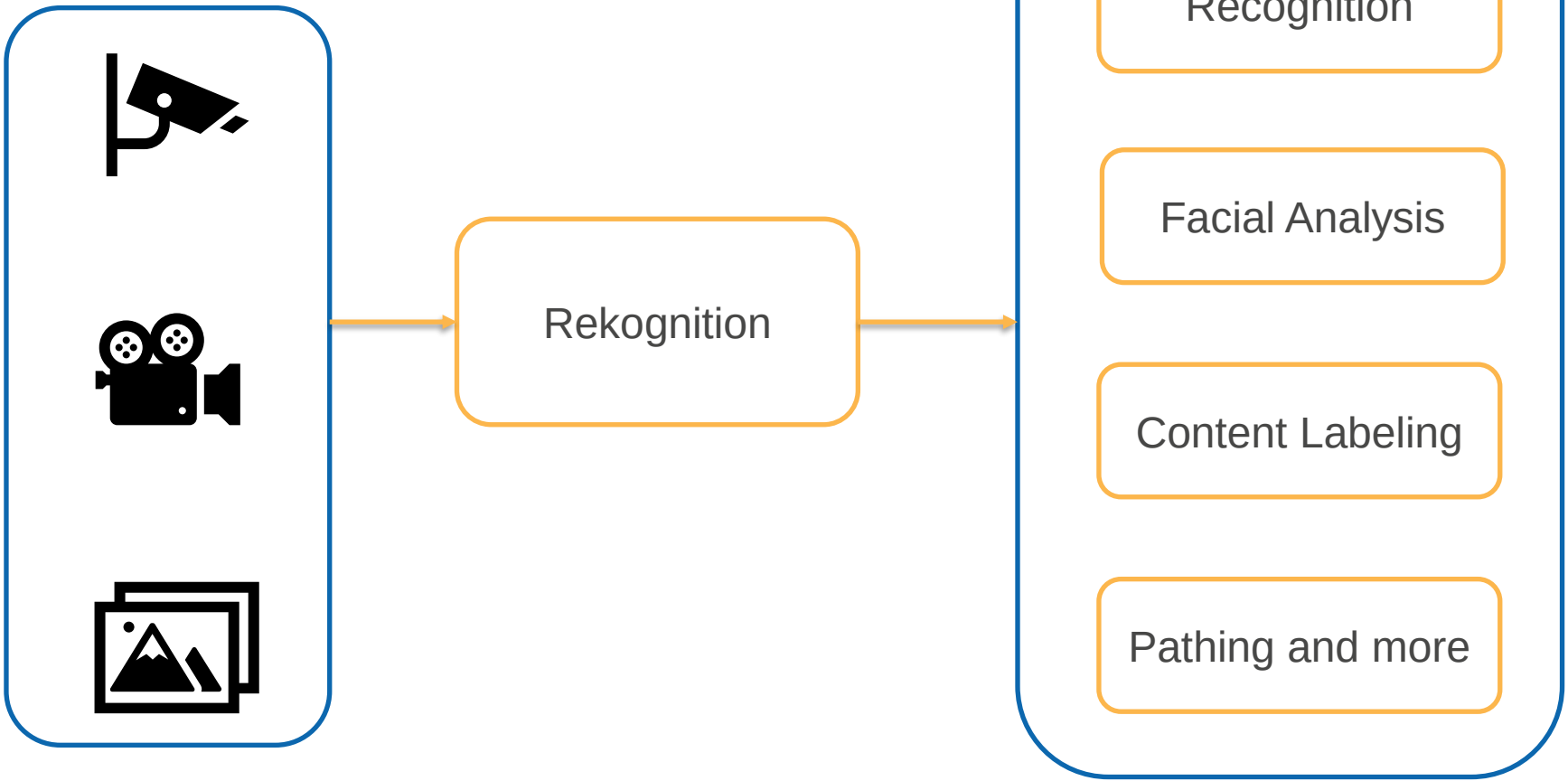
Amazon Lex - Motivation



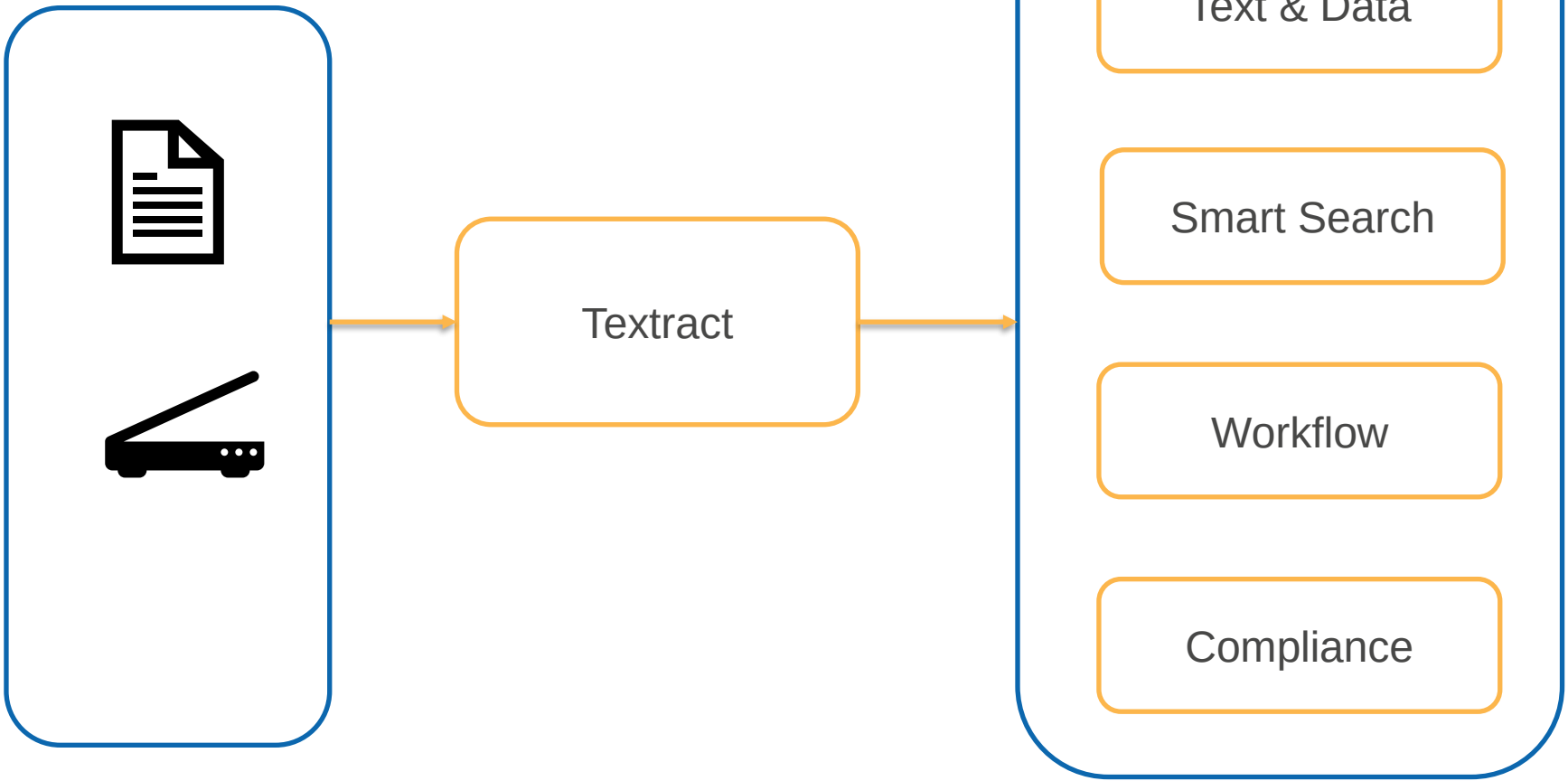
Amazon Lex



Amazon Rekognition



Amazon Textract



Data Lake in AWS

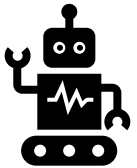
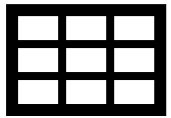
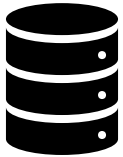
Storage, Data Governance, Analytics

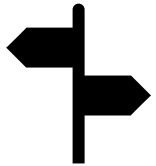
Chandra Lingam

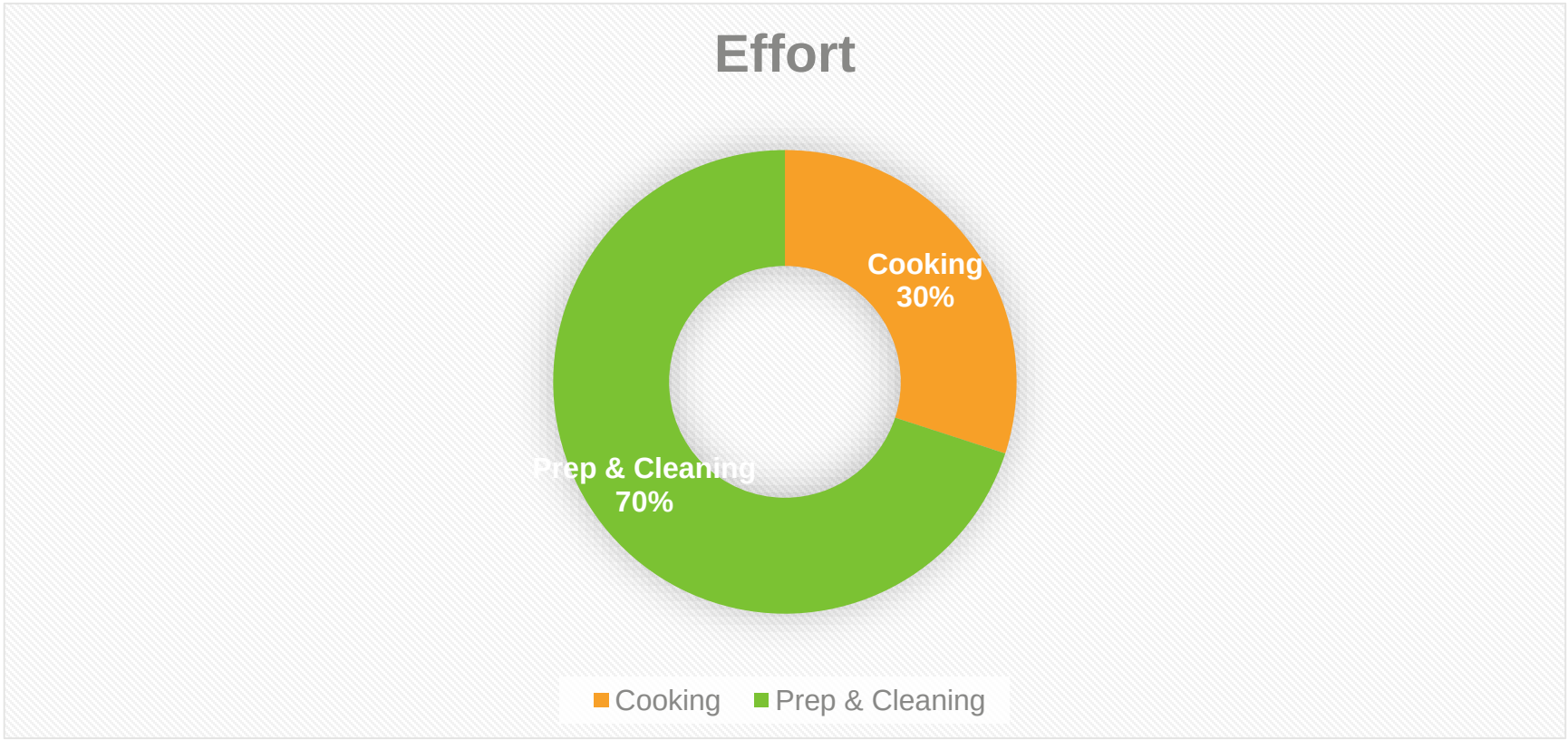
Cloud Wave LLC

Data Lake Motivation



Data Lake

Streamline Data Management

Date Lake Vs Data Warehouse

"A data lake is a vast pool of raw data, the purpose for which is not yet defined. A data warehouse is a repository for structured, filtered data that has already been processed for a specific purpose."

Reference: Talend, https://www.talend.com/resources/data-lake-vs-data-warehouse/

Data Lake

“A data lake is a centralized repository that allows you to migrate and store all structured and unstructured data at unlimited scale…”

Reference: AWS, https://aws.amazon.com/products/storage/data-lake-storage/infographic/

AWS - Whitepaper

1. Storage
2. Governance
3. Analytics

Data Lake on AWS:
https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/building-data-lake-aws.html

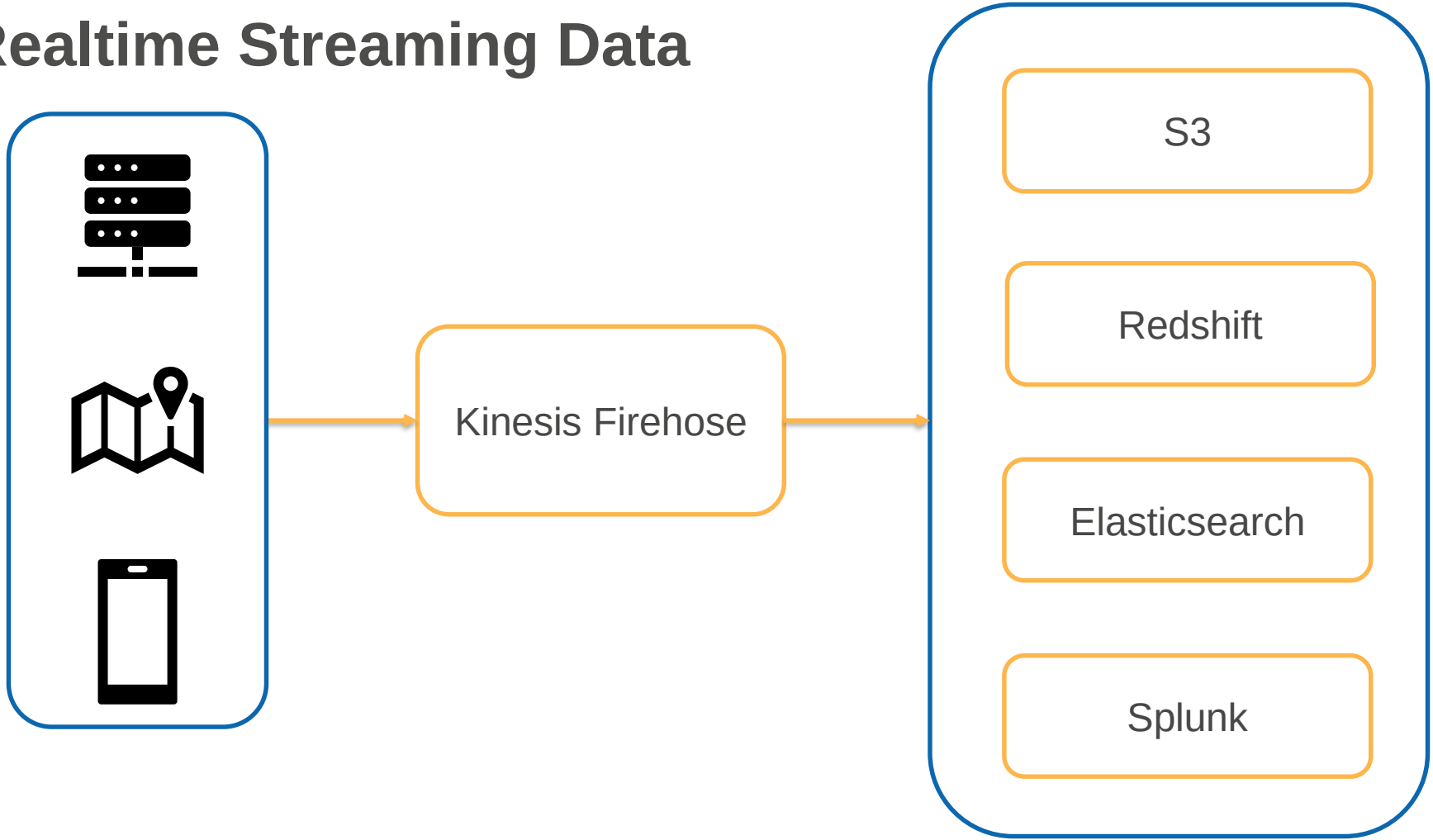
Storage

Service	Purpose	Use
S3	Storage (Exabyte scale)	Object Storage to store and retrieve any amount of data. Cost effective with 99.999999999% (11 9s) of durability Object Life cycle management and Tiered storage based on access patterns
Glacier	Backup and Archiving (Exabyte scale)	Backup and Long term archival (multi-year) at extremely low cost and 11 9s durability.

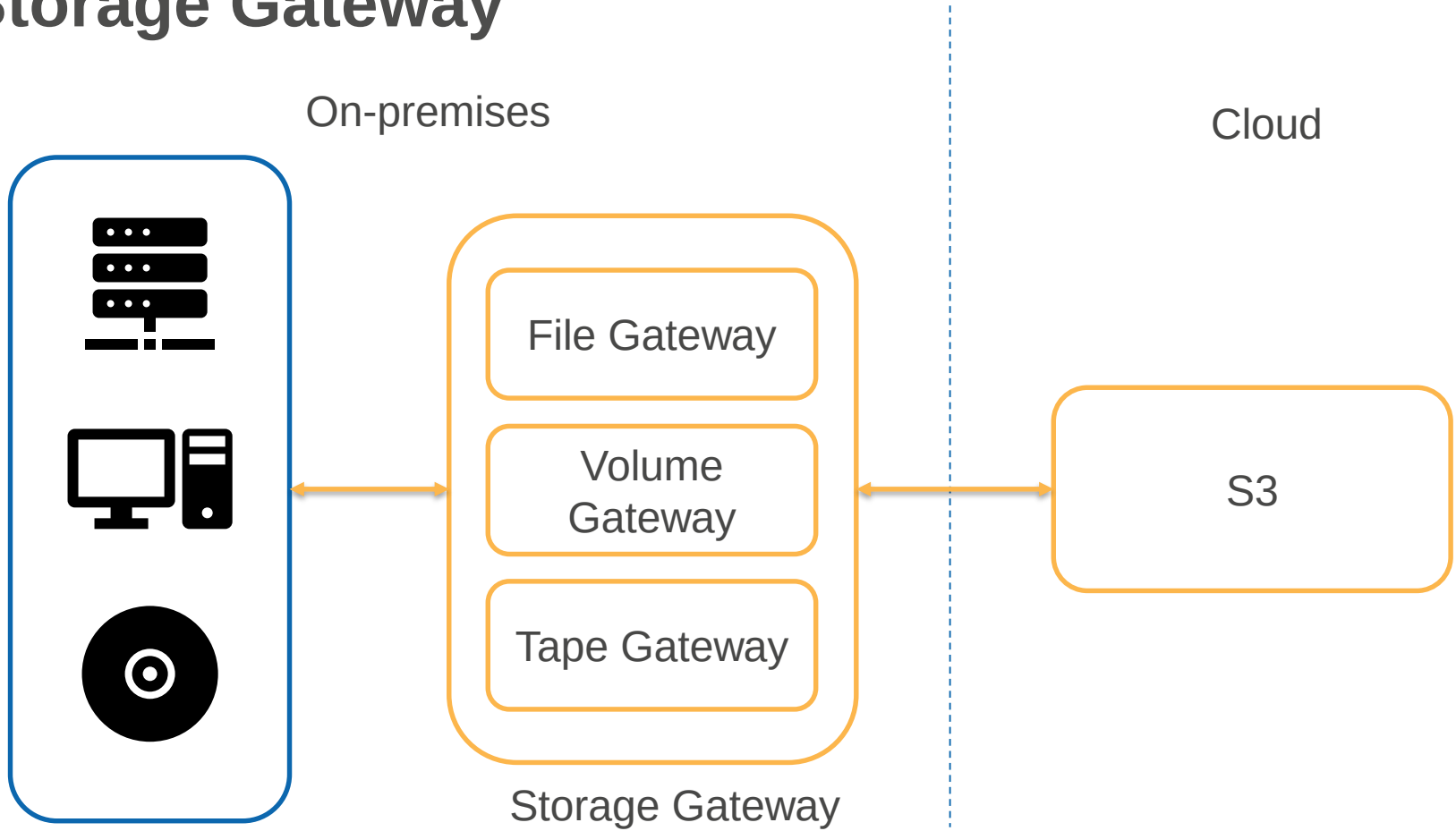
Ingestion

Service	Purpose	Use
Kinesis Firehose	Real-time Streaming Data Ingestion	Capture and deliver real time streaming data directly to S3 and other destinations like Redshift, Elasticsearch, Splunk
Storage Gateway	Hybrid Cloud Storage	Integrate legacy on-premises data processing platforms to S3 Data Lake. Files, volumes and tape backups
Snowball, Snowmobile	Very Large Data Transfer	Appliance to move petabytes to exabytes of data to AWS cloud at one-fifth the cost of moving over internet
SDK, CLI and more	Transfer data to S3	Software driven infrastructure - easy to integrate with variety of tools

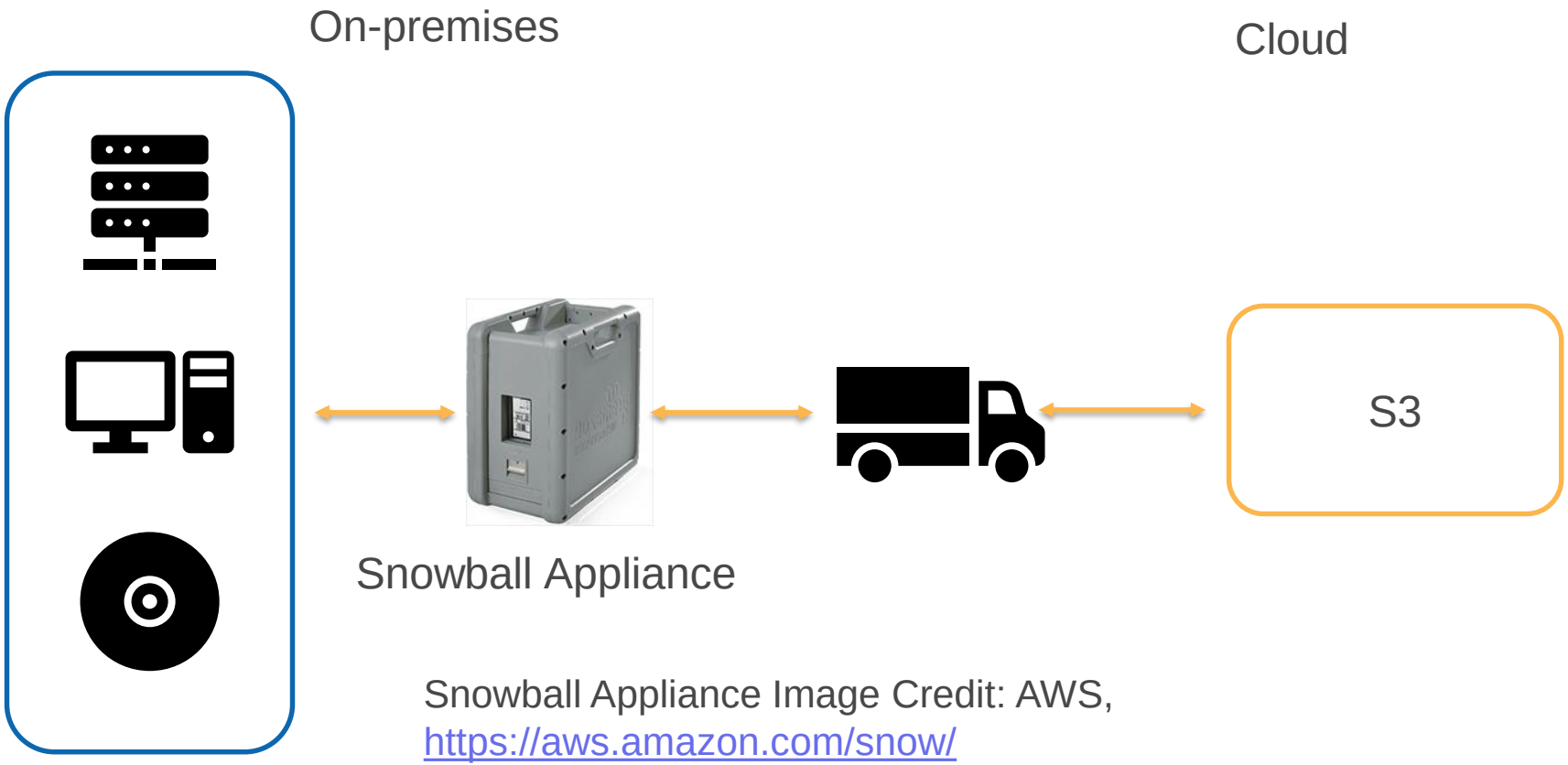
Realtime Streaming Data



Storage Gateway



Snowball

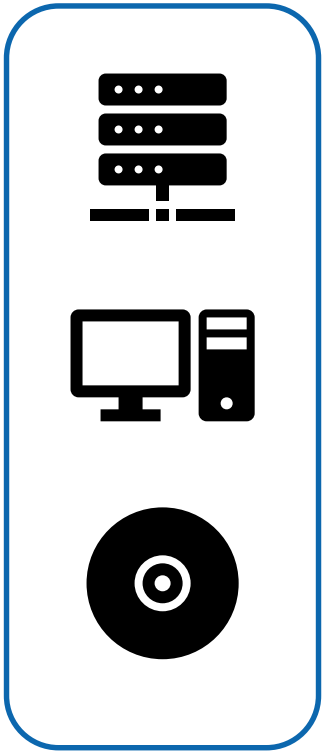


Snowball Appliance Image Credit: AWS, https://aws.amazon.com/snow/

Snowmobile

On-premises

Cloud



S3

Snowmobile Container

Snowmobile Image Credit: AWS, https://aws.amazon.com/snow/

Data Catalog

Service	Purpose	Use
Do-it-yourself	Comprehensive Data Catalog	Make data discoverable and usable. Use services like S3, Lambda, Elasticsearch, DynamoDB to maintain metadata
Glue	Data Catalog (Metadata repository)	Make data discoverable and usable. Automatically crawl and collect metadata from S3, DynamoDB and any other database that supports JDBC connectivity

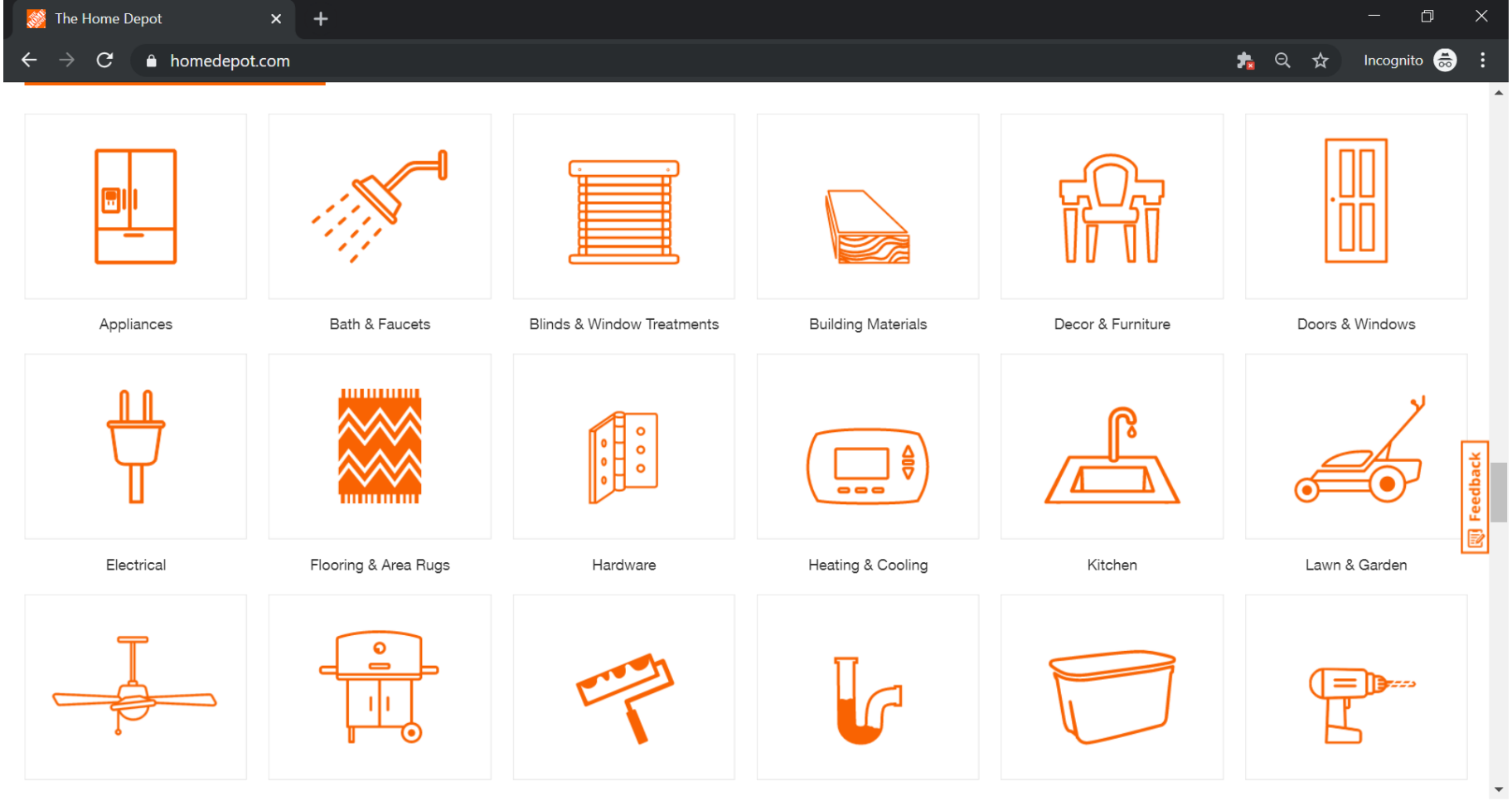


Image Credit: HomeDepot

Image Credit: webhamster, flickr

Data Swamp

“A **data swamp** is a deteriorated and unmanaged data lake that is either inaccessible to its intended users or is providing little value”

Reference: Data Swamp
https://en.wikipedia.org/wiki/Data_lake

Amazon Kinesis

Collect, Process, Analyze Data Streams

Amazon Kinesis

“Amazon Kinesis enables you to ingest, buffer and process streaming data in real-time…..you can derive insights in seconds or minutes.”

“Handle any amount of streaming data from hundreds of thousands of sources with very low latencies”

Reference: Amazon Kinesis, https://aws.amazon.com/kinesis/

Stream Vs. Batch Processing

What is stream processing?

How does it differ from batch processing?

Streaming Data

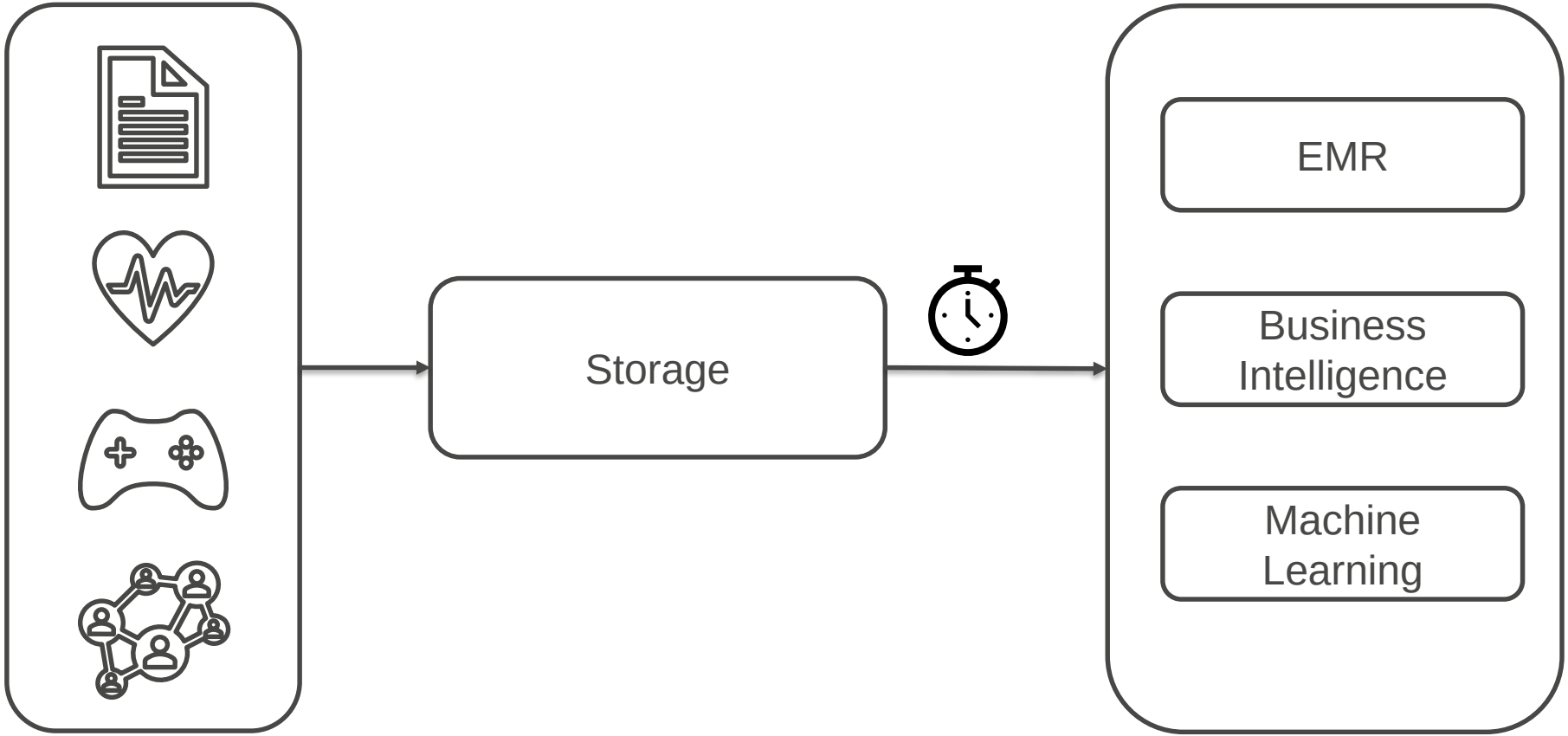
Thousands of sources

Generated Continuously

Small Payloads




Batch Processing

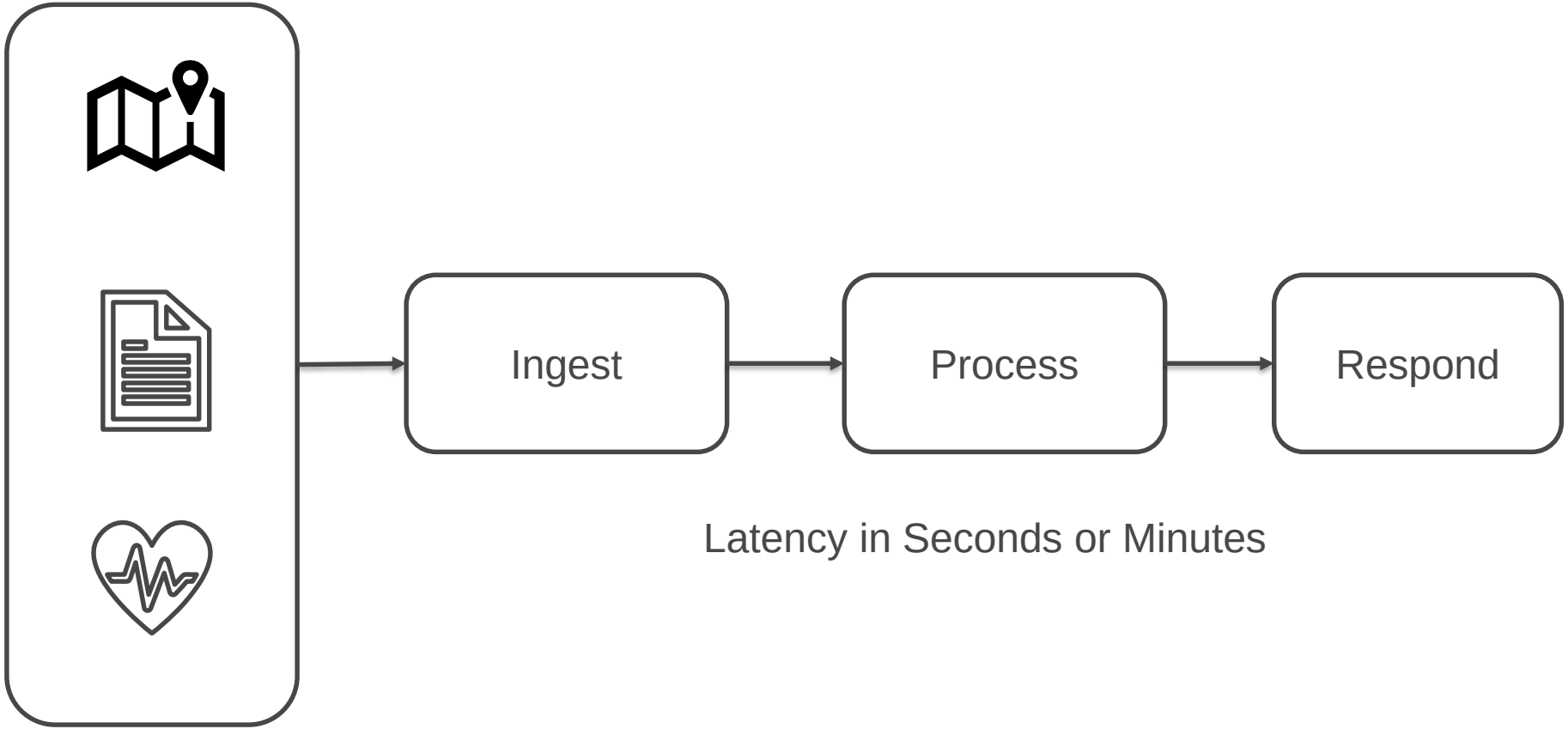


Batch Processing Use Cases

Utility bill generation

Daily, Monthly Manufacturing Reports

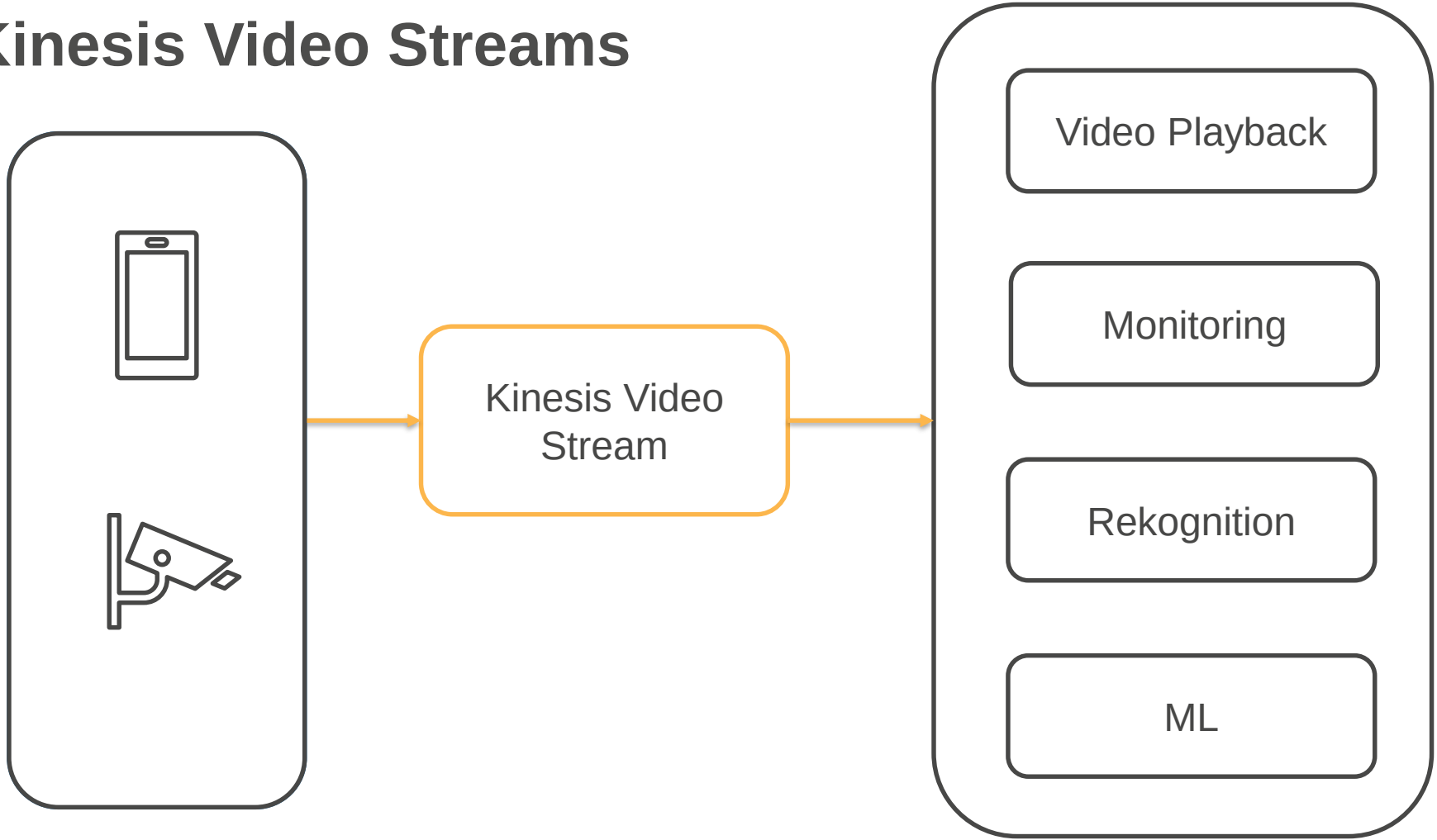
Stream Processing



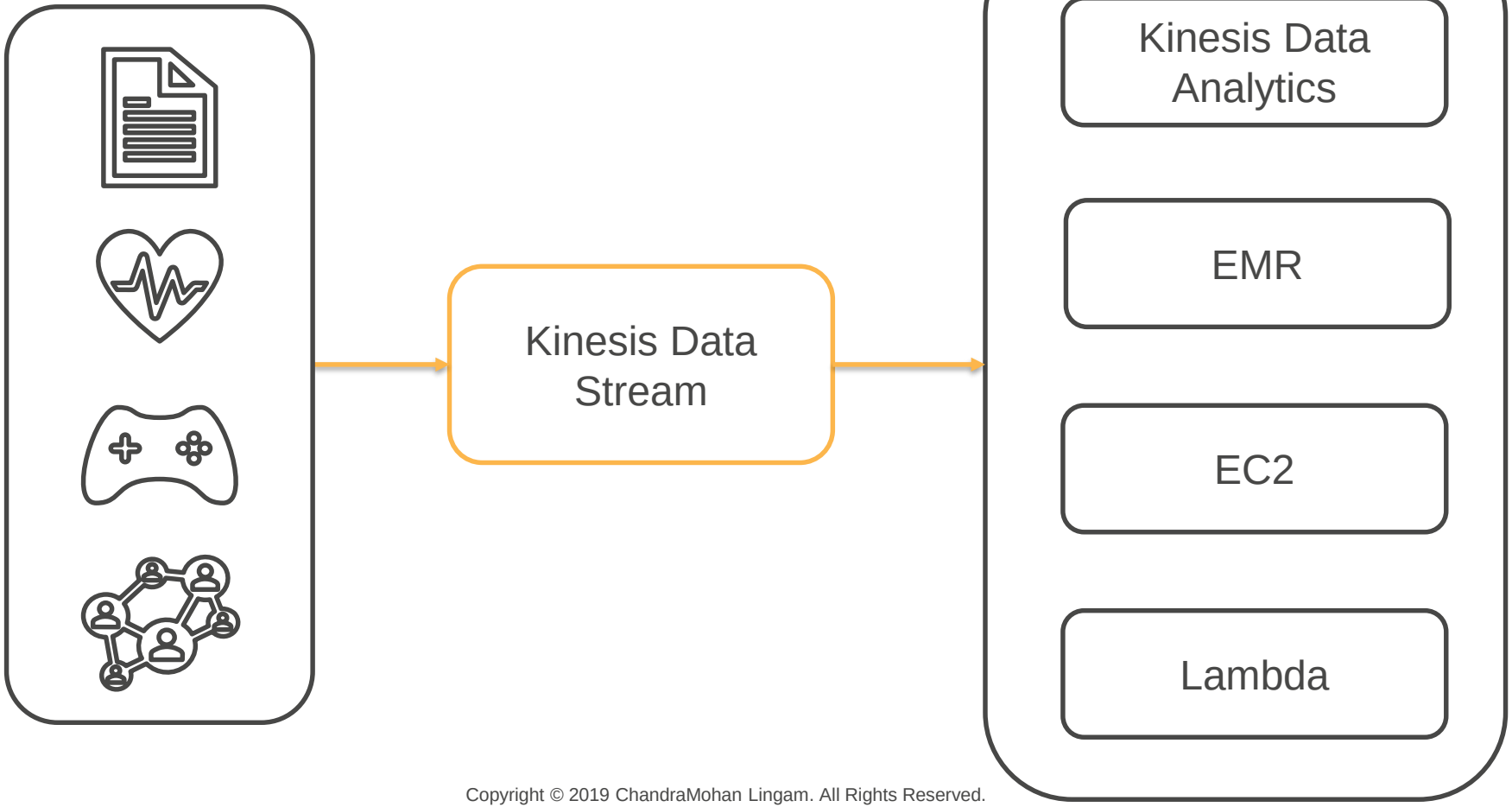
Kinesis

Service	Purpose	Use
Video Streams	Capture and Analyze Video Stream	Security Monitoring, Video Playback, Face detection
Data Streams	Capture and Analyze Data Stream	Custom real-time application
Firehose	Capture and deliver data streams to AWS Data Stores	Use Existing BI tools for Streaming Data: S3, Redshift, ElasticSearch, Splunk.
Data Analytics	Analyze data streams with SQL and Java	Real-time analytics, Anomaly detection

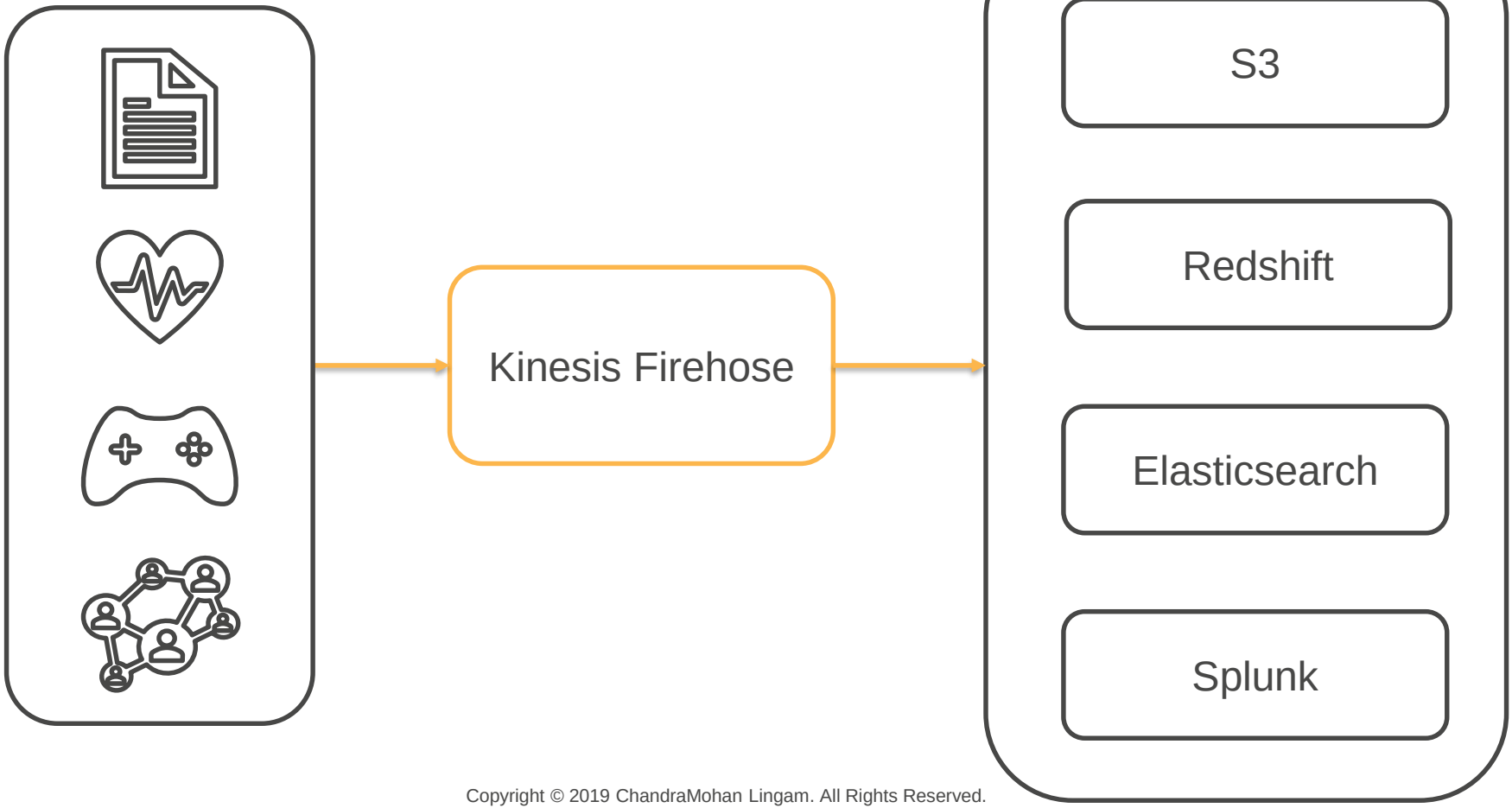
Kinesis Video Streams



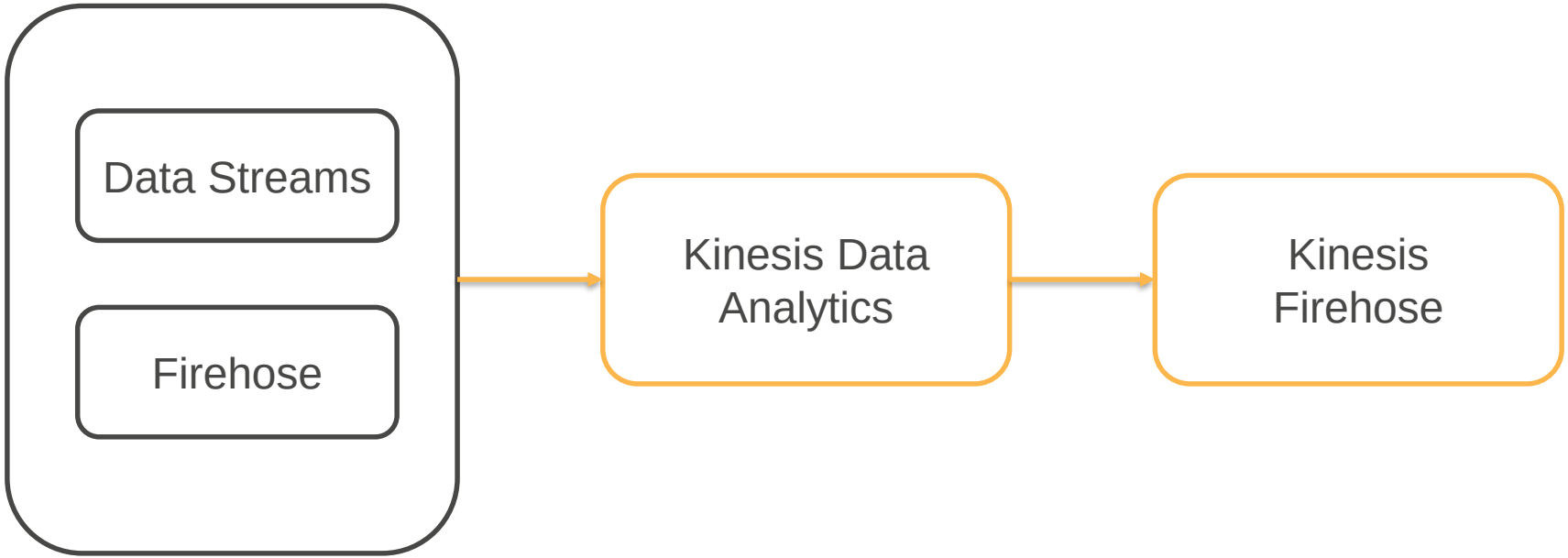
Kinesis Data Streams



Kinesis Data Firehose



Kinesis Data Analytics



Data Formats

Popular Formats, Tools for Conversion

Data Formats

Variety of Formats

Optimal Format can -

- Lower Storage Cost
- Improve Query Performance

Question: When and Where to do the format conversion?

Data Formats

“One of the core values of a data lake is that it is the collection point and repository for all of an organization’s data assets, in whatever their native formats are”

Reference: Data Lake on AWS, https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/building-data-lake-aws.html

Data Formats

Collect Data in Native Format

Transform Data in Data Lake

Data Organization:

- Row Store – Optimized for reading entire row
- Column Store – Optimized for reading subset of columns

Data Formats (Text)

Format	Organization	Use
CSV, TSV	Row	Easy to use No data type support Duplication when used for hierarchical data: For example, in an employee-department CSV file, department information is duplicated for every employee Not optimized for reading only specific columns
JSON	Row	Format of choice for communication between web services Supports data types Efficiently represent hierarchical data
JSON Lines	Row	New Line Delimited JSON Convenient for processing one record at a time

Data Formats (Binary)

Format	Organization	Use
Parquet	Columnar	Ideal for use cases that require only subset of columns Efficiently query large amount of data Write Once Read Many (WORM) Compressed Storage Extensive Tool Support Data Type Support Reduce storage footprint, improve query performance and lower query cost

https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/monitoring-optimizing-data-lake-environment.html

Data Formats (Binary)

Format	Organization	Use
ORC	Columnar	Like Parquet
Avro	Row	Ideal for write-heavy use cases Ideal for scenarios where you need to read the entire record Data Type Support

Data Transformation

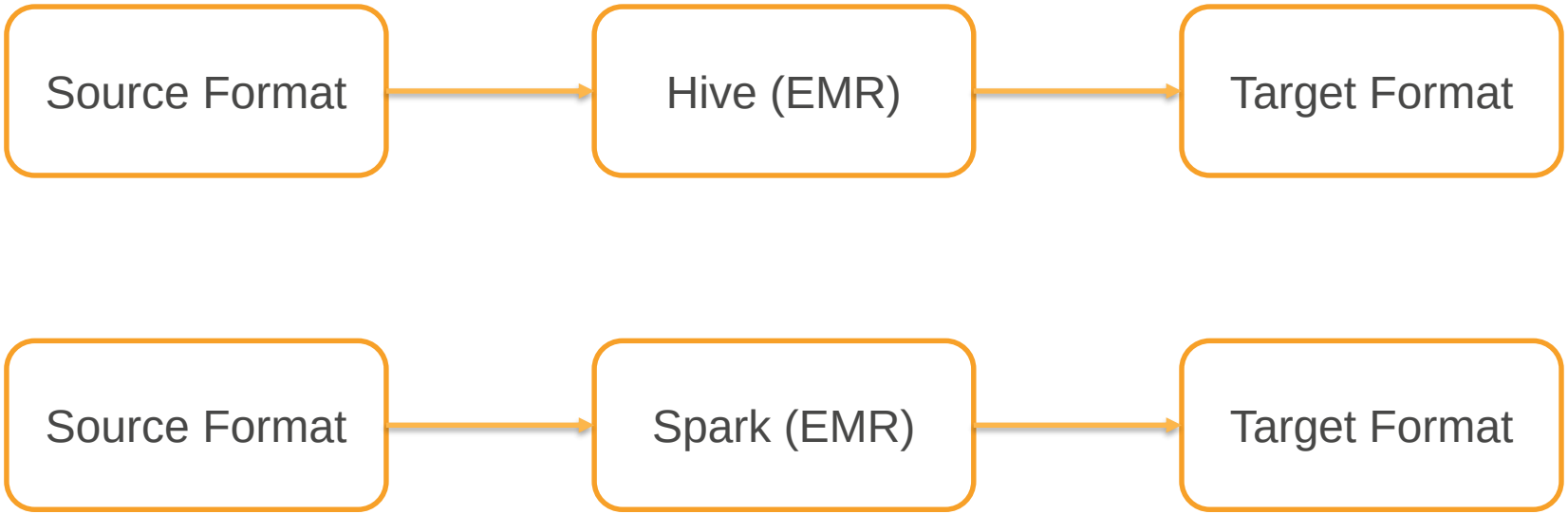
When and Where to do the format conversion?

- Collect in Native Format
- Transform in Data Lake

Data Transformation

Service	Purpose	Use
Amazon EMR	Big Data Preparation and Processing	Managed Hadoop environment Support for tools like Spark, Hive, HBase Support for ML tools like TensorFlow and MXNet List of tools: https://aws.amazon.com/emr/features/

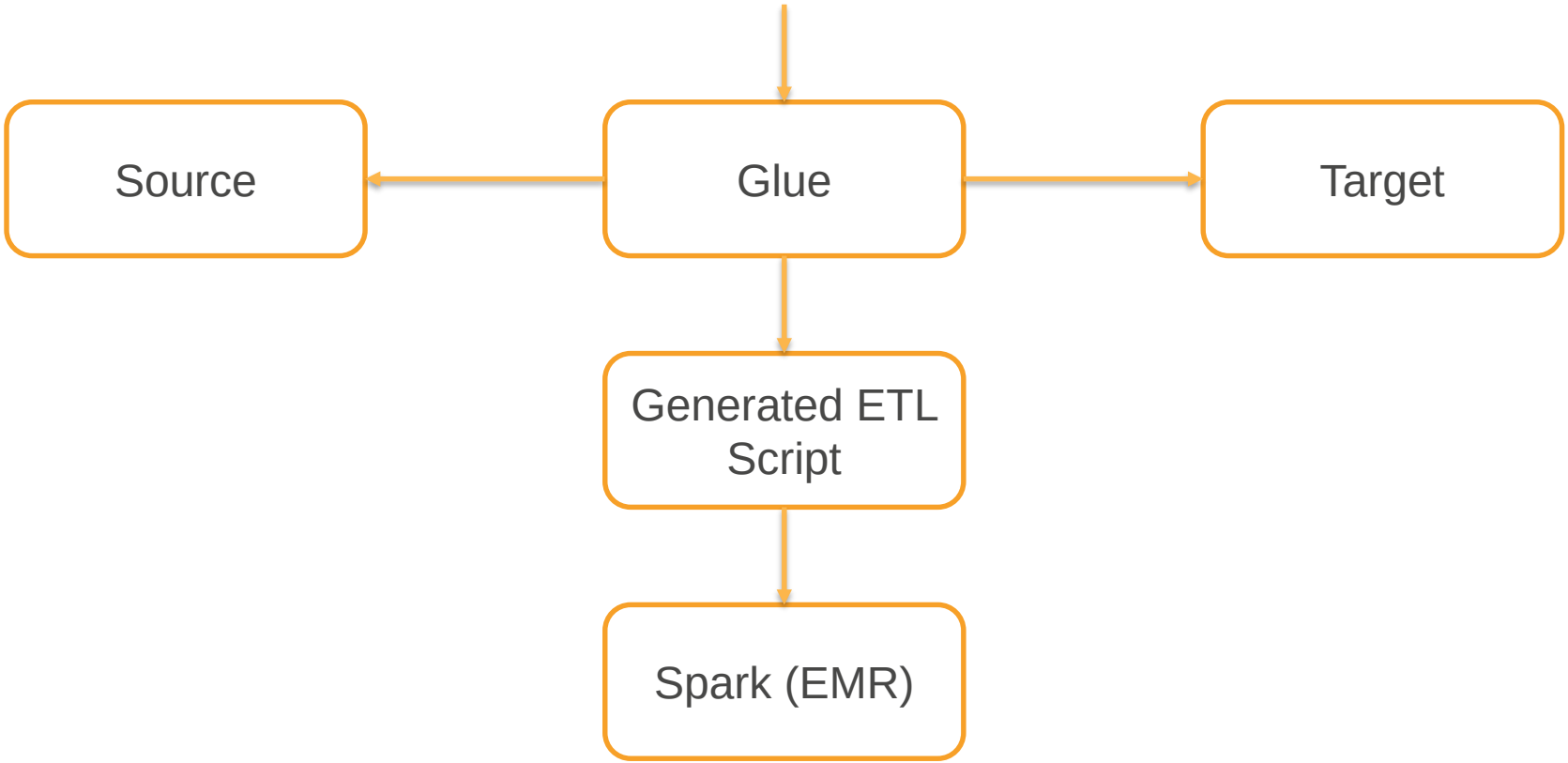
Amazon EMR – Format Conversion



Data Transformation

Service	Purpose	Use
Glue	ETL	Automatically Generate ETL Scripts Schedule and Run on Managed Spark Environment

Glue ETL – Generate and Run Script



Data Transformation

Service	Purpose	Use
Kinesis Firehose	Streaming Data Transformation	Transform streaming data to Parquet, ORC Deliver transformed data to AWS Data Stores Backup original data to S3

In-Place Querying

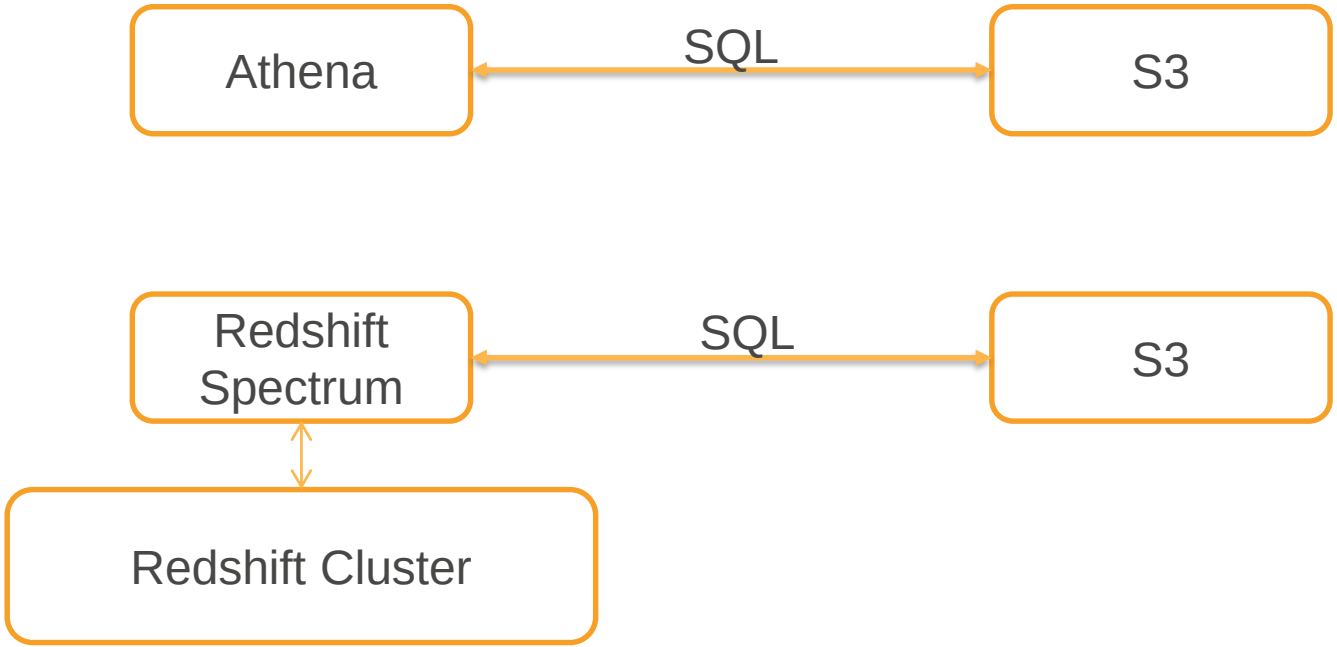
Directly query data in S3 using SQL

In-place Query

“This makes vast amount of unstructured data accessible to any data lake user who can use SQL.”

Reference: Data Lake on AWS, https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/building-data-lake-aws.html

In-Place Query



In-Place Query

Service	Purpose	Use
Athena	In-place SQL Query	Query data in S3 without needing to extract, load into a separate service or platform. Charged based on amount of data scanned https://aws.amazon.com/s3/features/
Redshift Spectrum	In-place SQL Query (Redshift Compatible SQL)	Query data in S3 without needing to extract, load into a separate service or platform. More suitable for complex queries and large datasets (up to Exabytes). https://aws.amazon.com/s3/features/

Recommendations

Athena

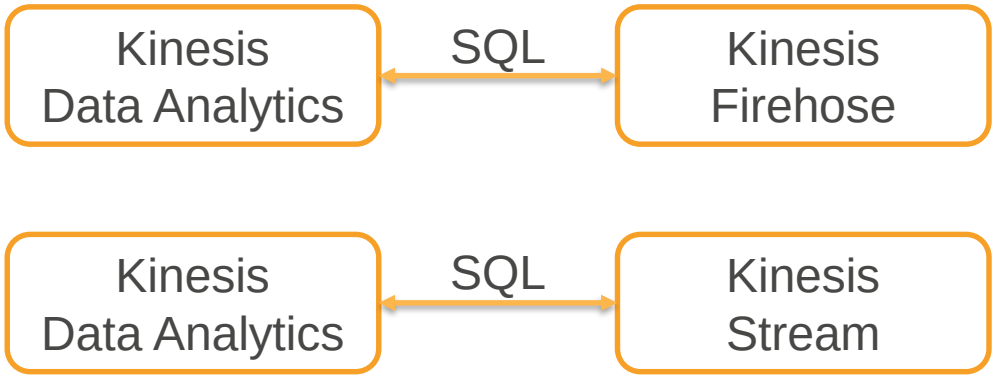
- Ad-hoc data discovery and SQL querying

Redshift Spectrum

- More complex queries
- Large number of users

Streaming Query

Service	Purpose	Use
Kinesis Data Analytics	Streaming Data SQL Querying	Query and analyze Streaming data with SQL https://aws.amazon.com/kinesis/data-analytics/



Broader Analytics Portfolio

Service	Purpose	Use
Amazon EMR	Hadoop Ecosystem tools	You can run variety of workloads using Hadoop tools: Spark, Hive, Pig, Hbase, TensorFlow, MxNet and so forth
SageMaker	Machine Learning	Managed Machine Learning service with wide selection of algorithms
Artificial Intelligence	Video, Image, Natural language	Pre-trained, ready-to-use AI service for video analysis, speech and natural language processing

Broader Analytics Portfolio

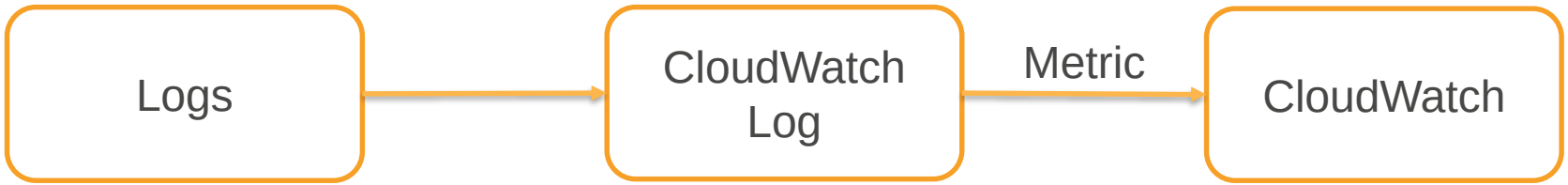
Service	Purpose	Use
Quicksight	Business Intelligence	Managed BI tool to create interactive dashboards
Redshift	Data warehouse (Columnar Storage)	Managed Petabyte scale data warehouse. SQL based querying and easily integrates with your existing Business Intelligence tools
Lambda	Business Logic (Function as a service)	Serverless Backend processing logic with trigger-based code execution

Monitoring and Optimization

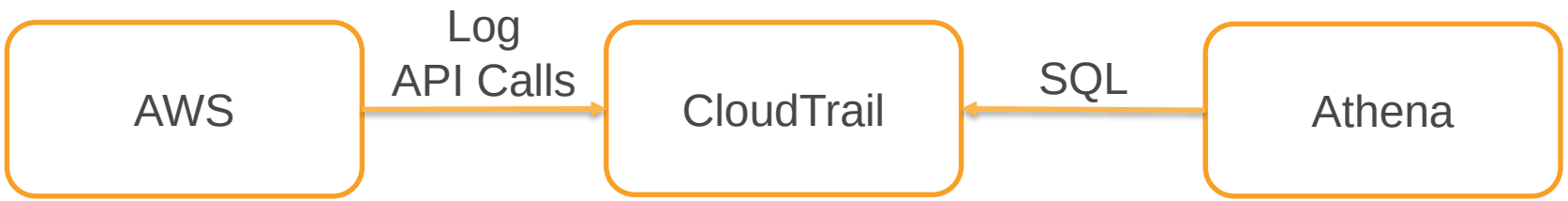
Monitoring

Service	Purpose	Use
CloudWatch	Monitoring	Monitor your resources Configure Alarms to alert Take automated action
CloudWatch Log	Monitor Logs	Monitor log files
CloudTrail	Audit Trail	Logs all activities and who performed those actions Useful for investigation, compliance monitoring

CloudWatch Log – Consolidate Logs and Monitor



CloudTrail – Audit Trail of all API Activities



Optimization

“Data storage is often a significant portion of the costs associated with a data lake.”

Reference: Data Lake on AWS, https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/building-data-lake-aws.html

Cost Optimization

1. S3 Lifecycle Management
2. S3 Storage Class Analysis
3. Intelligent Tiering
4. Amazon Glacier and Glacier Deep Archive
5. Data Formats

Lifecycle Storage Tiering and Expiration

1. Object Age
2. Name and Folder Structure
3. S3 Object Tags

S3 Lifecycle Management

	Standard	Infrequent Access	Glacier
Cost - 500GB per month	USD 11.50	USD 6.25	USD 2.00
Retrieval Fee	-	Per GB	Per GB
Suitable for	Frequently Accessed	Rarely Accessed	Rarely accessed
First byte latency	Immediate	Immediate	Restore can take minutes to hours

Storage Class Analysis

“One of the challenges of developing and configuring lifecycle rules for the data lake is gaining an understanding of how data assets are accessed over time.”

Reference: Data Lake on AWS, https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/building-data-lake-aws.html

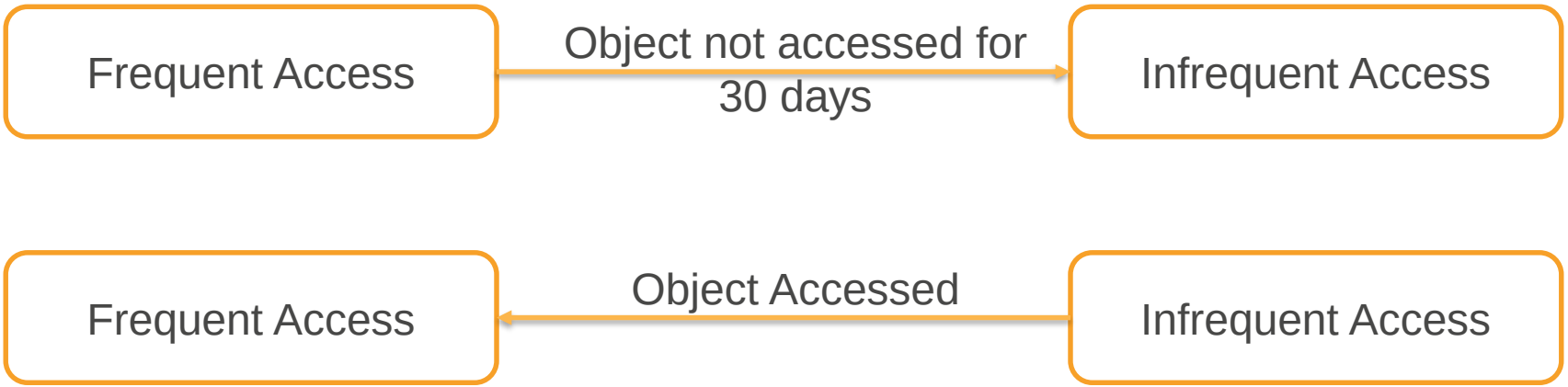
Storage Class Analysis

“This new Amazon S3 analytics feature observes data access patterns to help you determine when to transition less frequently accessed STANDARD storage to the STANDARD_IA storage class”

Reference: S3, https://docs.aws.amazon.com/AmazonS3/latest/dev/analytics-storage-class.html

S3 Intelligent Tiering

Objects are automatically moved between frequent access and infrequent access storage class



Glacier, Glacier Deep Archive

Service	Purpose	Use
Glacier	Archive and Backup	Cost: USD 2.00 for 500 GB/Month Durability: 11 9's Retrieval Time: Minutes to Hours Vault Lock to prevent future edits
Glacier Deep Archive	Archive and Backup	Cost: USD 0.50 for 500 GB/Month Durability: 11 9's Retrieval Time: 12 to 48 hours Vault Lock to prevent future edits

Security and Protection

Data Lake Security

- Data Lake is Centralized
- Consolidates all data in one place

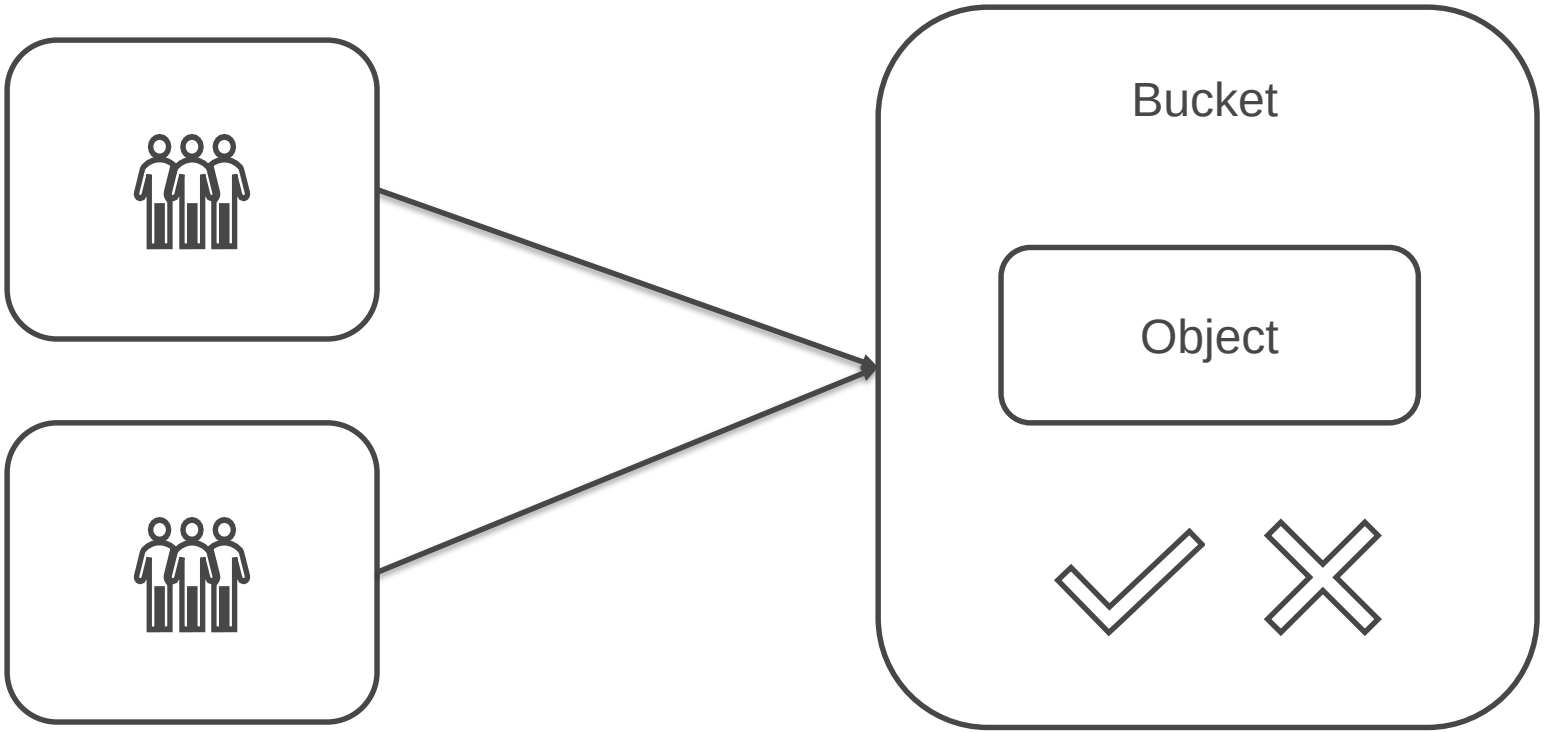
Protecting and Managing Data is very important

S3 Access Control

Resource-based Policy and Access Control

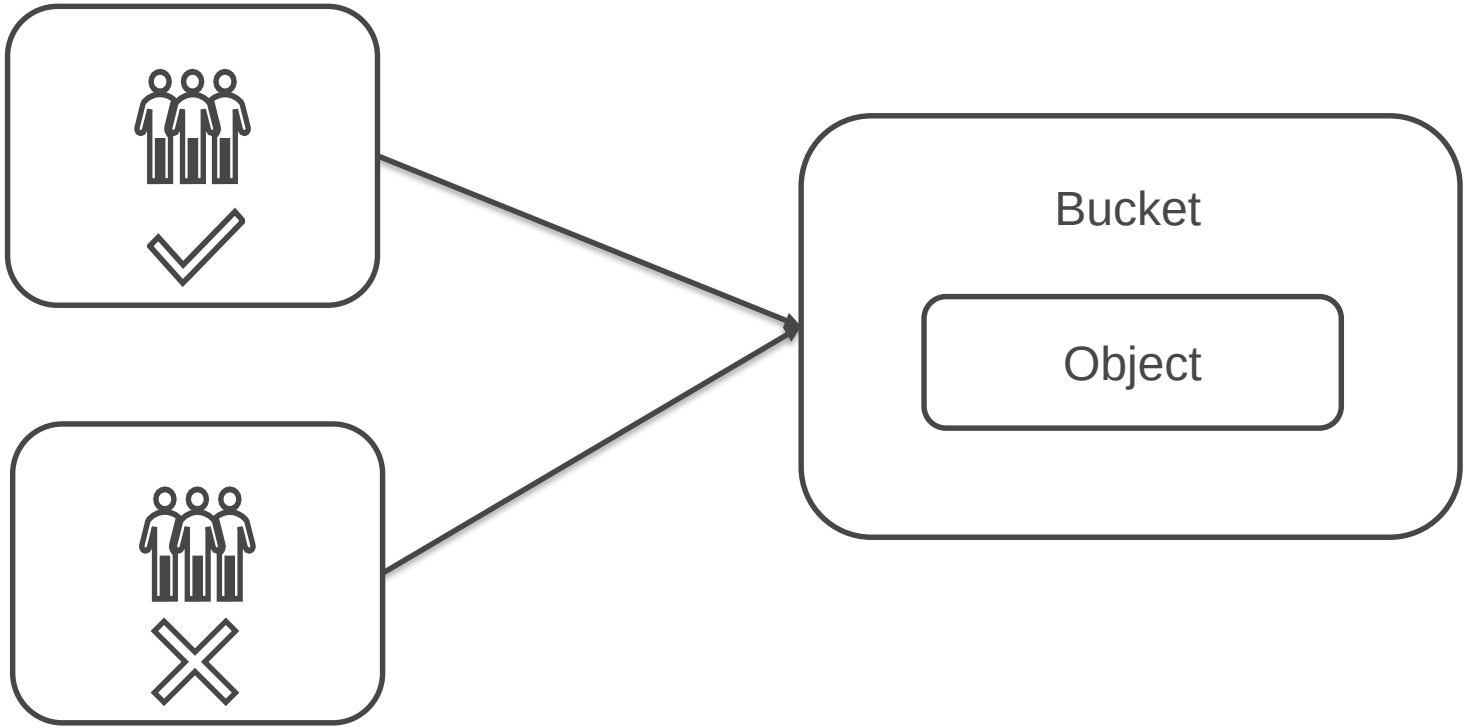
User-based Policy

S3 Resource Based Policy



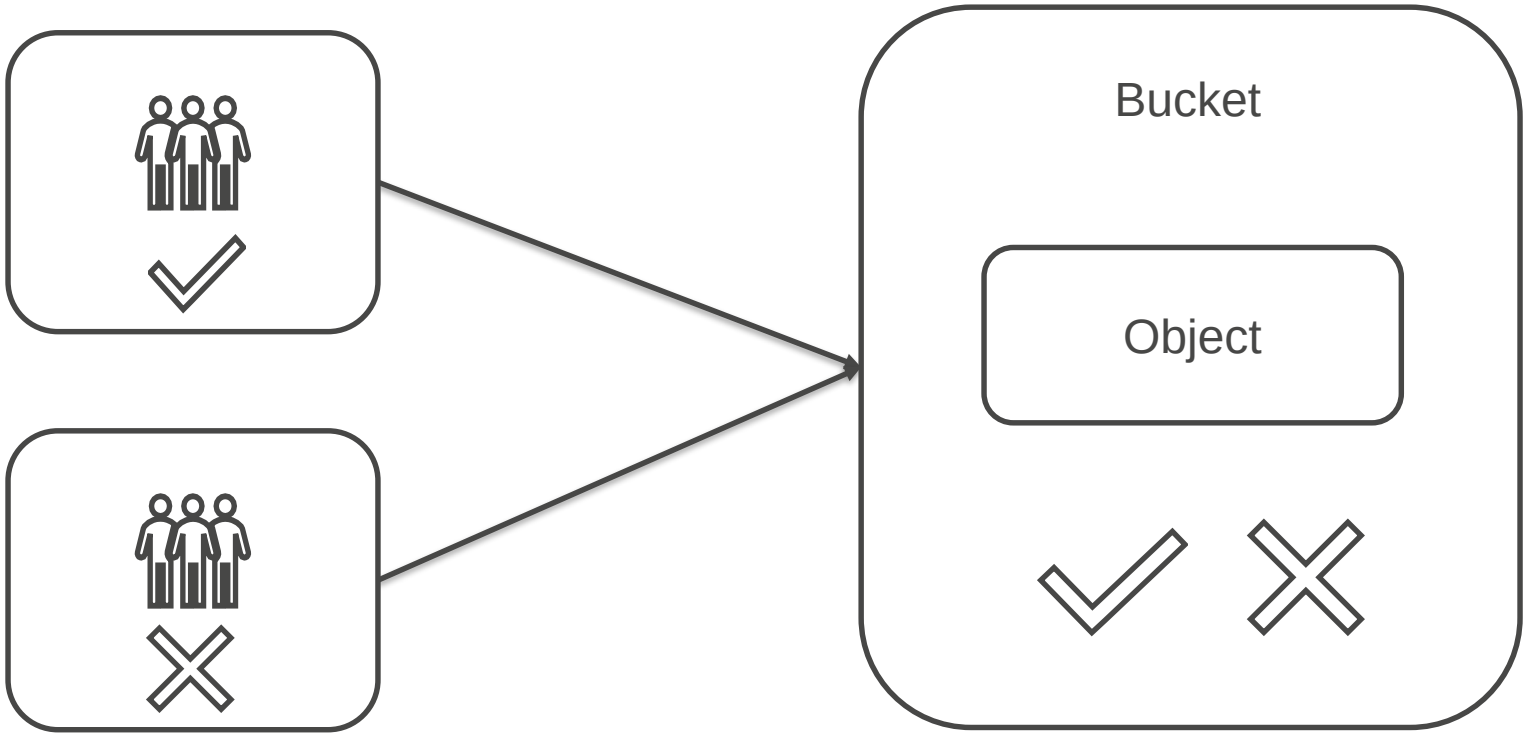
Permissions are embedded as part of Bucket and Object

S3 User Based Policy



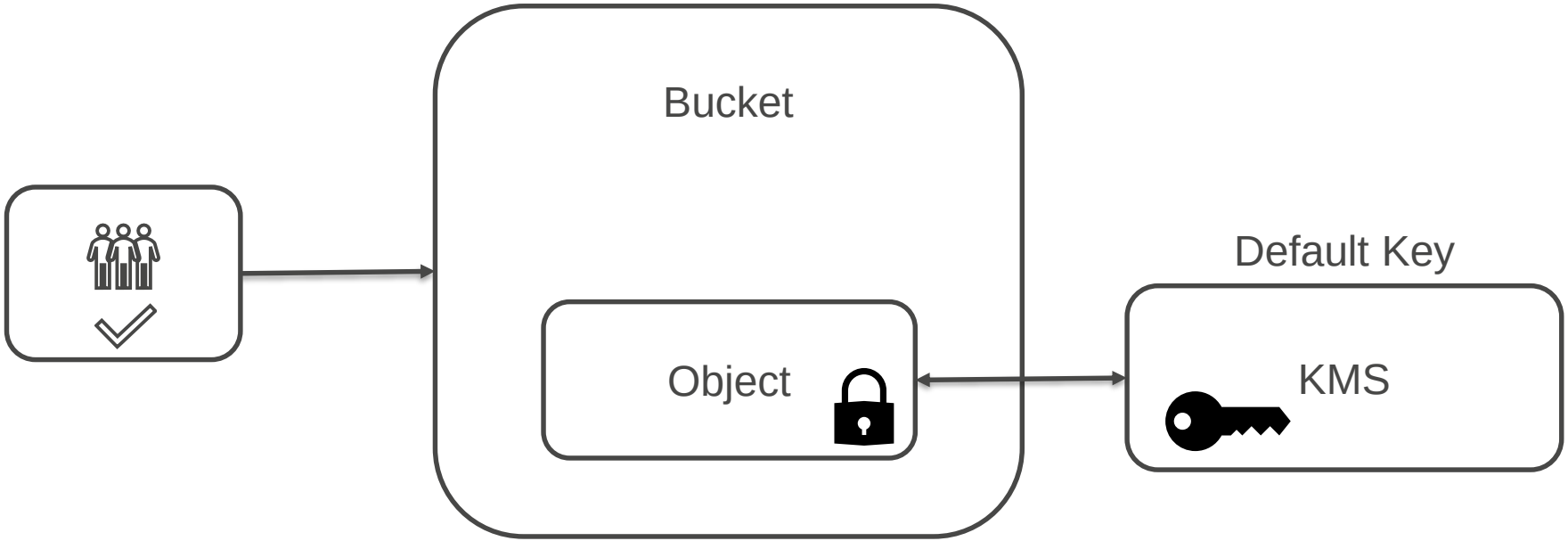
Permissions are granted to Users and Groups

S3 User and Resource Based Policy



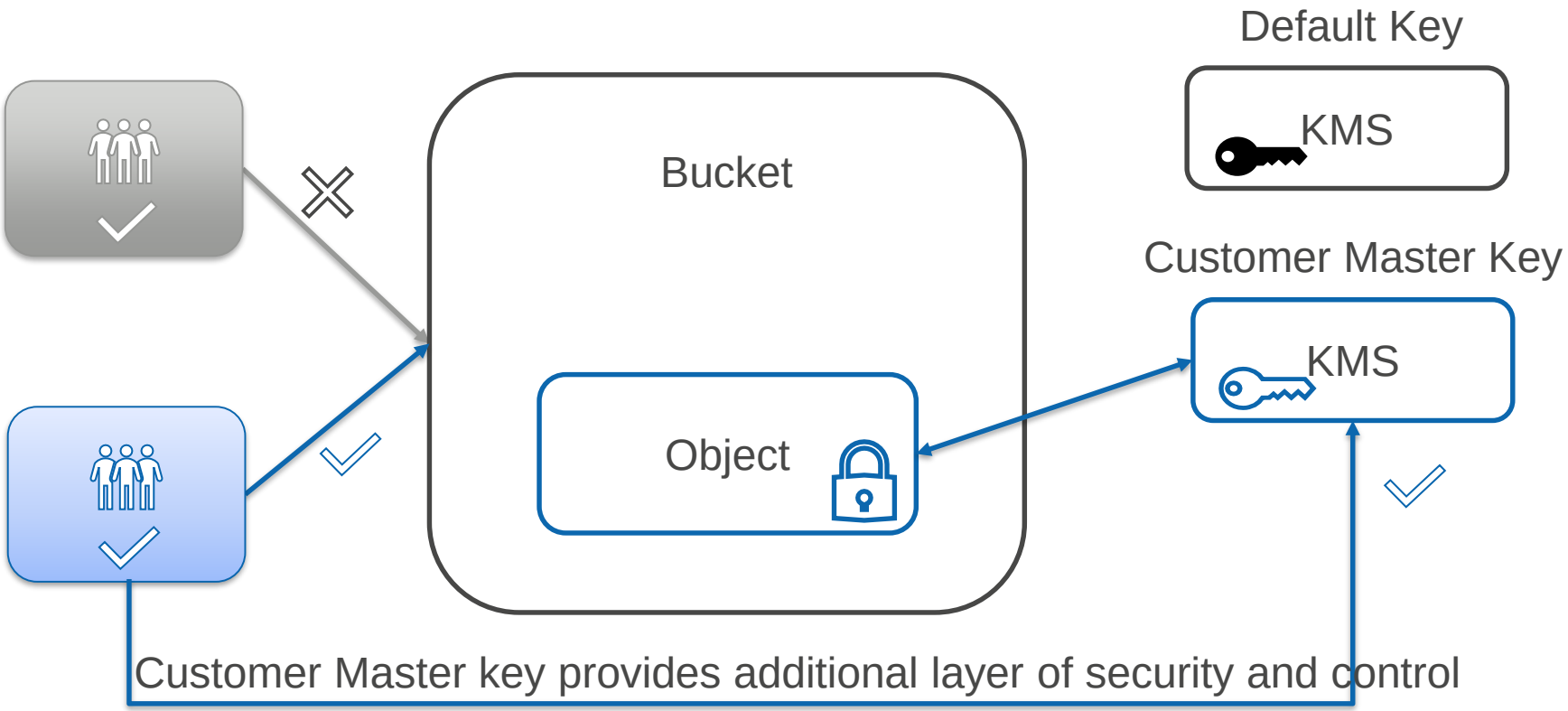
Deny all access that do not originate from on-premises

S3 Data Encryption – Default Key



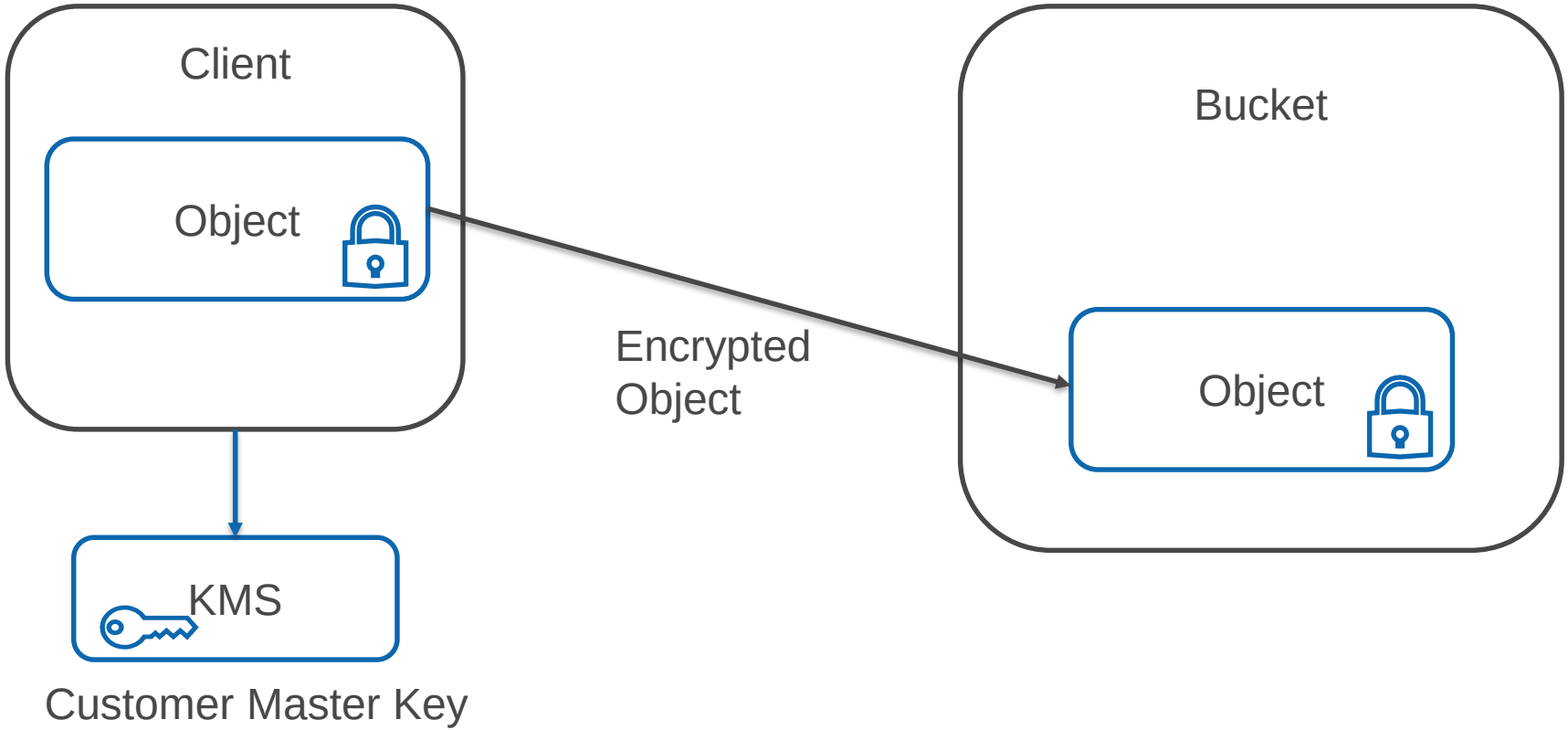
With default key, S3 automatically decrypts object for any user who is allowed access to the bucket or object

S3 Data Encryption – Customer Master Key (CMK)



S3 Client-Side Encryption – Customer Master Key (CMK)

Object encryption and decryption is client responsibility



Protection

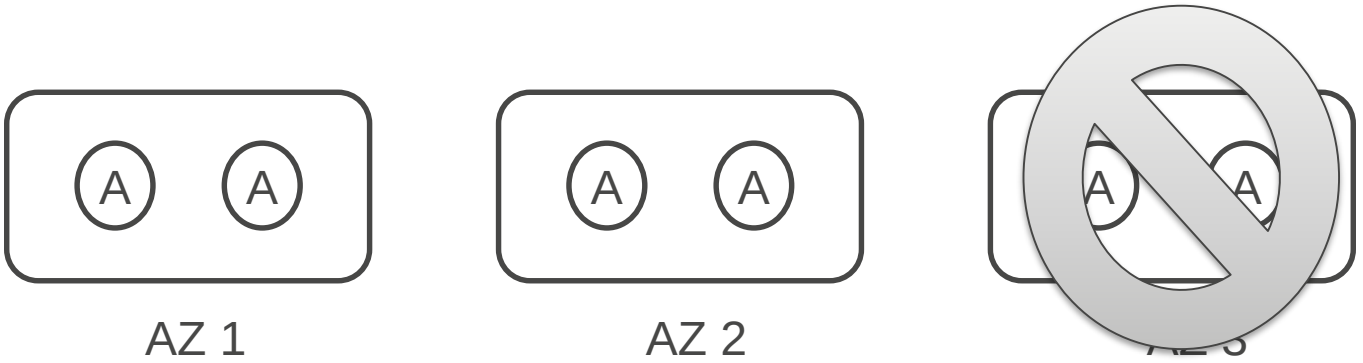
“A data lake must protect data against corruption, loss, accidental or malicious overwrites, modifications, and deletions.”

Reference: Data Lake on AWS, https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/building-data-lake-aws.html

S3 Durability

S3 Durability 99.999999999% (11 9's)

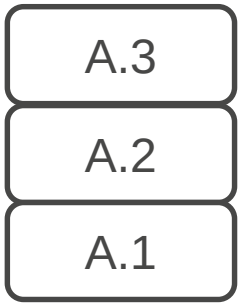
Measure of protection against data loss and corruption



S3 Versioning

Protection against accidental and malicious deletes

S3 maintains versions of objects



Configure Lifecycle Rules for current and previous versions

Multi-Factor Authentication (MFA) for additional layer of authentication

S3 Cross Region Replication (CRR)

Replicate S3 bucket in another region for Disaster Recovery

Automatic and continuous replication

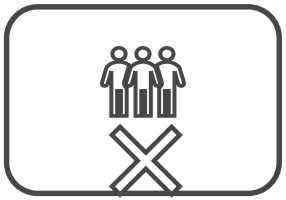
Deletes are not replicated

S3 Object Tagging

Tags are additional meta-data that you can add to Object

Define access control policies based on tags

ALLOW Classification=PHI



DENY Classification=PHI

Object

Classification=PHI

Security and Protection

AWS and S3 provides several features to secure and protect your data

As part of Shared Responsibility Model, Customers are responsible for configuring these security features according to their organization needs

Data Lake Summary

S3 Data Lake Architecture provides a template on how to design and run a data lake for your organization

- Ingest and Store Data
- Discover and Make data usable
- Transform data
- Analyze data in-place
- Future proofing
- Monitor
- Optimize
- Security and Protection

Lab – Glue Data Catalog and Athena

In-place Querying of files stored in S3

- Store file in S3
- Collect metadata with Glue Crawler
- Run Query using Athena

Example Queries (Lab)

- Query first 10 rows

```
SELECT * FROM "demo_db"."iris_csv" limit 10;
```

- Query for a specific class

```
SELECT * FROM "demo_db"."iris_csv"
WHERE class = 'Iris-setosa';
```

- Query by wildcard

```
SELECT * FROM "demo_db"."iris_csv"
where class like '%setosa%';
```

- Get a count

```
SELECT count(*) AS COUNT FROM "demo_db"."iris_csv"
```

- Compute new columns

```
SELECT sepal_length, sepal_width,
       sepal_length * sepal_width as sepal_area
FROM "demo_db"."iris_csv";
```

Lab – Glue ETL

Use Glue ETL to convert files to Parquet format

- Glue automates process of ETL script generation, scheduling and execution
- Glue ETL provisions required Apache Spark infrastructure to run the job

Example Queries - Parquet (Lab)

- Query Iris Parquet Table

```
SELECT sepal_length, sepal_width,
       sepal_length * sepal_width as sepal_area
FROM "demo_db"."iris_parquet" limit 10;
```

Lab – Customer Review

Query Amazon Customer Reviews Public Dataset using Athena

- Create table definition (instead of using Glue Crawler)
- Update catalog with partition
- Query using Athena

Reference:

https://s3.amazonaws.com/amazon-reviews-pds/readme.html

https://registry.opendata.aws/

Example Queries – Customer Review

- Highly Rated Books

```
SELECT product_title, star_rating,review_body
FROM  "demo_db"."amazon_reviews_parquet"
WHERE product_category = 'Books'
and star_rating > 3
limit 10;
```

- Book Reviews for specified book title pattern

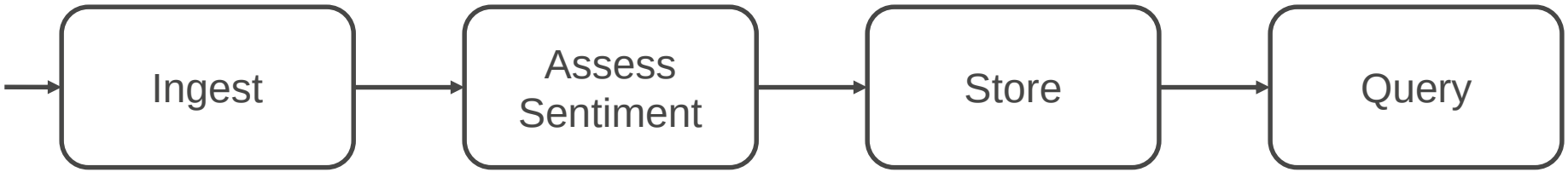
```
SELECT product_title, star_rating,review_body
FROM  "demo_db"."amazon_reviews_parquet"
WHERE product_category = 'Books'
and product_title like 'Harry Potter%'
and star_rating > 3
limit 100;
```

Lab – Sentiment of the Customer Review

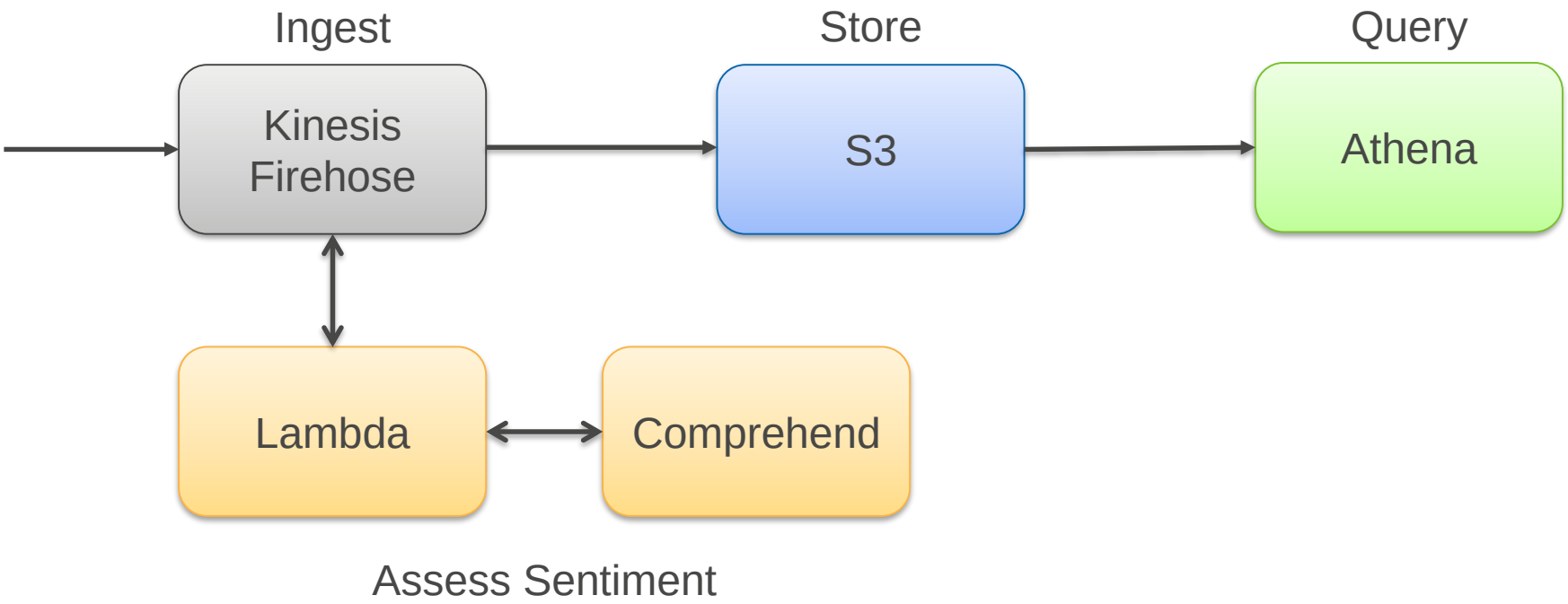
Find Sentiment of the customer review using Comprehend AI Service

With Athena, Query the reviews using sentiment

Lab – Serverless Customer Review Solution



Lab – Serverless Customer Review Solution



Deep Learning

Neural Networks

Chandra Lingam

Cloud Wave LLC

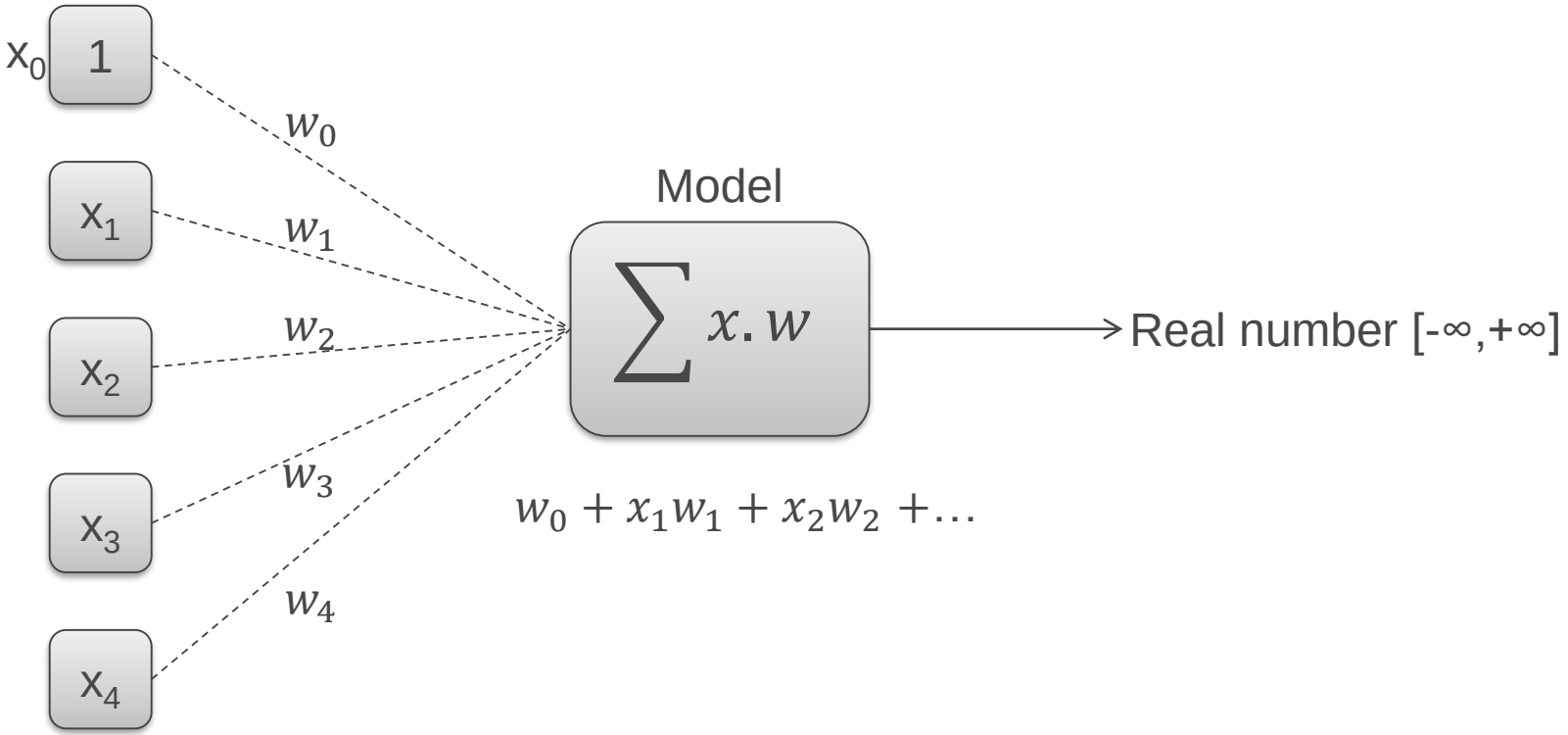
Linear Model

Model Training

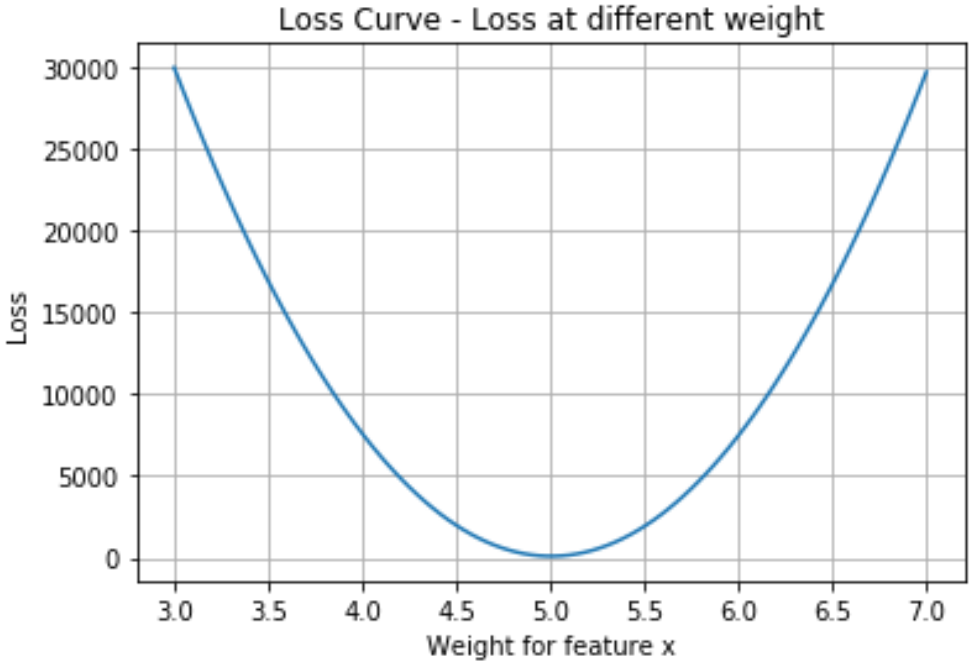
Gradient Descent Optimizer

Loss Function

Linear Regression



Loss Curve



Gradient Descent Variants

- RMSProp
- AdaGrad
- Adam
- …

Additional Reading:

https://ruder.io/optimizing-gradient-descent/

Gradient Descent Modes

Mode	Description
Batch	• Compute loss for all training examples • Adjust weight Example: 150 samples in training set. For each iteration, weight is adjusted once
Stochastic	• Compute loss for next example • Adjust weight Example: 150 samples in training set. For each iteration, weight is adjusted 150 times
Mini-batch	• Compute loss for specified number of examples • Adjust weight Example: 150 samples in training set. Mini-batch size is 15. For each iteration, weight is adjusted 10 times

Loss Plot with Multiple Features

Image Courtesy: Anantha Metals,
https://www.ananthaonline.com/

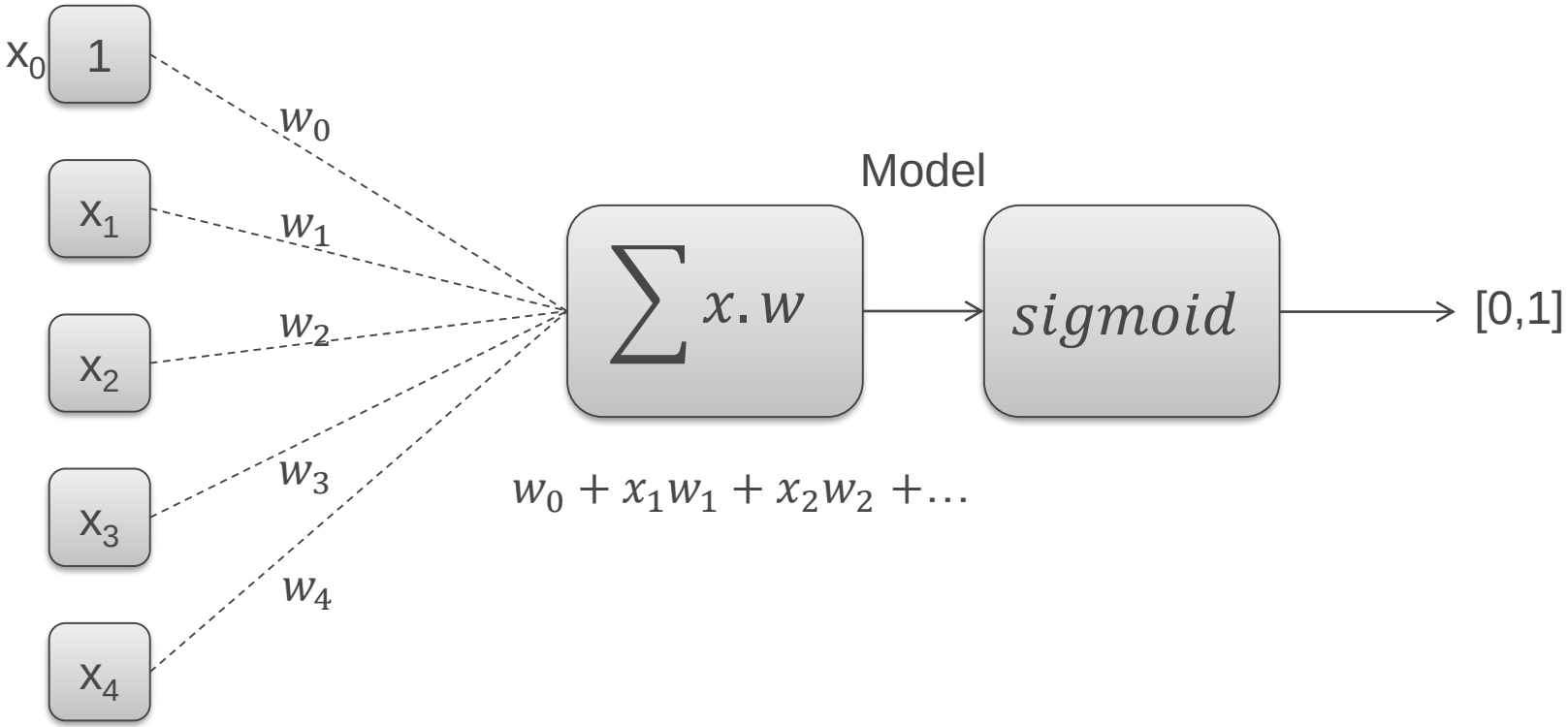
Linear Model - Summary

Loss Function

Learning Rate

Gradient Descent Optimizer

Logistic Regression (Binary Classification)

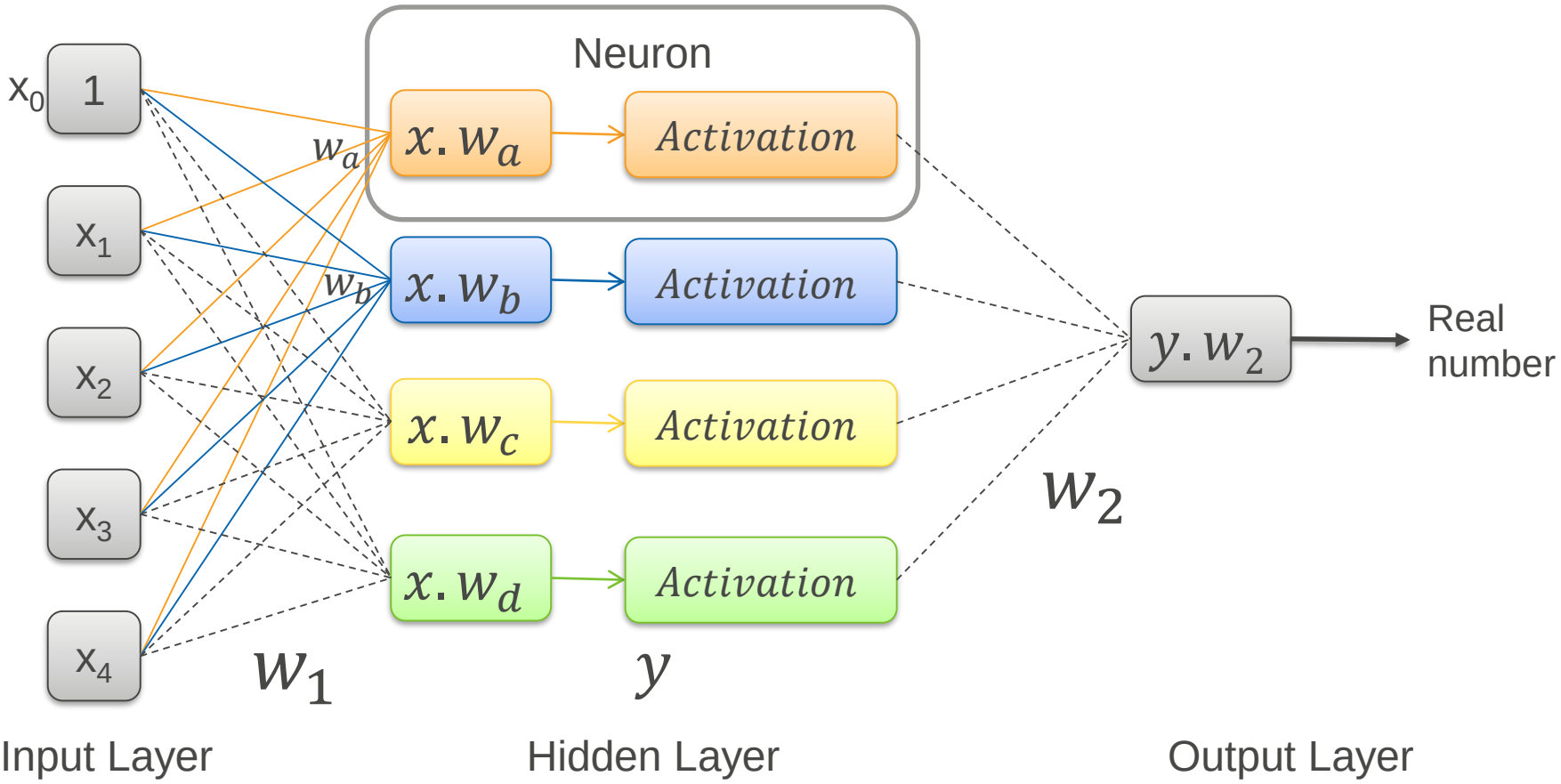


Summary - Linear Models

- Features and Weights
- Loss Function, Gradient Descent
- Simple to understand and use
- Difficult to use with Non-linear datasets
- Requires complex feature engineering
- Data needs to be in similar range and scale – preferably standardized or normalized

****Foundation for deep learning******

Neural Networks – Regression



Neural Networks

- Automatic feature-engineering – mixes features to create new ones
- Handles non-linear datasets
- Easy to overfit (apply regularization, reduce model complexity and so forth)

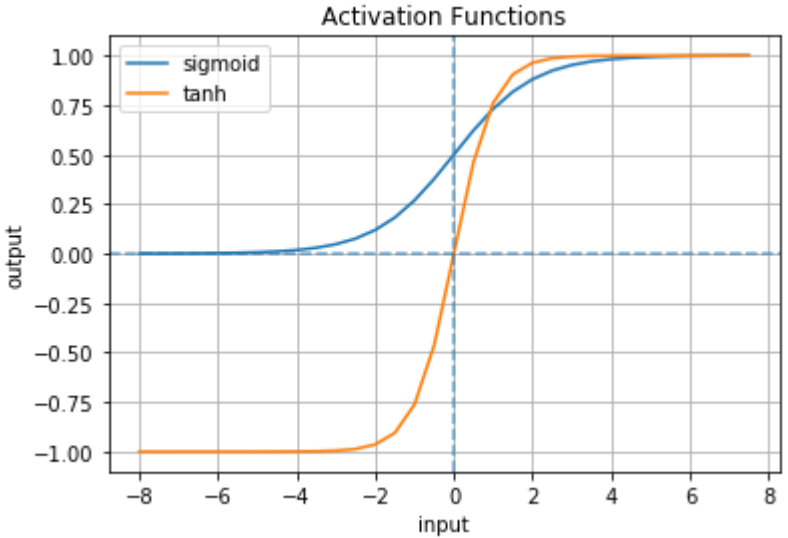
Activation Functions

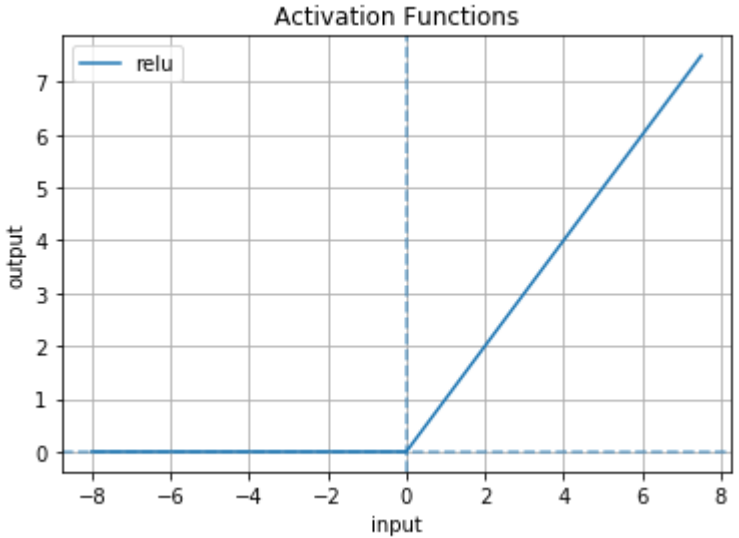
Introduce non-linearity in the model

Improves ability of model to fit complex non-linear datasets

Three popular activation functions: sigmoid, tanh, relu

Activation Functions





Sigmoid: for any input, output is bounded between 0,1

Tanh: for any input, output is bounded between -1,1

ReLU : for any input x, output is max(0,x)

Deep Learning

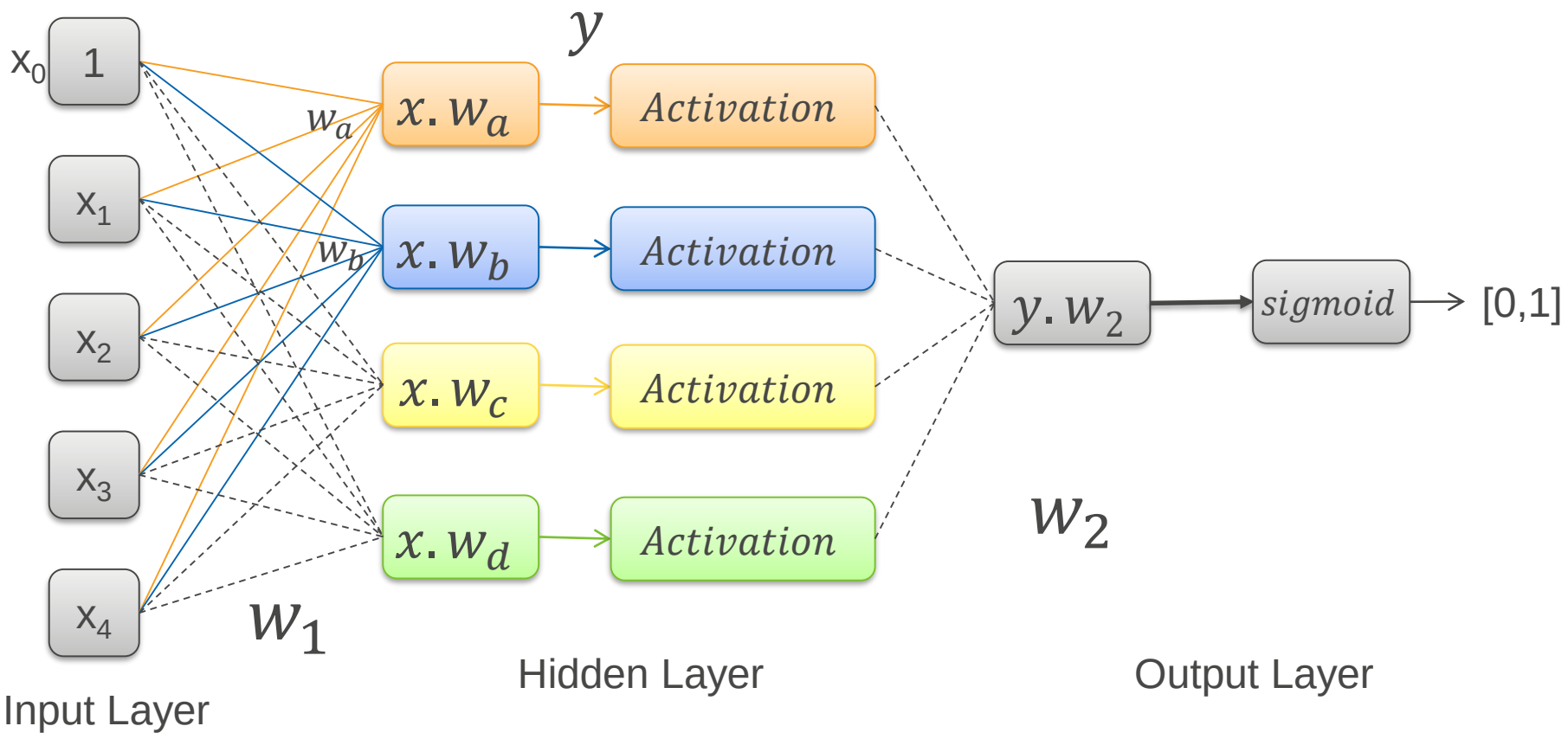
“Traditional ML algorithms appear to saturate on how much they can learn. So, having massive amounts of data does not translate to ‘more learning’”

“Small NN can learn better. Medium NN can learn even more. Large NNs can keep learning with more data (several hidden layers)”

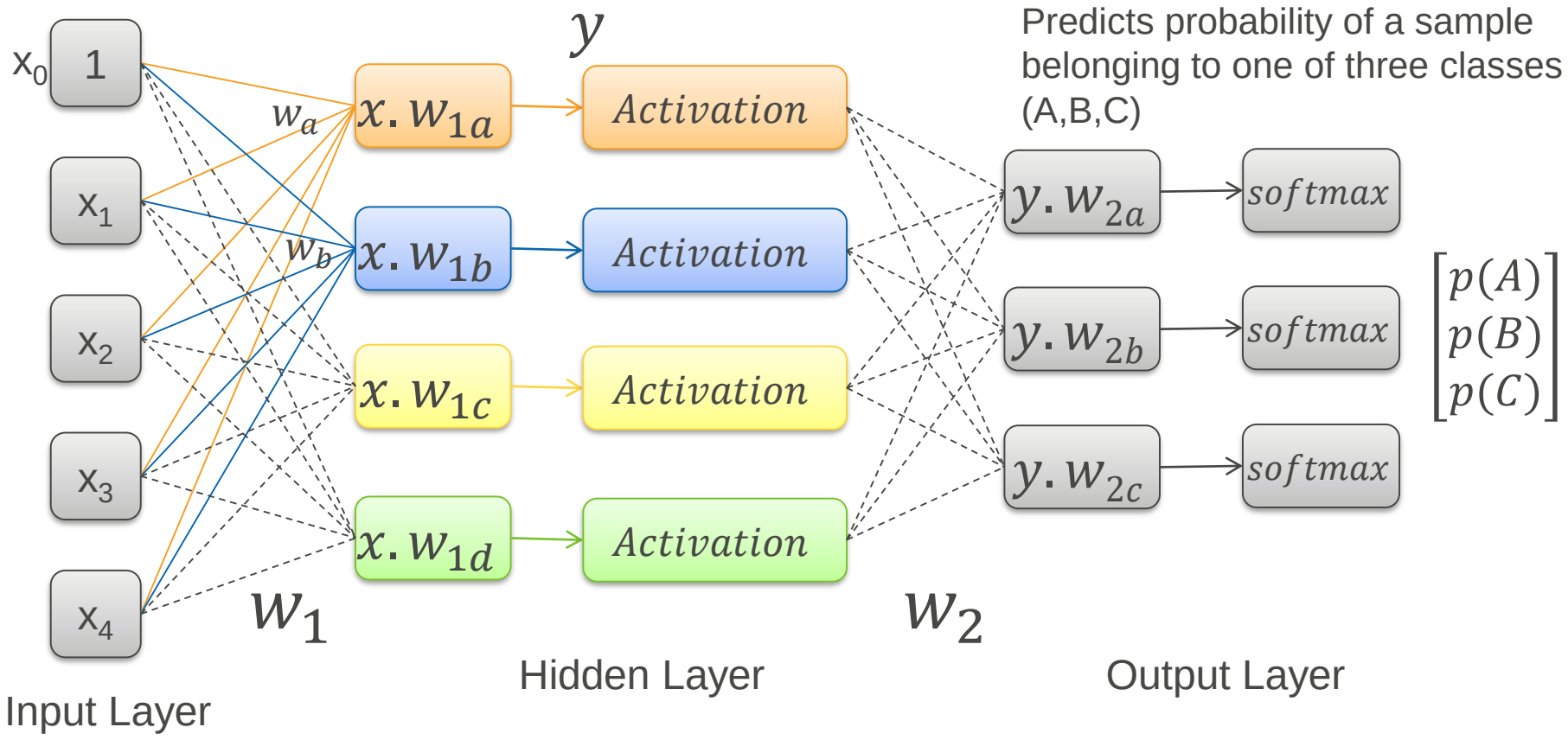
Reference: Dr. Andrew Ng

"Nuts and bolts of building AI applications using Deep Learning“.
https://www.youtube.com/watch?v=wjqaz6m42wU

Neural Networks – Binary Classification



Neural Networks – Multiclass Classification



Popular Neural Network Architectures

General Purpose

- Fully connected network
- Example: treats each pixel as a separate feature

Convolutional Neural Network (CNN or convnet)

- Useful for image analysis
- Example: considers pixel and its surrounding pixels

Popular Neural Network Architectures

Recurrent Neural Network (RNN)

- Looks at history
- Useful for timeseries forecasting, natural language processing
- Example: time series forecasting – model looks at current value and historical values

Lab – Regression with SKLearn Neural Network

- Kaggle Bike Rental Data
- Train using SKLearn's MLPRegressor (Multi-layer Perceptron)
- Data Preparation:
    - One Hot Encode all categorical features
    - Standardize or Normalize all numeric features

Deep Learning Libraries

Keras API

“Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano.

It was developed with a focus on enabling fast experimentation.”

Reference: Keras.io, https://keras.io/

Lab – Regression with TensorFlow Neural Network

- Kaggle Bike Rental Data
- Train using Keras API with TensorFlow as Backend
- SageMaker Notebook comes pre-installed with TensorFlow, Apache MxNet, Pytorch and other DeepLearning Libraries
- To change the backend, use the appropriate Jupyter Notebook kernel in SageMaker

Lab – Binary Classification with Neural Network

- Mobile Operator Customer Churn Prediction
- Predict probability of a customer joining a competitor
- Build a neural network for binary classification with Keras
- Data transformation to handle highly correlated features, categorical and numeric data

Adapted from Predicting Customer Churn with Amazon Machine Learning by Denis V. Batalov, and SageMaker Examples

Lab – Multiclass Classification with Neural Network

- Iris Plant Classification
- Transform target using one hot encoding

Additional Resources - Books

Introduction to Machine Learning with Python

by Andreas C. Müller and Sarah Guido

Solid introduction to scikit-learn and machine learning

Deep Learning with Python

by François Chollet

Great introduction to Deep Learning with Keras

Additional Resources - Videos

MIT 6.S191: Introduction to Deep Learning

by Alexander Amini

Excellent introduction to deep learning, loss function, optimizers

MIT 6.S191: Convolutional Neural Networks

MIT 6.S191: Recurrent Neural Networks

by Ava Soleimany

Excellent overview of CNN and RNNs

Additional Resources - Videos

NIPS 2016 tutorial: "Nuts and bolts of building AI applications using Deep Learning"

by Andrew Ng

Practical tips, tricks and industry experience

Transfer Learning

by Andrew Ng

Additional Resources - Articles

The importance of hyperparameter tuning for scaling deep learning training to multiple GPUs

by Sina Afrooze

Guide To Multi-Class Multi-Label Classification With Neural Networks In Python

by Tobias Sterbak

Bring Your Own Algorithms

AWS SageMaker

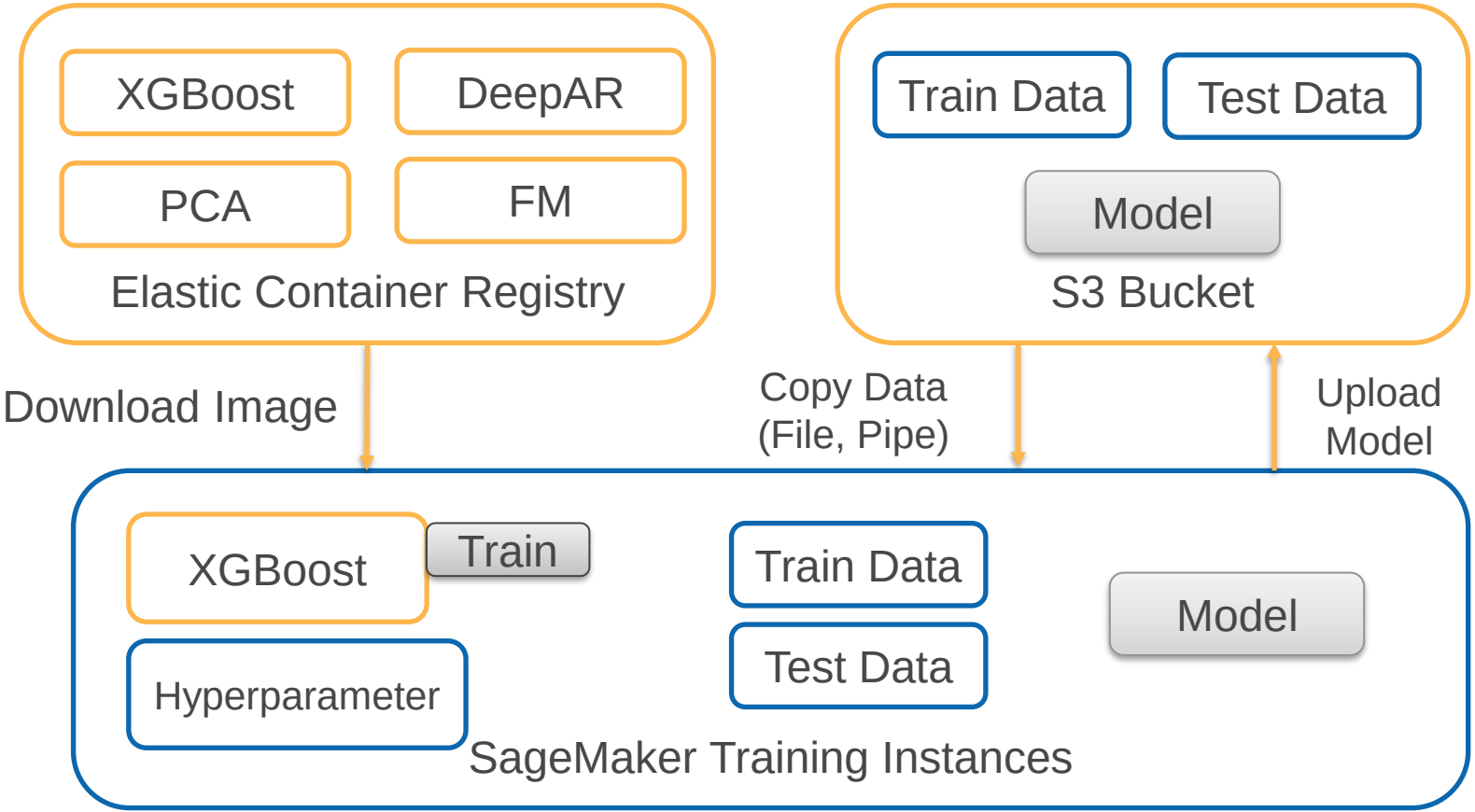
Chandra Lingam

Cloud Wave LLC

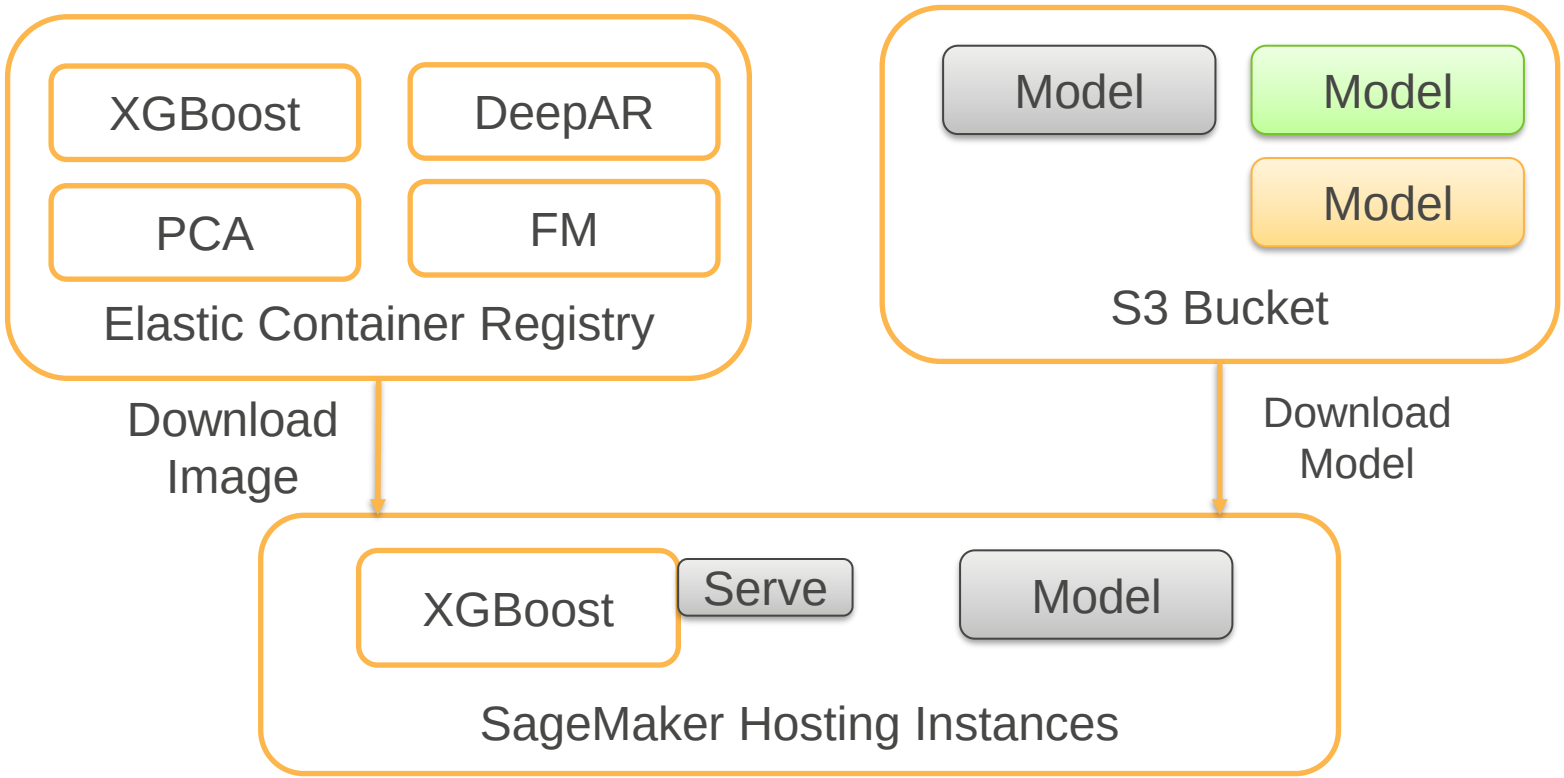
SageMaker – Training and Hosting

Options	Usage Scenario
Built-in Algorithms	Training algorithms provided by SageMaker Easy to use and scale Optimized for AWS Cloud
Pre-built Container Images	Supports popular frameworks like MxNet, TensorFlow, scikit-learn, PyTorch Flexibility to use wide selection of algorithms
Extend Pre-built Container Images	Extend pre-built container images to your needs
Custom Container Images	Use different language and framework

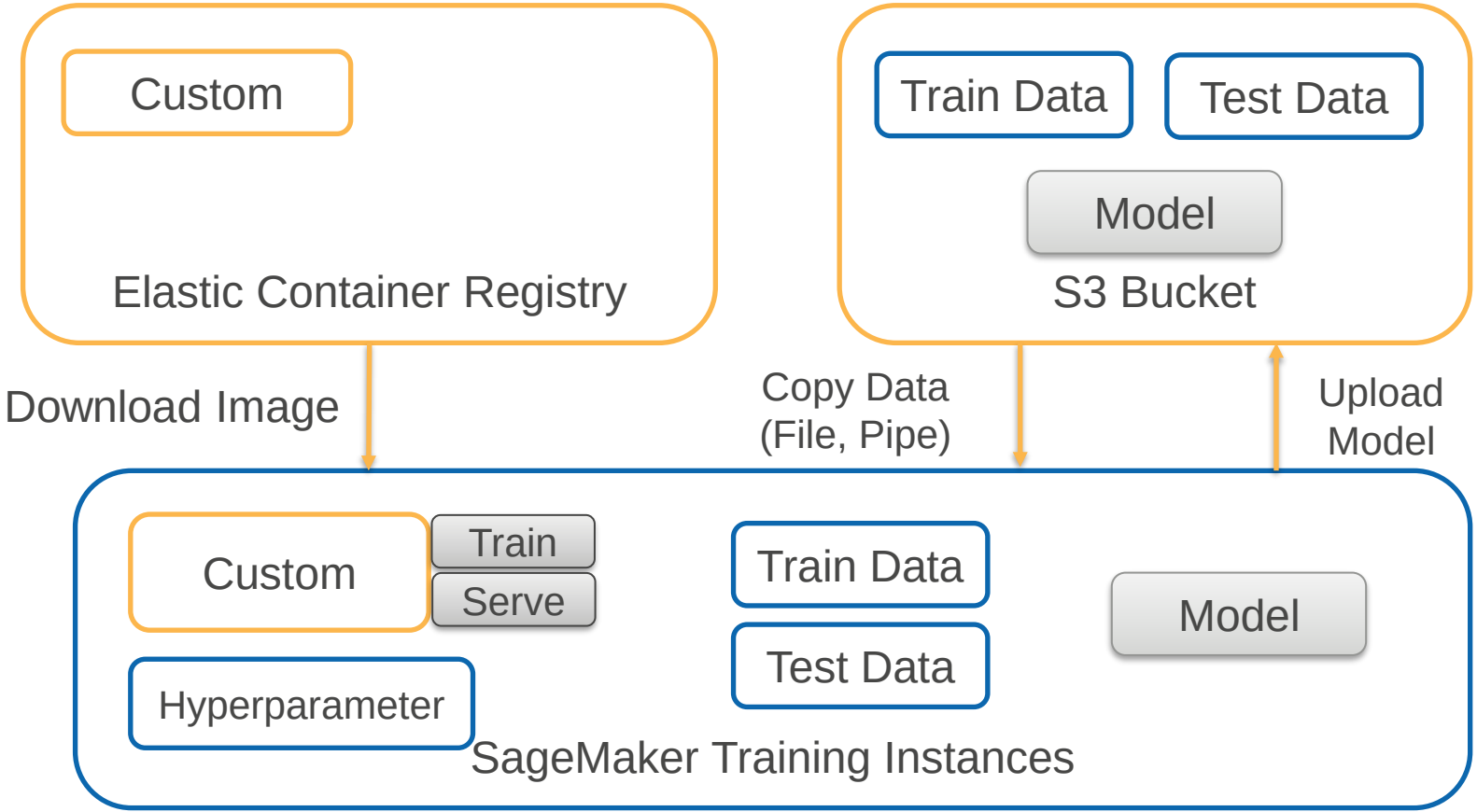
Built-in Algorithms - Training



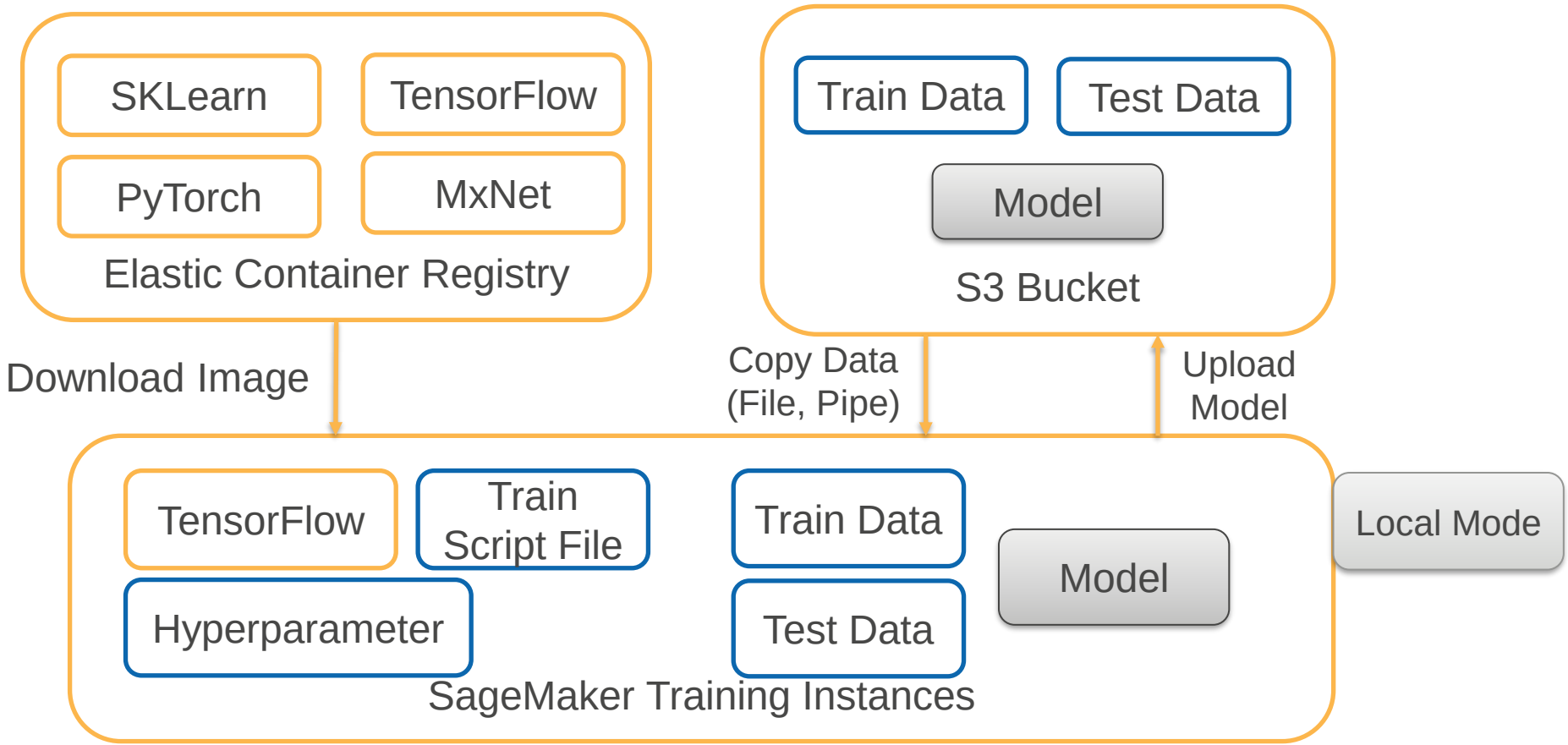
Built-in Algorithms – Hosting (Realtime, Batch)



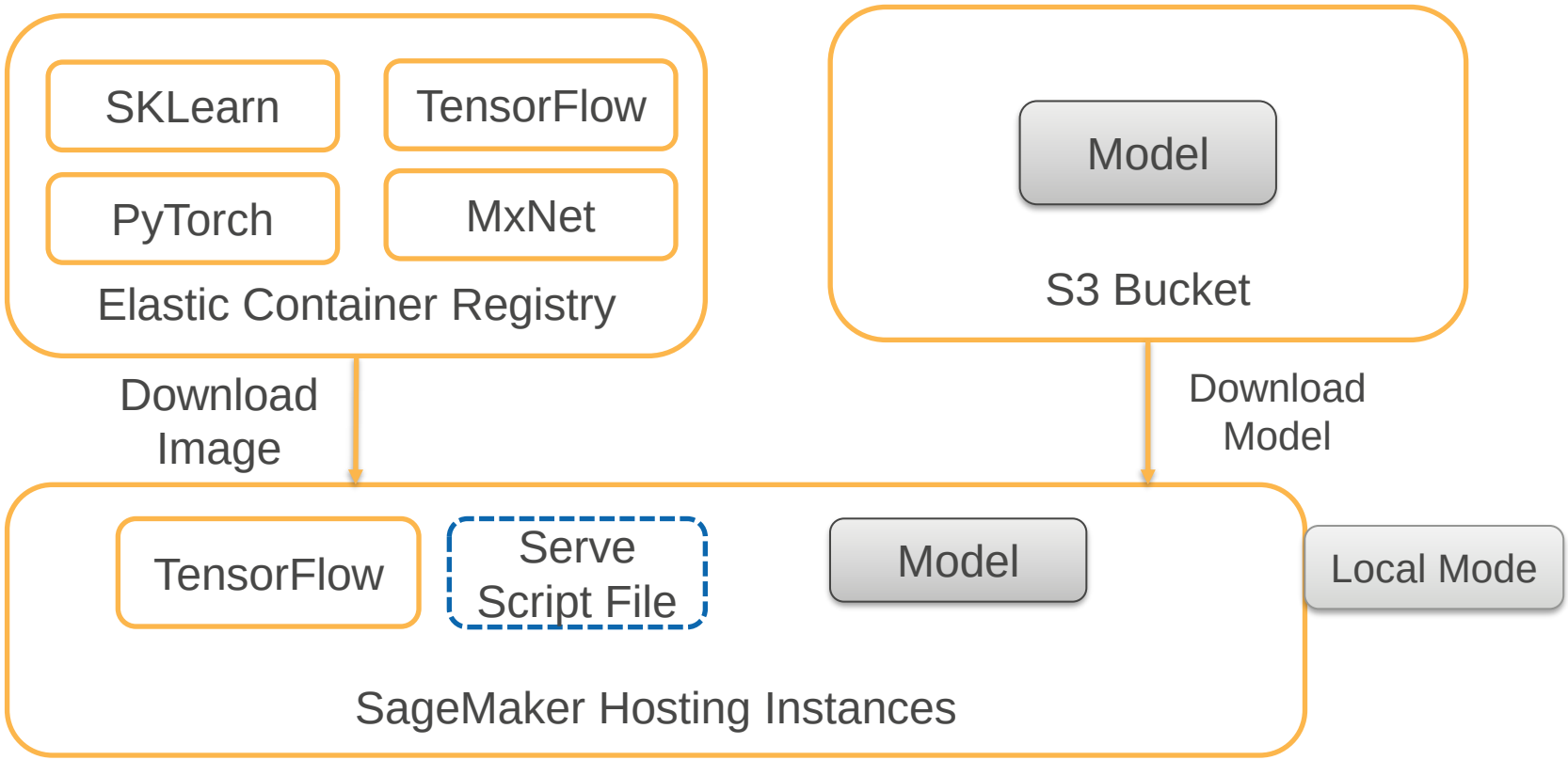
Custom Image – Training, Hosting



Framework - Training



Framework - Hosting



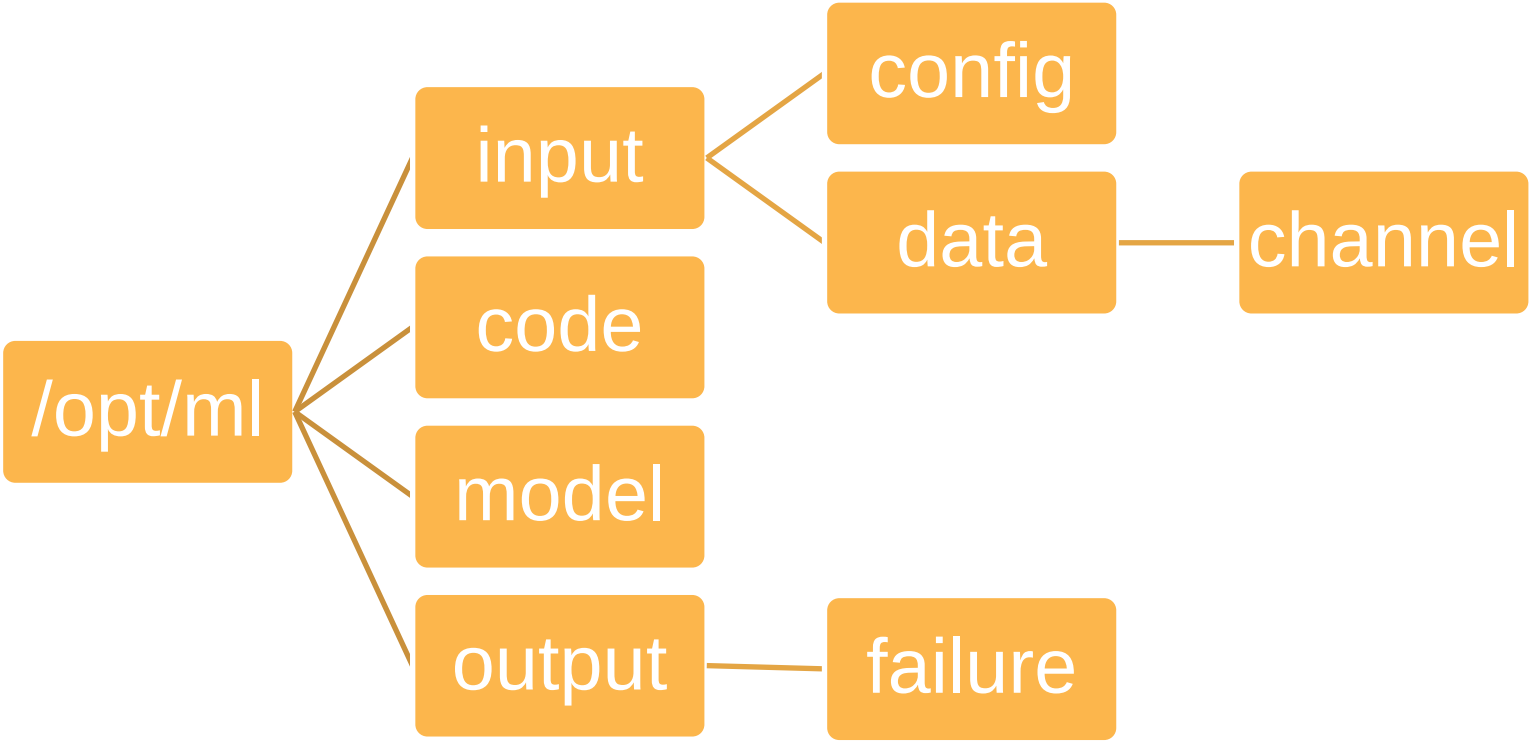
Bring Your Own Algorithm Training and Hosting

"Amazon SageMaker has certain contractual requirements that a container must satisfy to be used with it."

- Standard folder structure for reading data and resources
- Entry point that contains the code to run when container is started
- Instrumentation – Use StdOut, StdErr. SageMaker sends these message to CloudWatch log
- Metric Capture – Log metrics and define regex patterns to capture values from log
- One image for training and hosting (or) separate images (when compute resource requirements are substantially different)

Reference: https://docs.aws.amazon.com/sagemaker/latest/dg/amazon-sagemaker-containers.html

Container Folder Structure



https://docs.aws.amazon.com/sagemaker/latest/dg/amazon-sagemaker-containers.html

Container Folder Structure - Training

Folder	Purpose
/opt/ml/input/config/	• hyperparameters.json for training • resourceConfig.json - Container network layout for distributed training
/opt/ml/input/data/channel/	• channel = training, testing, … • Contains files for each channel /opt/ml/input/data/training/ /opt/ml/input/data/testing/
/opt/ml/input/data/channel_epoch/	• Channel = training, test, eval, … • Epoch = 0,1,2,… • Read the pipe to stream data from S3 for each epoch
/opt/ml/code/	• Scripts to run from container

Container Folder Structure – Training Output

Folder	Purpose
/opt/ml/model/	• Script should write the generated model to this directory • Store your model checkpoints and final output. • SageMaker uploads the content of model folder to your S3 bucket
/opt/ml/output/failure	• If the training fails, your script should write the error description to the failure file • SageMaker returns the first 1024 characters from this file as Failure Reason in the job description • SageMaker uploads content of output folder to your S3 bucket

Container Folder Structure – Hosting

Folder	Purpose
/opt/ml/model/	• Model files to use for inference
/opt/ml/code/	• Scripts to run from container

Important Environment Variables –Your script can use

Variable	Value & Purpose
SM_MODEL_DIR	/opt/ml/model – use this to store your model checkpoints and final output. SageMaker uploads this to your S3 bucket
SM_CHANNELS	Contains the list of input data channels in the container. Example: ["training“, "testing"]
SM_CHANNEL_{channel_name}	Directory containing channel data files Example: SM_CHANNEL_TRAINING='/opt/ml/input/data/training' SM_CHANNEL_TESTING='/opt/ml/input/data/testing'

Reference & Usage Examples: https://github.com/aws/sagemaker-containers#how-a-script-is-executed-inside-the-container

Important Environment Variables –Your script can use

Variable	Value & Purpose
SM_HPS	Contains a JSON encoded dictionary with the user provided hyperparameters Example: SM_HPS='{"batch-size": "256", "learning-rate": "0.0001","communicator": "pure_nccl"}'
SM_HP_{hyperparameter_name}	Contains value of the hyperparameter Example: SM_HP_LEARNING-RATE=0.0001 SM_HP_BATCH-SIZE=256 SM_HP_COMMUNICATOR=pure_nccl
NOTE: Hyperparameters are also provided as arguments to your script	

Reference & Usage Examples: https://github.com/aws/sagemaker-containers#how-a-script-is-executed-inside-the-container

Important Environment Variables –Your script can use

Variable	Value & Purpose
SM_HOSTS	JSON encoded list containing all the containers that are used for training Example: SM_HOSTS=["algo-1", "algo-2"]
SM_CURRENT_HOST	Name of the current container Example: SM_CURRENT_HOST=algo-1
SM_NUM_GPUS	The number of gpus available in the current container Example: SM_NUM_GPUS=1

Reference & Usage Examples: https://github.com/aws/sagemaker-containers#how-a-script-is-executed-inside-the-container

Lab – Bring Your Own Algorithm with SKLearnEstimator

- Develop scikit-learn model using scripts
- Train and host using SageMaker SKLearnEstimator
- Test using local mode
- Train and deploy on cloud Instance

Modified version of AWS Example: https://github.com/awslabs/amazon-sagemaker-examples/blob/master/sagemaker-python-sdk/scikit_learn_iris/Scikit-learn%20Estimator%20Example%20With%20Batch%20Transform.ipynb

Lab – Bring Your Own Algorithm TensorFlow Estimator

- Develop TensorFlow model using scripts
- Train and host using SageMaker TensorFlow Estimator
- Test using local mode
- Deploy to cloud instance

Modified version of AWS Example: https://github.com/awslabs/amazon-sagemaker-examples/blob/master/sagemaker-python-sdk/tensorflow_script_mode_training_and_serving/tensorflow_script_mode_training_and_serving.ipynb

Optional Lab – Built your own container

Most complex requires knowledge of Docker Containers, Web Stack for hosting

Walk through the code example here:

https://github.com/awslabs/amazon-sagemaker-examples/blob/master/advanced_functionality/scikit_bring_your_own/scikit_bring_your_own.ipynb

Chandra Lingam

50,000+ Students

Up-to-date Content

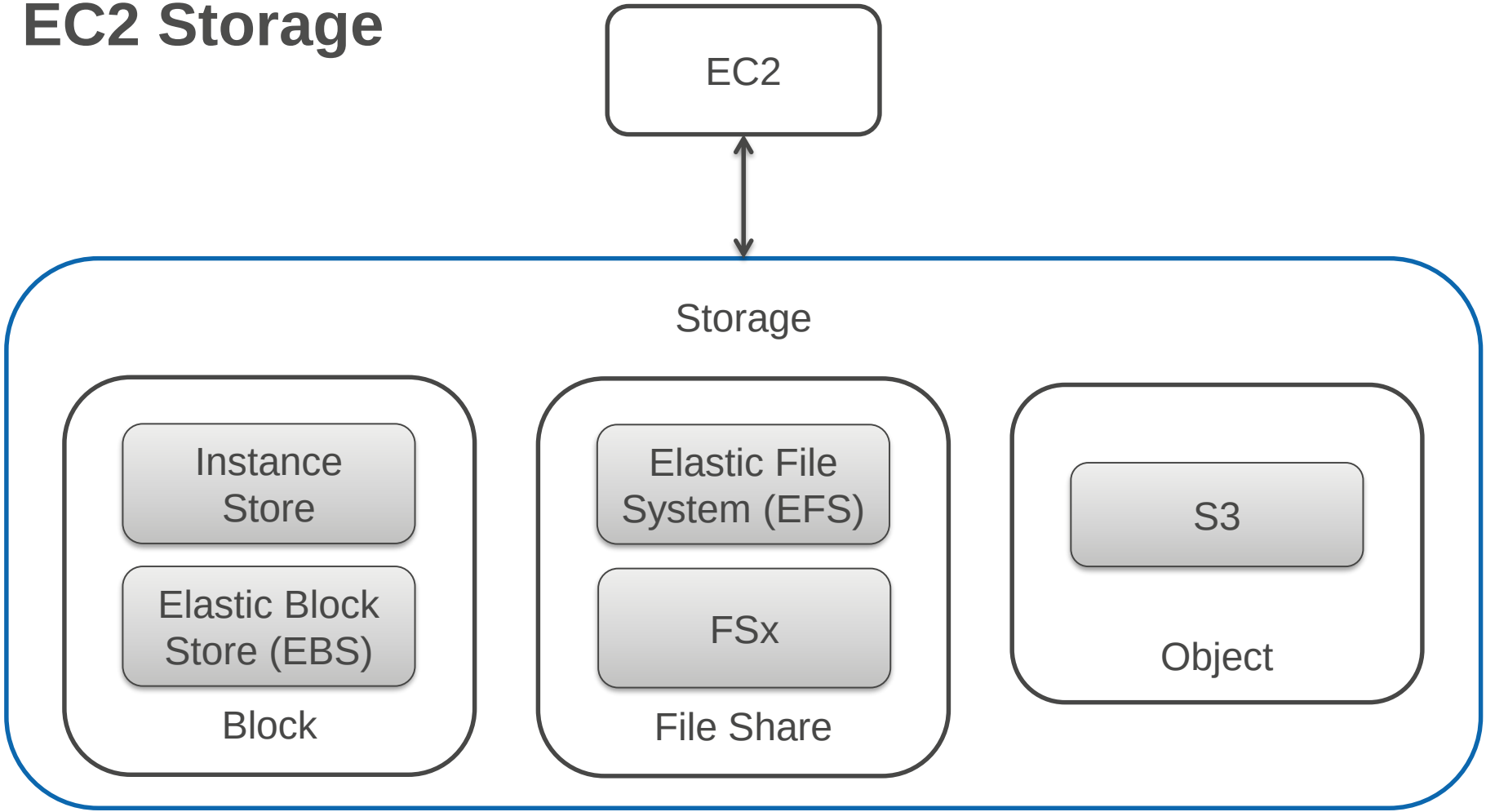
EC2 Storage

Options, Usage

Chandra Lingam

Cloud Wave LLC

EC2 Storage



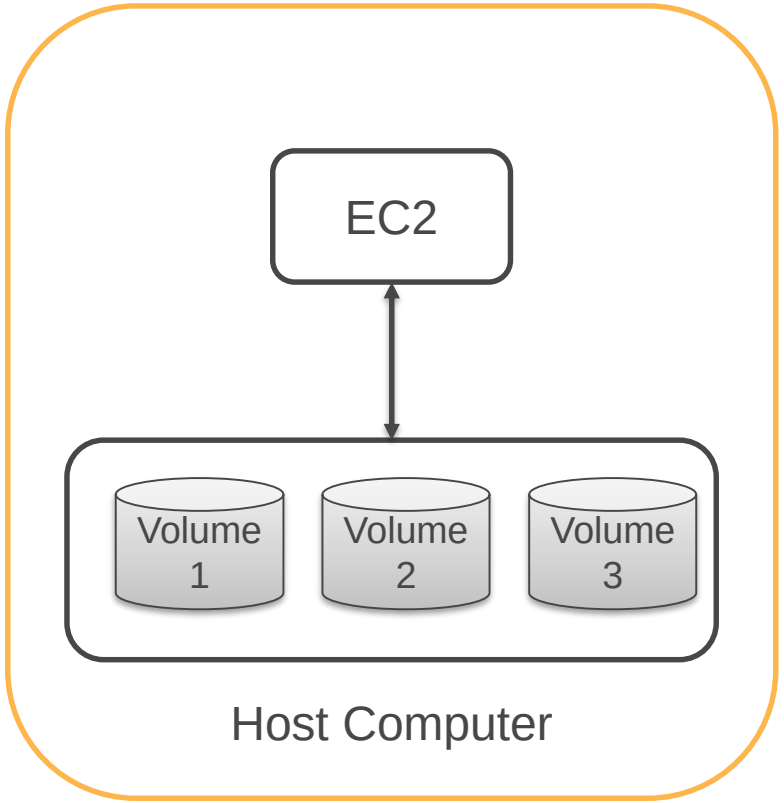
Storage Units

- In the context of computer memory,
  - 1 KB = 1,024 Bytes (2^10)
  - 1 MB = 1,024 KB
- In the context of SSD/HDD
  - 1 KB = 1,000 Byte (10^3)
  - 1 MB = 1,000 KB
- Amazon uses KiB (Kibibyte), MiB (Mebibyte), GiB (Gibibyte) Standard units – Matches with memory
  - 1 KiB = 1,024 Bytes
  - 1 MiB = 1,024 KiB

Block Storage

Instance and Elastic Block Store (EBS)

Instance Store (Block)



Storage of host computer is assigned to EC2 instance

Temporary Storage

Highest Performance

Storage included as part of instance pricing

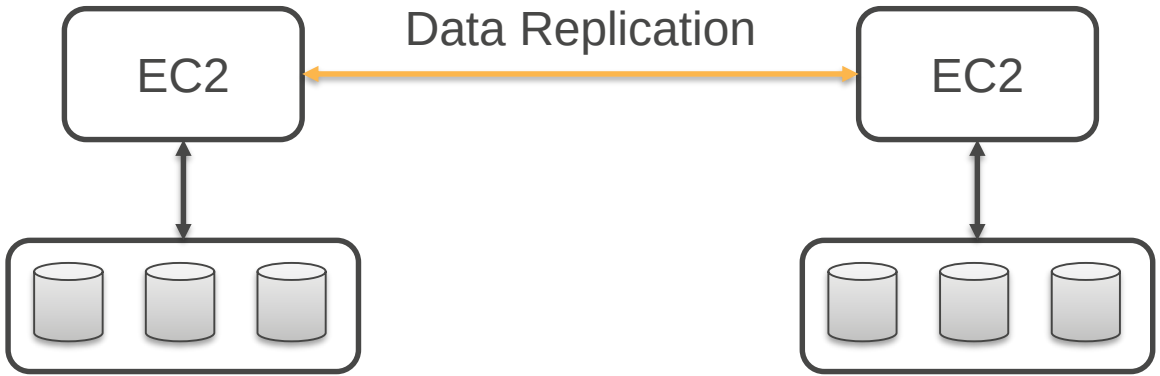
Instance Store Durability

Data persists only for the lifetime of the instance

Reboot – Data Persists

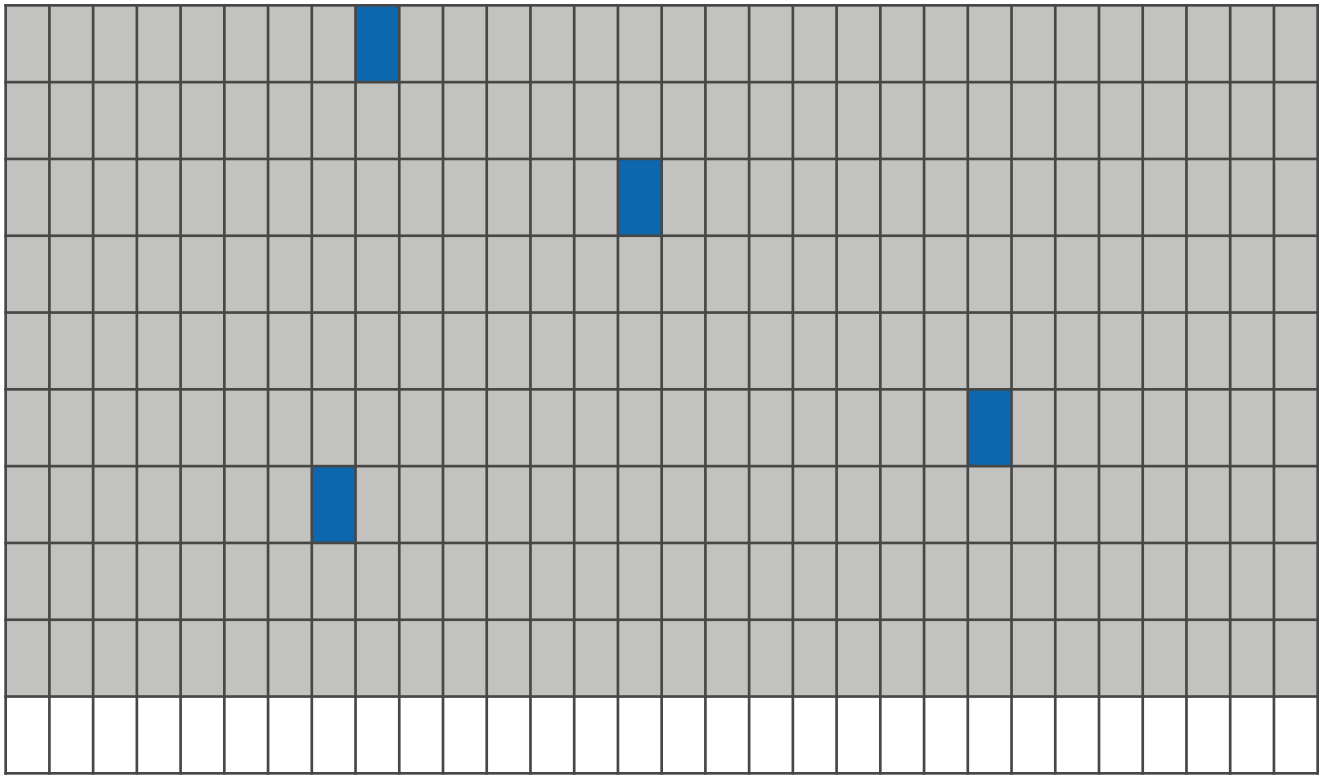
Data is lost – when underlying hardware fails, instance stops, or instance terminates

Instance Store – Do it yourself High Availability and Durability



1. Replicate Data to multiple instances to prevent data loss.
   Example: Hadoop File System, MongoDB
2. Backup using OS provided software or third-party software

Random I/O Workload – SSD Preferred



Allocated Used Read/Write

Performance Measured in IOPS

SSD based Instance Store offers very high IOPS

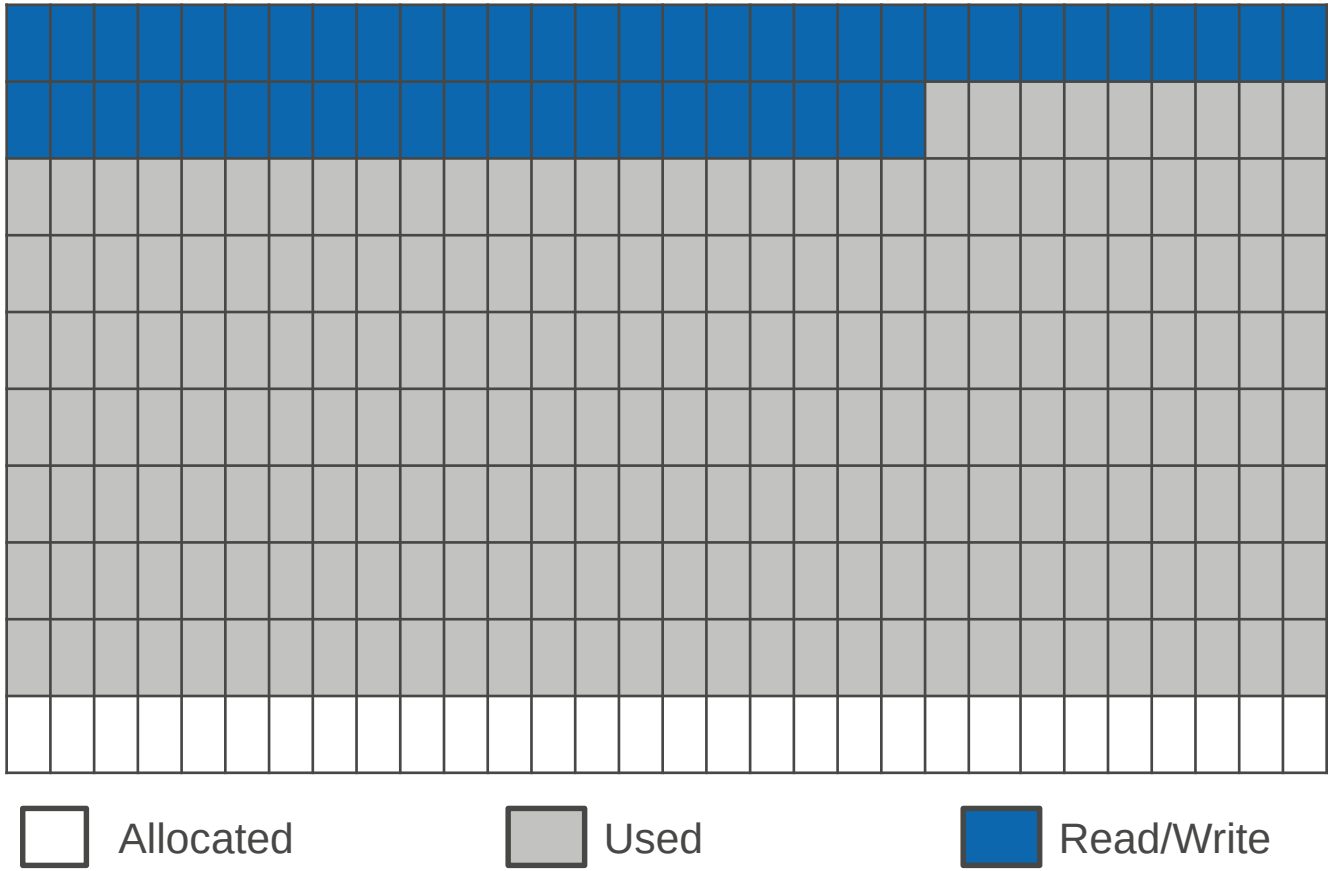
100,000 IOPS to Millions of IOPS (Based on Instance Type)

SSD I/O Performance

If you use a Linux AMI with kernel version 4.4 or later and use all the SSD-based instance store volumes available to your instance, you get the IOPS (4,096 byte block size) performance listed in the following table (at queue depth saturation). Otherwise, you get lower IOPS performance.

Instance Size	100% Random Read IOPS	Write IOPS
i3.large *	100,125	35,000
i3.xlarge *	206,250	70,000
i3.2xlarge	412,500	180,000
i3.4xlarge	825,000	360,000
i3.8xlarge	1.65 million	720,000
i3.16xlarge	3.3 million	1.4 million
i3.metal	3.3 million	1.4 million
i3en.large *	42,500	32,500
i3en.xlarge *	85,000	65,000
i3en.2xlarge *	170,000	130,000
i3en.3xlarge	250,000	200,000
i3en.6xlarge	500,000	400,000
i3en.12xlarge	1 million	800,000
i3en.24xlarge	2 million	1.6 million

Sequential I/O Workload – Magnetic/HDD Preferred

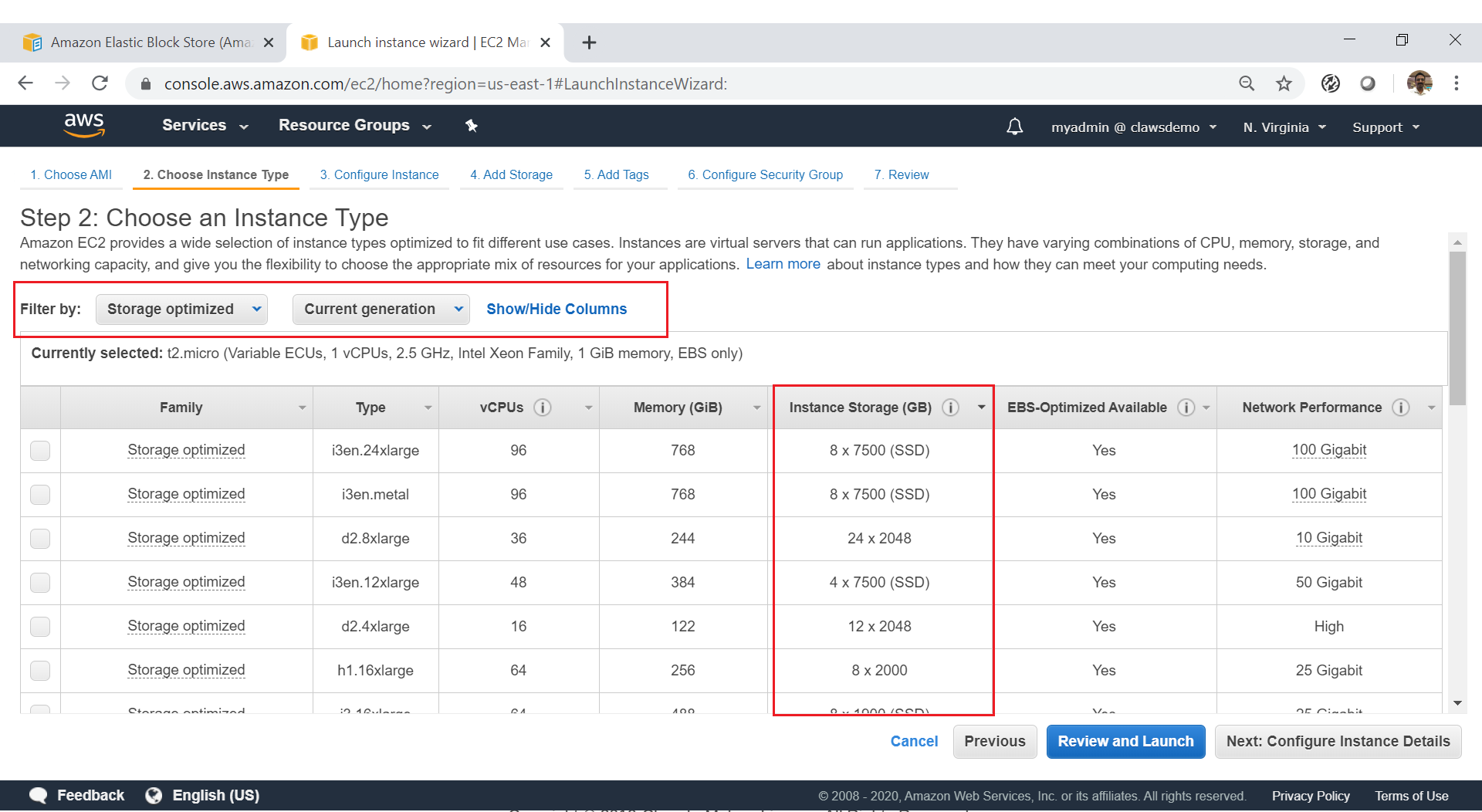


Performance Measured in Throughput (MiB/s)

Magnetic based Instance Store offers very high throughput

Lower Cost

SSD can be used – but a higher cost



Uses - Storage Optimized Instances

D2, H1 – Magnetic

- High Throughput Access to Large Datasets
- Massively Parallel Processing Data Warehouse (MPP)

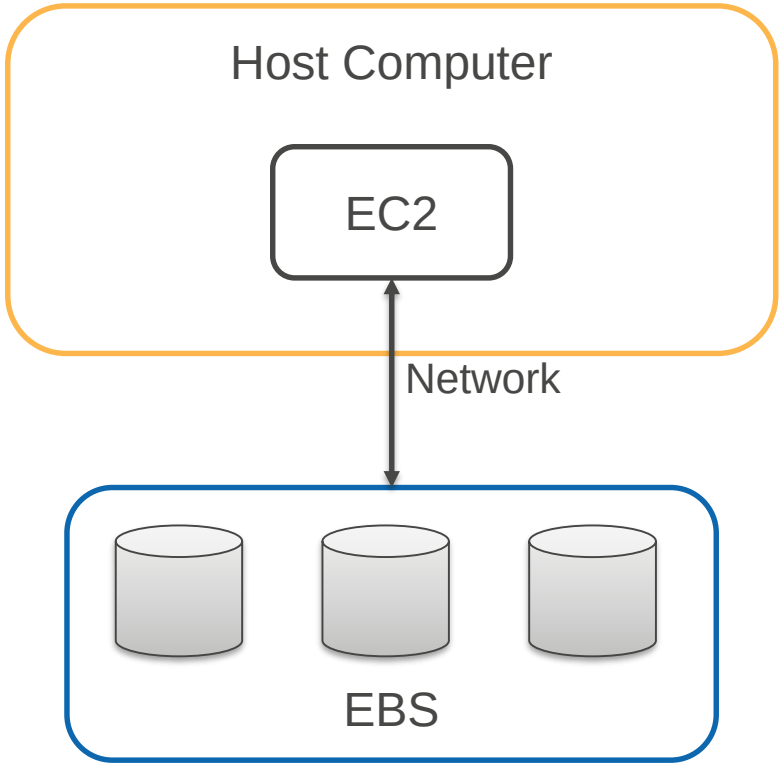
  For Example, a Redshift like solution
- MapReduce, Hadoop Distributed Compute
- Log or data processing

Uses - Storage Optimized Instances

I3 – SSD

- High frequency Online Transaction Processing Systems (OLTP)
- NoSQL Databases
- Relational Databases
- Data warehousing
- Cache for in-memory databases like Redis
- Distributed File Systems

Elastic Block Store (EBS)



EBS is a managed block storage service

Storage volume is outside of host computer – Long term persistence

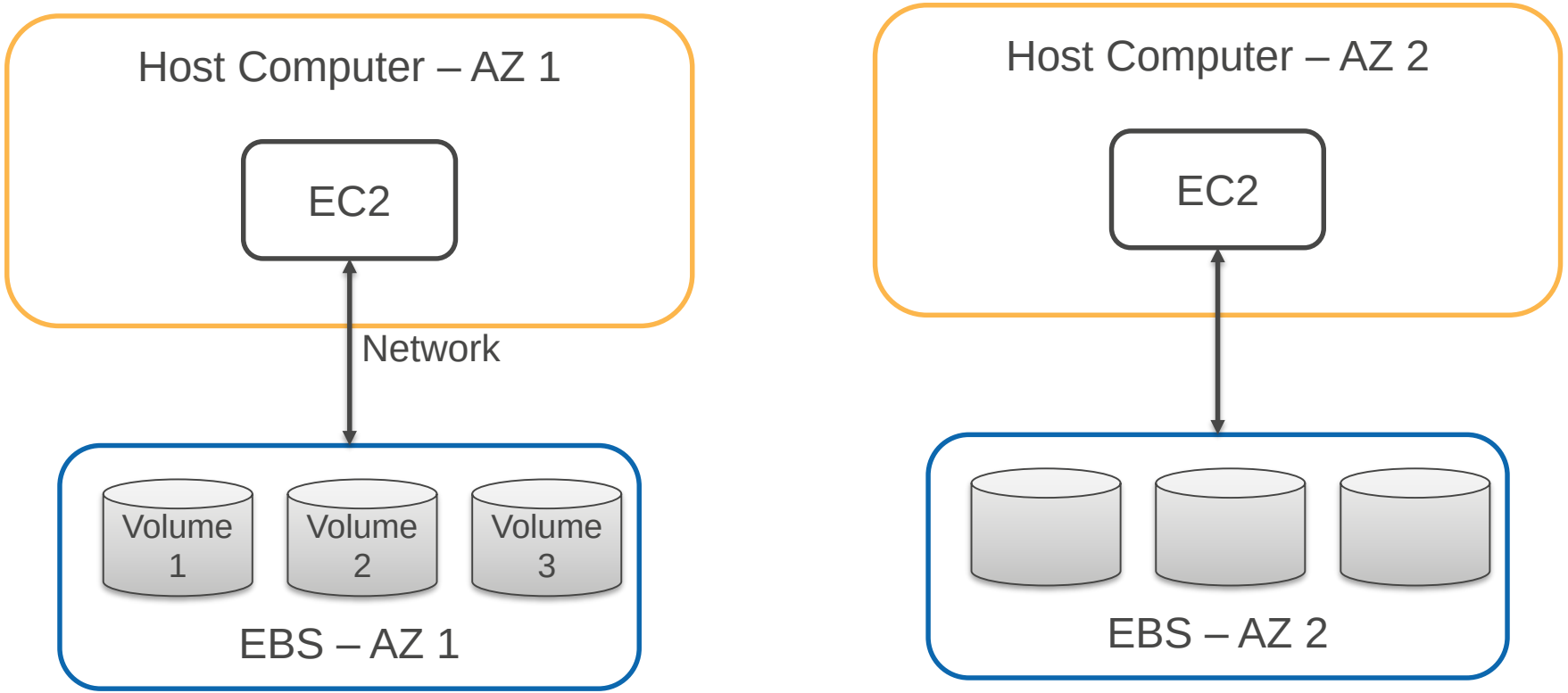
EC2 instance use EBS storage volume as a block device

You need to pay for allocated EBS storage

EC2 Benefits with EBS

- Stop-Start EC2 Instance
- Persist EBS volumes for terminated instances
- Detach and attach volume to a different instance in the same Availability zone
- Built-in Snapshot Capability for incremental backup to S3
- Create Amazon Machine Image (AMI) from Snapshots to launch new EC2 instances

Elastic Block Store (EBS) – Availability Zone Level



EBS Features

EBS volumes are created at Availability Zone level

Highly available and durable - Volume is replicated within the Availability Zone – to protect against hardware failure

Built-in Snapshot Capability for incremental backup to S3

Create volumes from Snapshot (any Availability Zone in the Region)

Copy Snapshots to another Region (disaster recovery, expansion)

EBS - Uses

Enterprise Applications

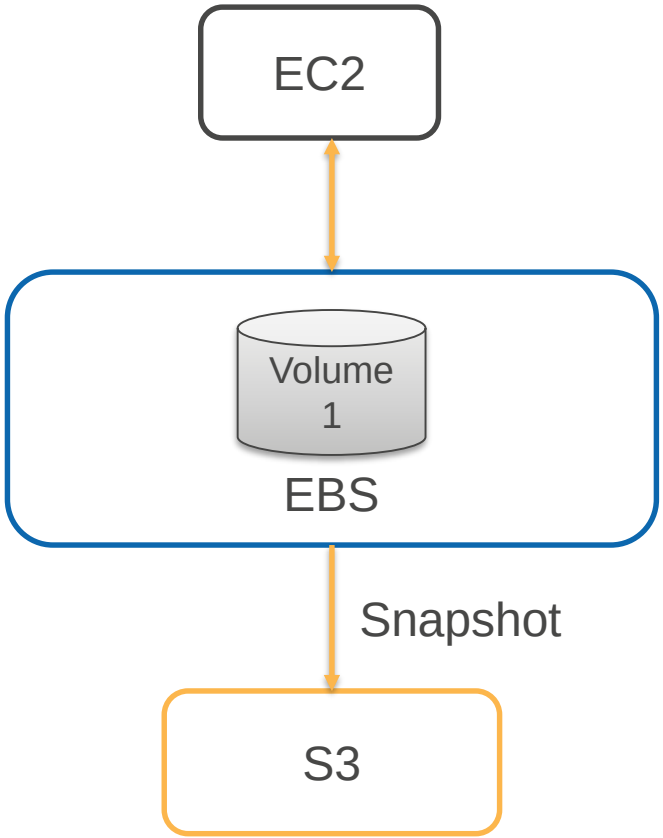
Relational Databases

NoSQL Databases

Big data analytics

Media Workflows

Snapshot



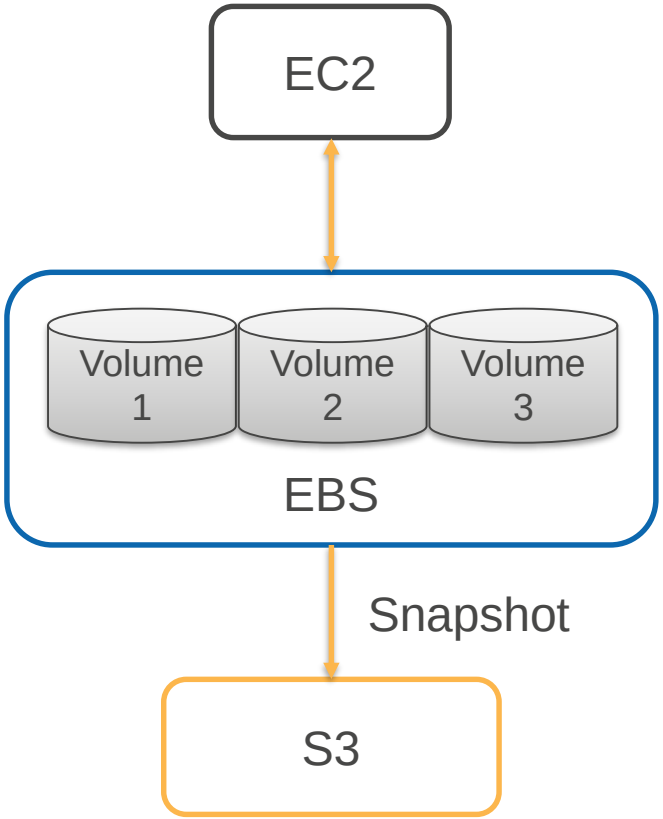
Snapshot – Incremental Backup of EBS Volume to S3

Point-in-time – Volume needs to be in consistent state (all application cache needs to be flushed and no traffic) when snapshot command is issued

Snapshot is async and works in the background

System can start using the volume – no need to wait for backup to complete

Snapshot – Multiple Volumes



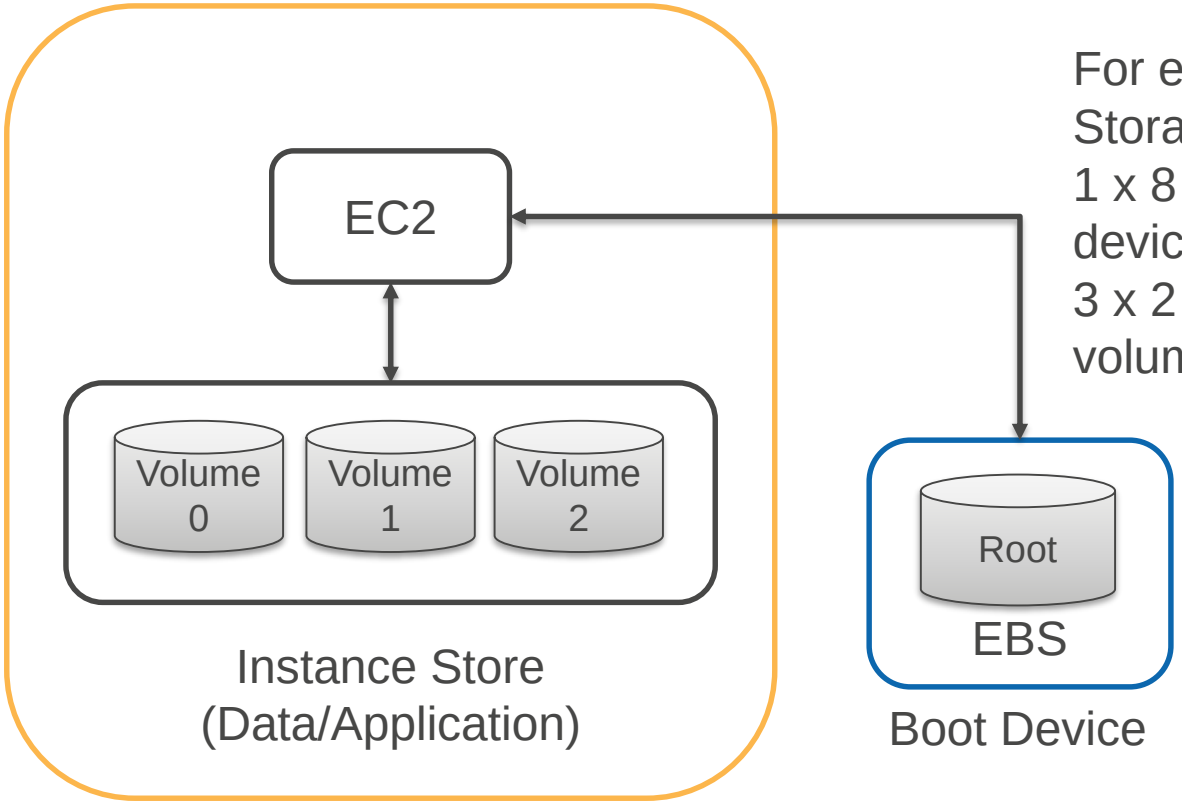
In this example, EC2 instance is using three volumes. You can use any of the following options:

1. Stop the system and issue snapshot for each volume and restart
2. Take backup at application level (for example, using database software)
3. Use the **New** multi-volume consistent snapshot

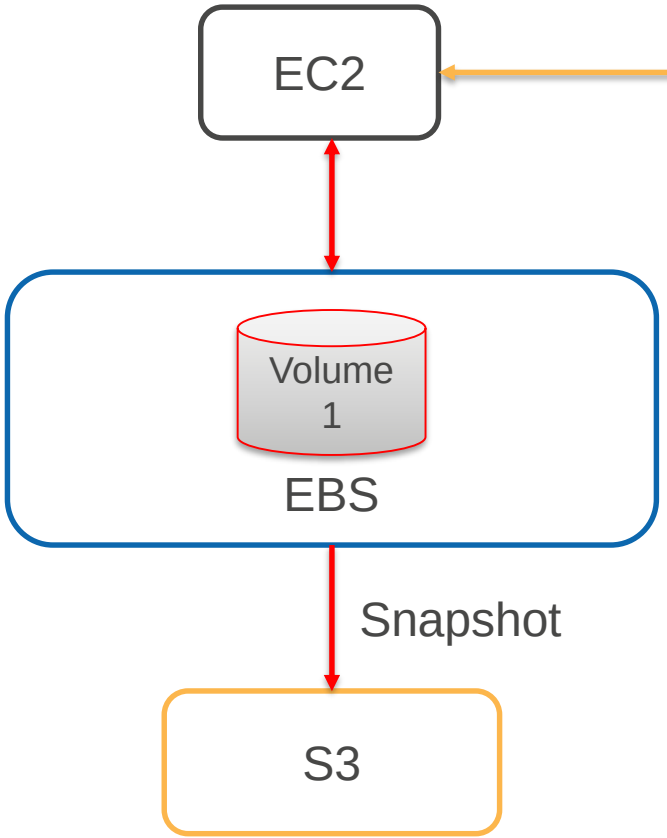
Mix-and-Match

Mix and match volumes of different types

For example,
Storage Optimized (d2.xlarge)
1 x 8 GiB EBS volume as boot device and
3 x 2 TiB HDD Instance Store volumes for application use



Encryption



EBS supports volume encryption

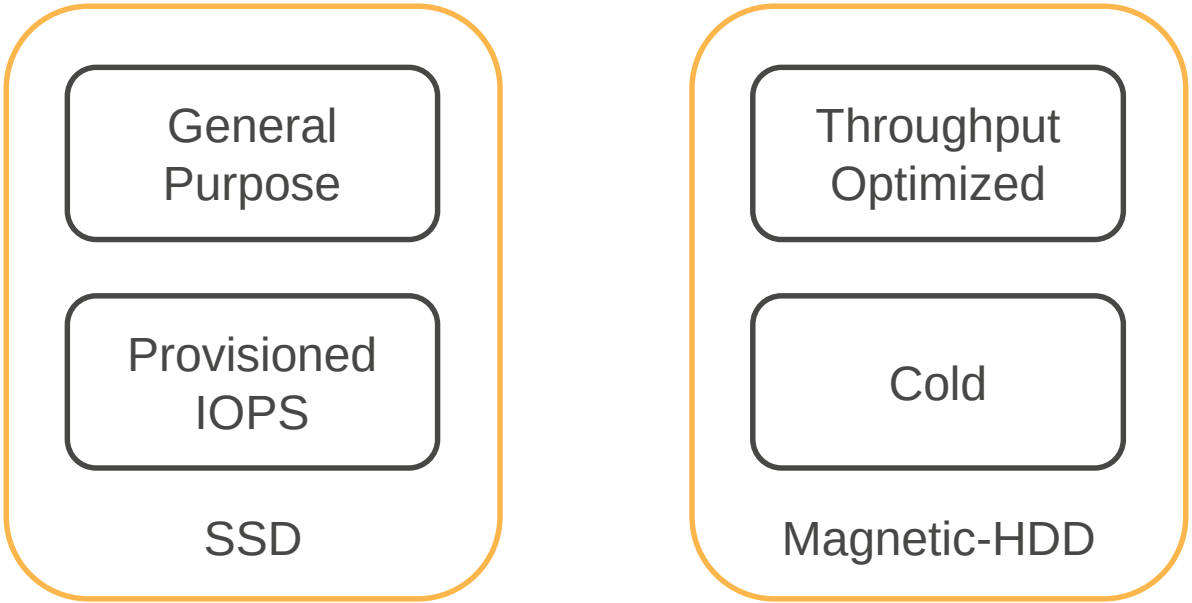
Data in transit from host computer and at rest in EBS volume is encrypted

Snapshots are encrypted

Restored volumes are also encrypted

Encryption Key – EBS managed Key or Customer Master Key (CMK). CMK gives you fine grained access control to encrypted data.

EBS Volume Types



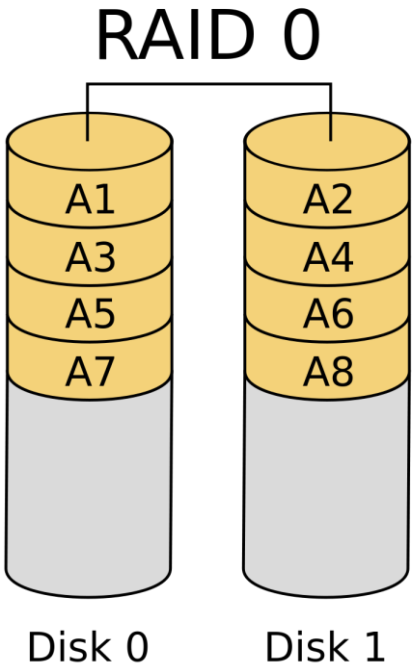
EBS – General Purpose SSD (gp2)

"Balances price-performance for a wide variety of workloads"

Feature	Value
Baseline IOPS per GiB	3
Burst IOPS	3,000
Max IOPS/Volume	**16,000 (at 5.3 TiB storage or larger)**
Max Throughput (MiB/s)	250
Size	1 GiB – 16 TiB

NOTE: Review the above link for up-to-date Max IOPS data

RAID using EBS volumes



One way to increase IOPS, Throughput, Volume is by using RAID 0

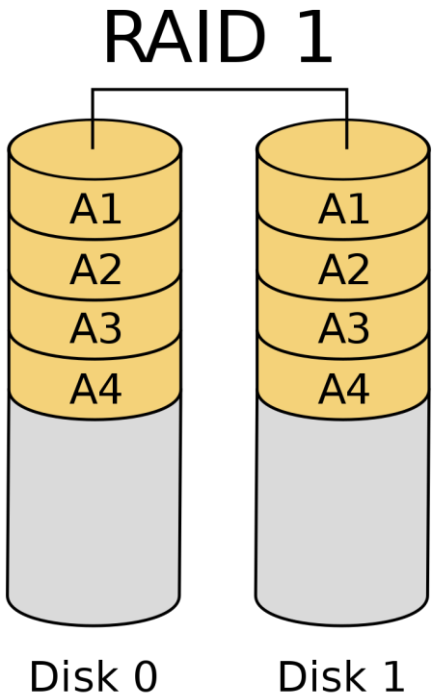
RAID 0 stripes the blocks across two EBS volumes

Each volume in-turn is automatically replicated by EBS service for high availability and durability

RAID 0 doubles the IOPS, Throughput and Size

RAID requires additional software components that you need to manage

Image Credit: Cburnett,
https://commons.wikimedia.org/w/index.php?curid=1509082
https://en.wikipedia.org/wiki/Standard_RAID_levels



RAID 1 mirrors the blocks

EBS volumes are already replicated – so there is no benefit in using RAID 1

RAID 1 is generally not used with EBS volumes

Uses - General Purpose SSD

Boot Volumes

Small to Medium databases

Development and test environment

EBS – Provisioned IOPS SSD (io1)

"Highest performance volume for latency sensitive transactional workloads"

Feature	Value
Max IOPS/Volume	**64,000**
Max Provisioned Performance to Volume Size (GB)	50:1 i.e., you can Provision IOPS up to 50 times the volume size in GiB
Max Throughput (MiB/s)	1,000
Size	4 GiB – 16 TiB
Provisioned Performance Guarantee	99.9% of the time

NOTE: Review the above link for up-to-date Max IOPS data

Uses - Provisioned IOPS SSD (io1)

Boot Volumes

Critical Business applications

Large databases – Cassandra, MongoDB, SQL Server, Oracle, PostgreSQL, MySQL

EBS – Throughput Optimized HDD (st1)

“Low cost volume designed for frequently accessed, throughput intensive workloads”

Feature	Value
Max IOPS/Volume	500
Max Throughput (MiB/s)	500
Size	500 GiB – 16 TiB

NOTE: Review the above link for up-to-date data

Uses - Throughput Optimized HDD (st1)

Big data

Data warehouse

Log processing

EBS – Cold HDD (sc1)

"Low cost volume designed for infrequently accessed workloads"

Feature	Value
Max IOPS/Volume	250
Max Throughput (MiB/s)	250
Size	500 GiB – 16 TiB

NOTE: Review the above link for up-to-date data

Uses - Throughput Optimized HDD (st1)

Inexpensive storage

Ideal for infrequently accessed sequential workloads

Pricing Example

Volume Type	Price per GB-month	IOPS Cost per month	Size (GB)	IOPS - Provisioned	Storage Cost	IOPS Cost	Total Monthly Cost
General Purpose SSD	0.100	-	500.00	Baseline 1500 IOPS (500x3) and can burst up to 3000	50.00		50.00
Provisioned IOPS SSD	0.125	0.065	500.00	3,000.00	62.50	195.00	257.50
Provisioned IOPS SSD	0.125	0.065	500.00	5,000.00	62.50	325.00	387.50
Throughput Optimized HDD	0.045		500.00		22.50		22.50
Cold HDD	0.025		500.00		12.50		12.50

AWS Simple Monthly Calculator - https://calculator.s3.amazonaws.com/index.html

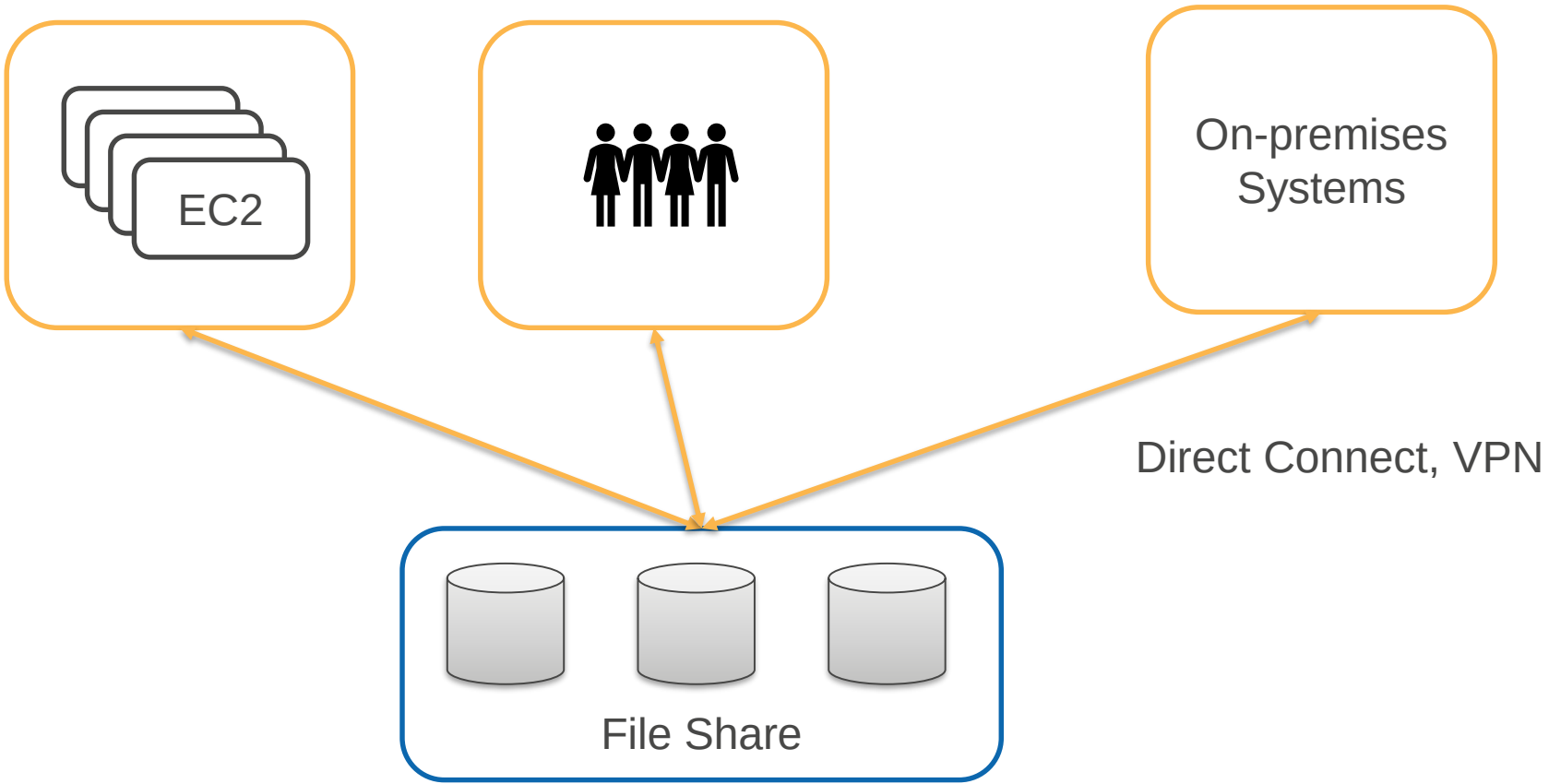
File Share

Elastic File System (EFS)

FSx for Windows

FSx for Lustre

AWS File Share Solution



AWS Managed File Shares - Uses

- User Home Directories
- Lift-Shift Enterprise Applications to Cloud
- Analytics – Data Storage
- Web Content – Common share for all webservers
- Media Workflows – Video Editing and Production
- Database backups – easily mount to database servers and take backup

File Storage Services

Service	Purpose
Elastic File System (EFS)	File share for Linux EC2 instances on AWS
FSx for Windows	File share for Windows EC2 instances on AWS
FSx for Lustre	File share Optimized for High Performance Computing – Linux EC2 instances Access S3 as a file share (like Storage Gateway)

- Fully managed – automatically grow and shrink
- High Durability and Availability – Replicated across multiple availability zones in a region
- All these file shares can be used from On-Premises using AWS Direct Connect or VPN

Elastic File System (EFS)

File Share for Linux EC2 instances on AWS

NFS Filesystem and traditional file permissions model

Standard and In-frequent Access Storage Tiers (Similar to S3)

Lifecycle Management – to move files to lower cost storage

FSx for Windows

File share for Windows Systems

NTFS Filesystem using SMB Protocol

Integrated with Active Directory

FSx for Lustre

High performance file share for Linux Systems

Optimized for high performance computing and fast processing

Two modes:

- Standalone File share or
- Link to S3 Bucket and access S3 as a file share

Chandra Lingam

50,000+ Students

Up-to-date Content

AWS Databases

Relational, NoSQL, In-Memory, Data warehouse, Specialized

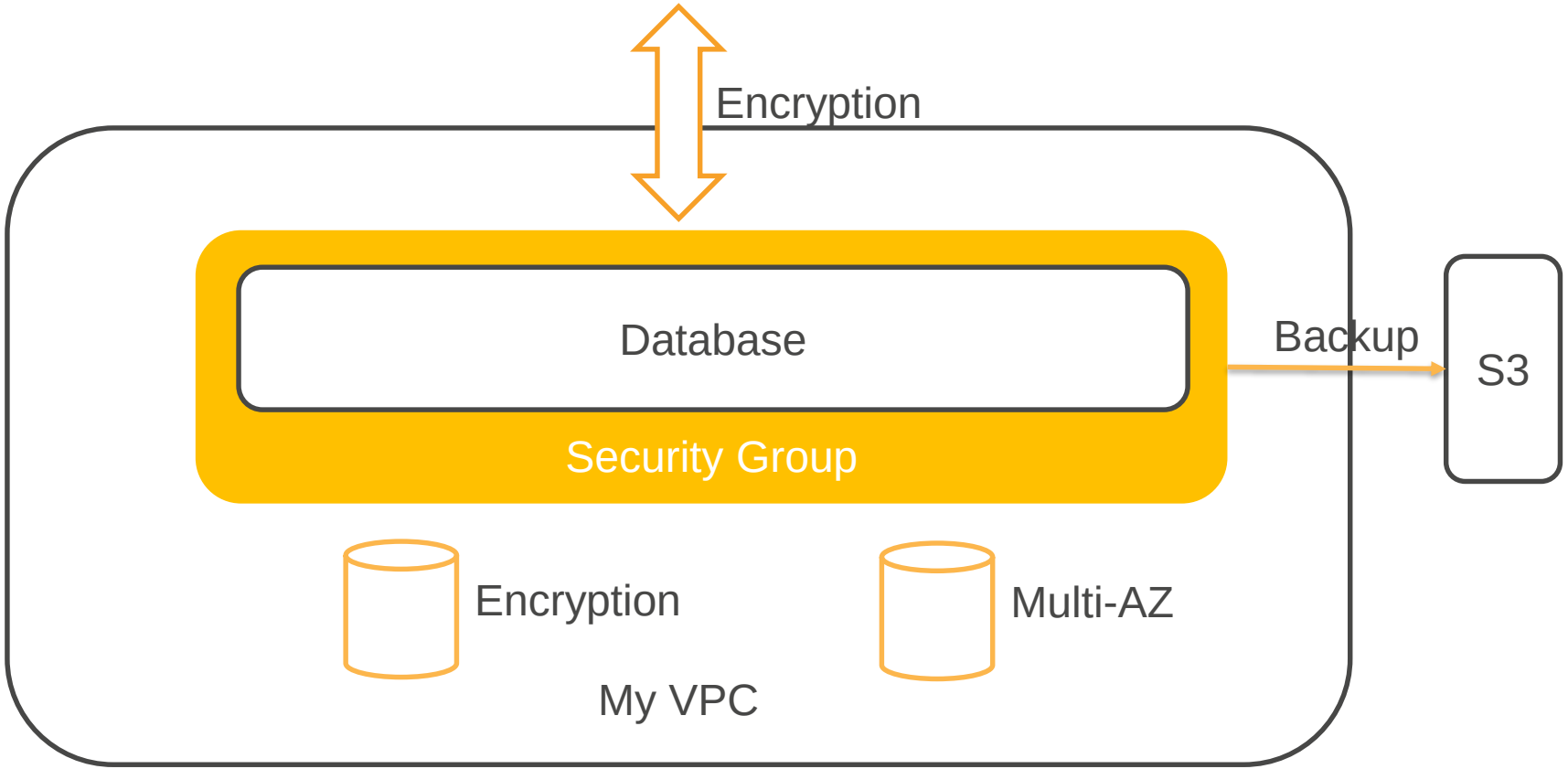
Chandra Lingam

Cloud Wave LLC

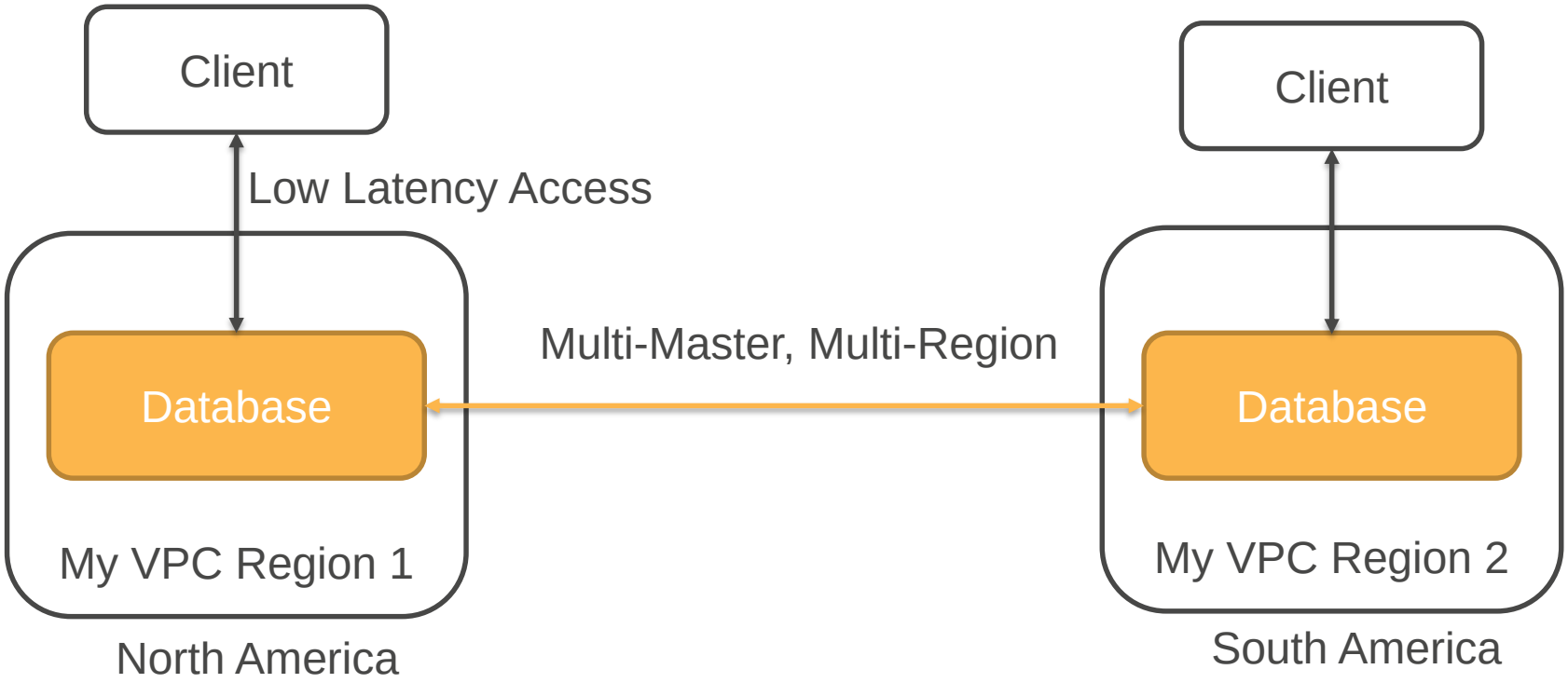
AWS Databases

Note: Not complete list

Standard Features – AWS Databases



DynamoDB Global Table - Multi-Region, Multi-Master



Benefits

- Wide selection of database engines
- Fully managed
- VPC Network Isolation
- Encryption at rest using KMS
- Encryption in transit
- Automated Backup
- Highly Durable and Available – Replicated across multiple devices in Availability Zone, Region
- Multi-Region, Multi-Master (some products) – Low latency access and disaster recovery

AWS Portfolio of Databases (1 of 2)

Service	Type of Database
RDS - Relational Database Service	Relational Database. Choice of database engines - Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, SQL Server Uses: Traditional applications, ERP, CRM, ecommerce
Redshift	Petabyte scale Data warehouse, Massively Parallel Columnar Storage, integrates with S3 data lake Uses: business intelligence, analytics, SQL to explore data lake
DynamoDB, Cassandra, DocumentDB	NoSQL Database Key-value storage, document store, consistent single digit millisecond latency at any scale Uses: high traffic web applications, ecommerce, gaming systems
ElastiCache	In-memory database - MemCached, Redis Sub-millisecond latency Uses: Caching, user session, gaming leaderboards, geospatial applications

AWS Portfolio of Databases (2 of 2)

Service	Type of Database
Neptune	Graph Database – optimized for highly connected datasets and querying relationships Uses: Social networks, recommendation engines
Timestream	Timeseries Database – optimized for storing and querying high volume timeseries data at 1/10th the cost of relational databases Uses: IoT applications, Industrial telemetry, DevOps
Quantum Ledger Database	Ledger Database – Blockchain based system for transparent, immutable, and cryptographically verifiable transaction log Uses: Systems of record, supply chain, banking transactions
Elasticsearch	Search database, store, analyze and correlate logs from disparate applications and systems Uses: search, infrastructure and application monitoring, Security info and event management

Database Migration

AWS Database Migration Service (DMS)

- One-time data replication
- Continuous data replication from on-premises to AWS (and reverse)
- Homogeneous and Heterogeneous replication

Summary

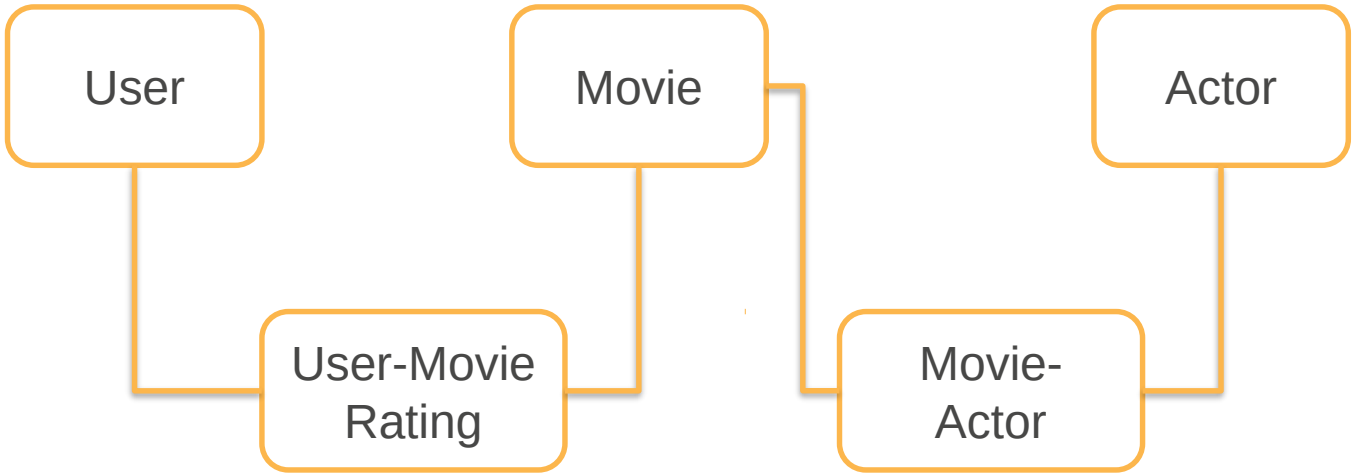
"The broadest selection of purpose-built databases for all your application needs"

"By picking the best database to solve a specific problem or a group of problems, you can breakaway from restrictive one-size-fits-all monolithic databases“

Reference: https://aws.amazon.com/products/databases/

Relational Database Service (RDS)

Relational Database



Relational Database

General Purpose – Design a schema for any need

Rigid Schema – difficult to change

SQL – Flexible Querying System

Complex System

Scaling Challenges

Amazon Relational Database Service (RDS)

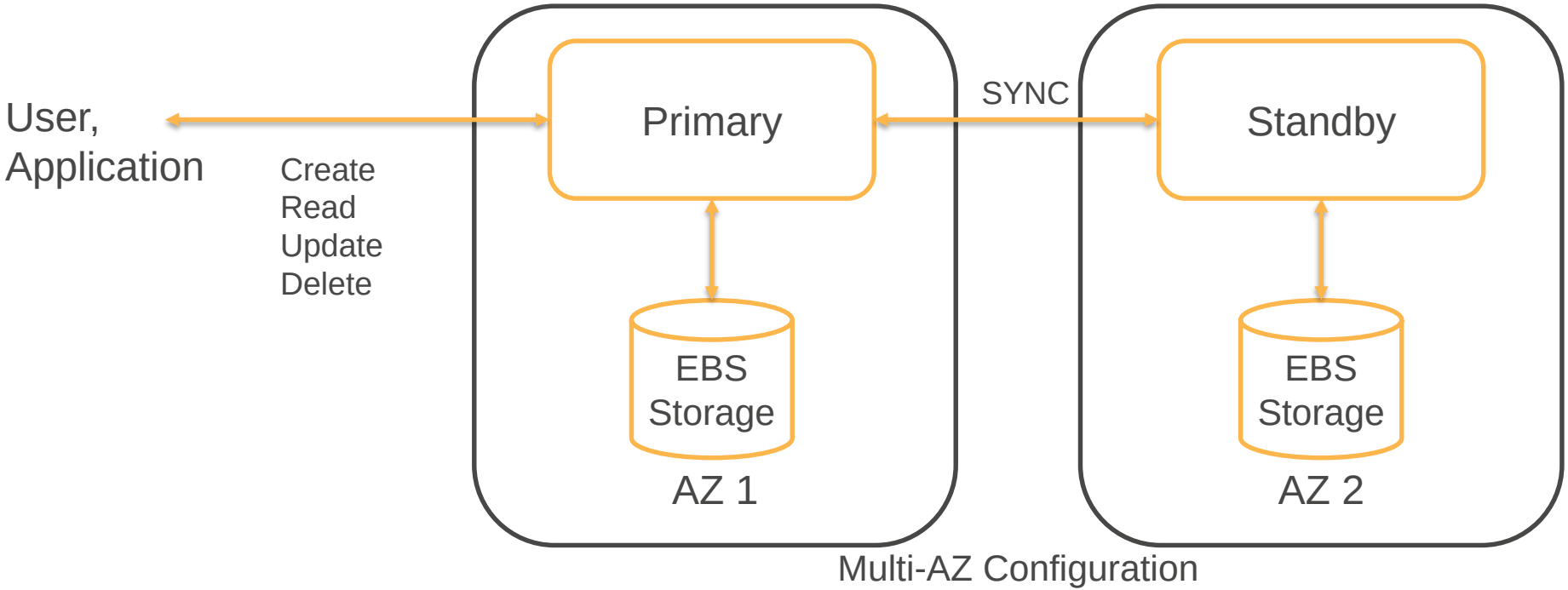
Automates time-consuming administrative tasks (hardware, installation, patching, backup)

Production ready database in minutes

Push button scaling (CPU, Memory, Storage)

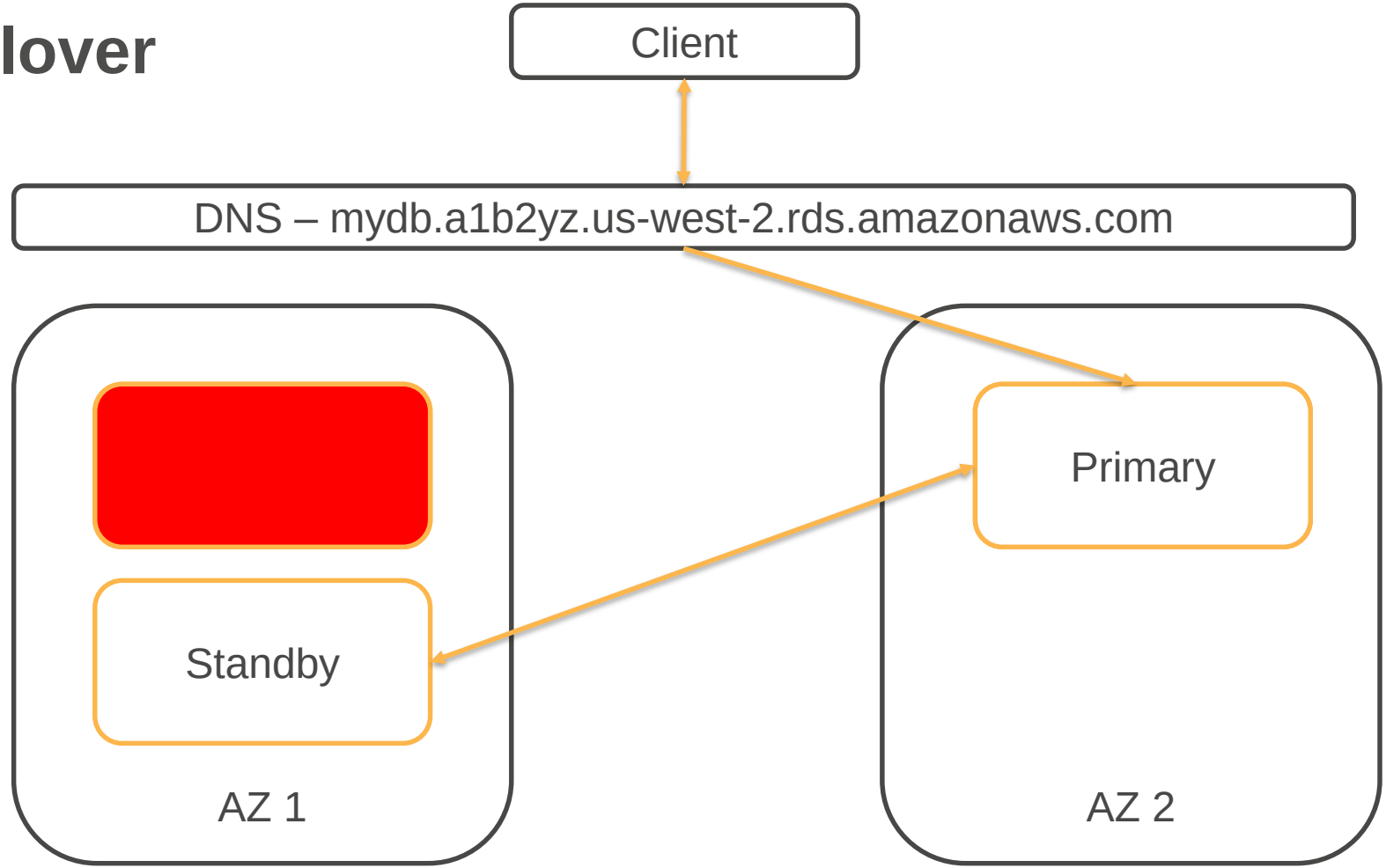
Six popular database engines: Aurora, MySQL, PostgreSQL, MariaDB, Oracle, SQL Server

Amazon Relational Database Service (RDS)

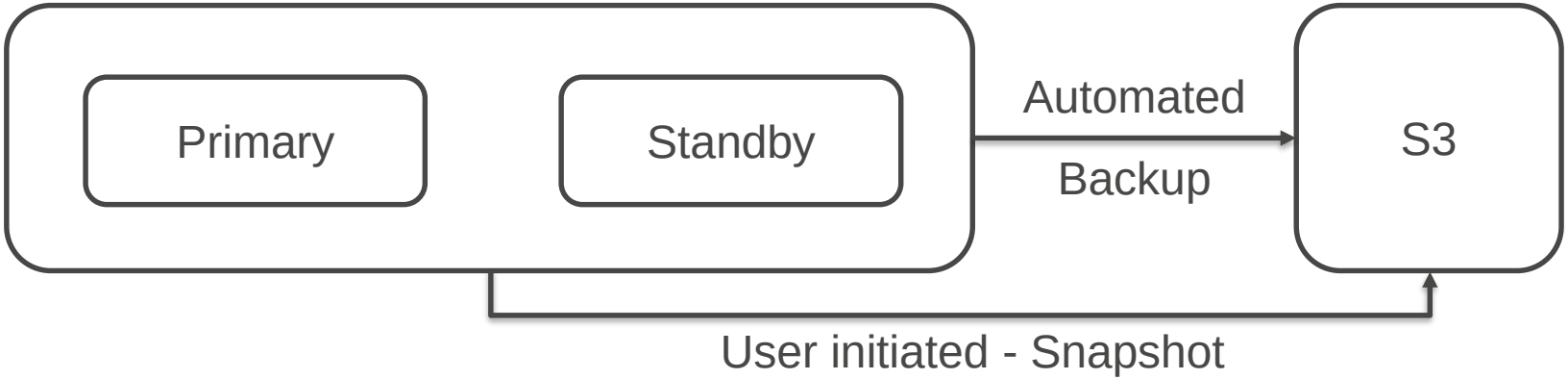


- Connect using DNS Name
- RDS maintains mapping between DNS Name and Primary Instance
- After failover, DNS is updated to point to new primary

Failover



RDS - Backup and Snapshot



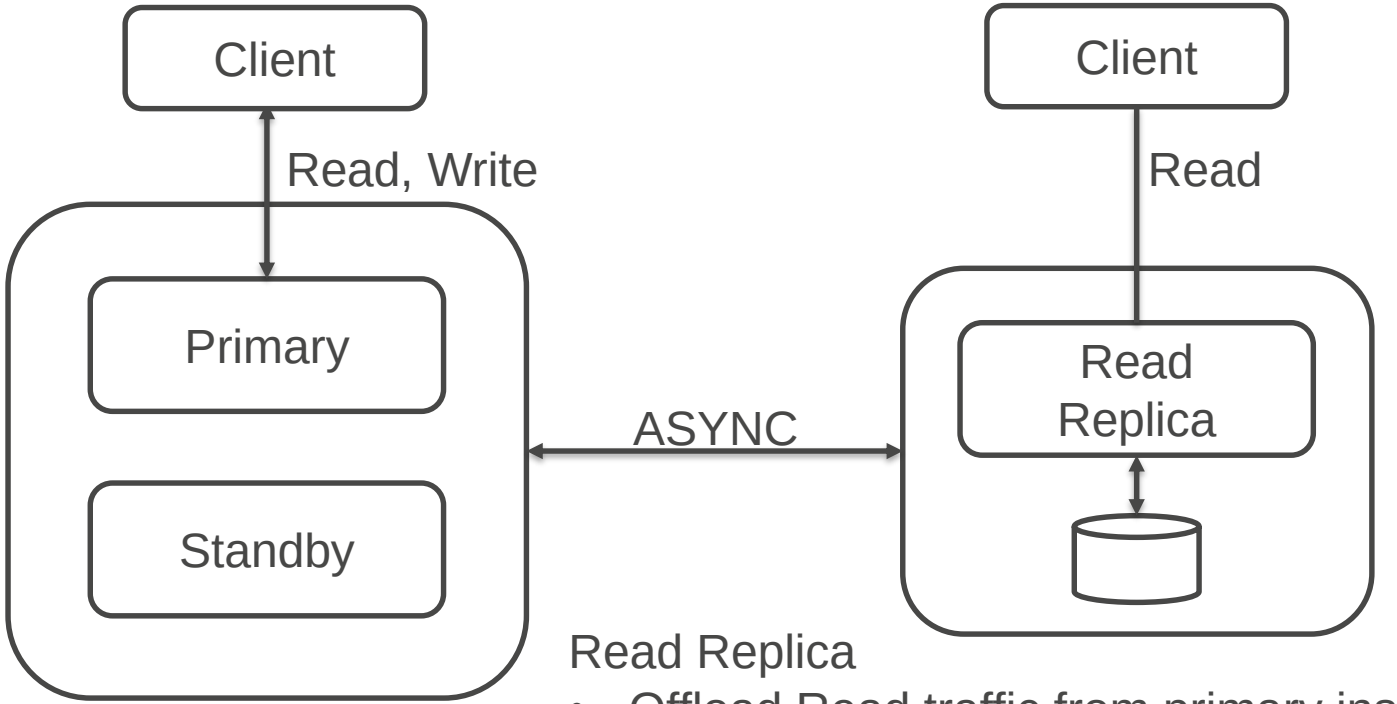
Automated Backup

- Configurable for a retention up to 35 days
- Last restorable time – typically within last 5 minutes
- Point-in-time restore up to specified second (to a new instance)

Snapshot

- User initiated
- Snapshot is kept until explicitly deleted
- Suitable for long term retention
- Copy to another region

RDS – Read Replica



Read Replica

- Offload Read traffic from primary instance
- Data can be stale
- One or more read replicas (depending on DB engine)

RDS Patching

"Amazon RDS will make sure that the relational database software powering your deployment stays up-to-date with the latest patches.“

You can specify a maintenance window that RDS can use for patching systems

RDS – Scaling CPU and Memory

- Specify desired CPU and Memory configuration and RDS takes care of scaling
- Completes in a few minutes (needs to spin up new instances)
- RDS performs failover during compute scaling (interruption to client for the duration of failover)

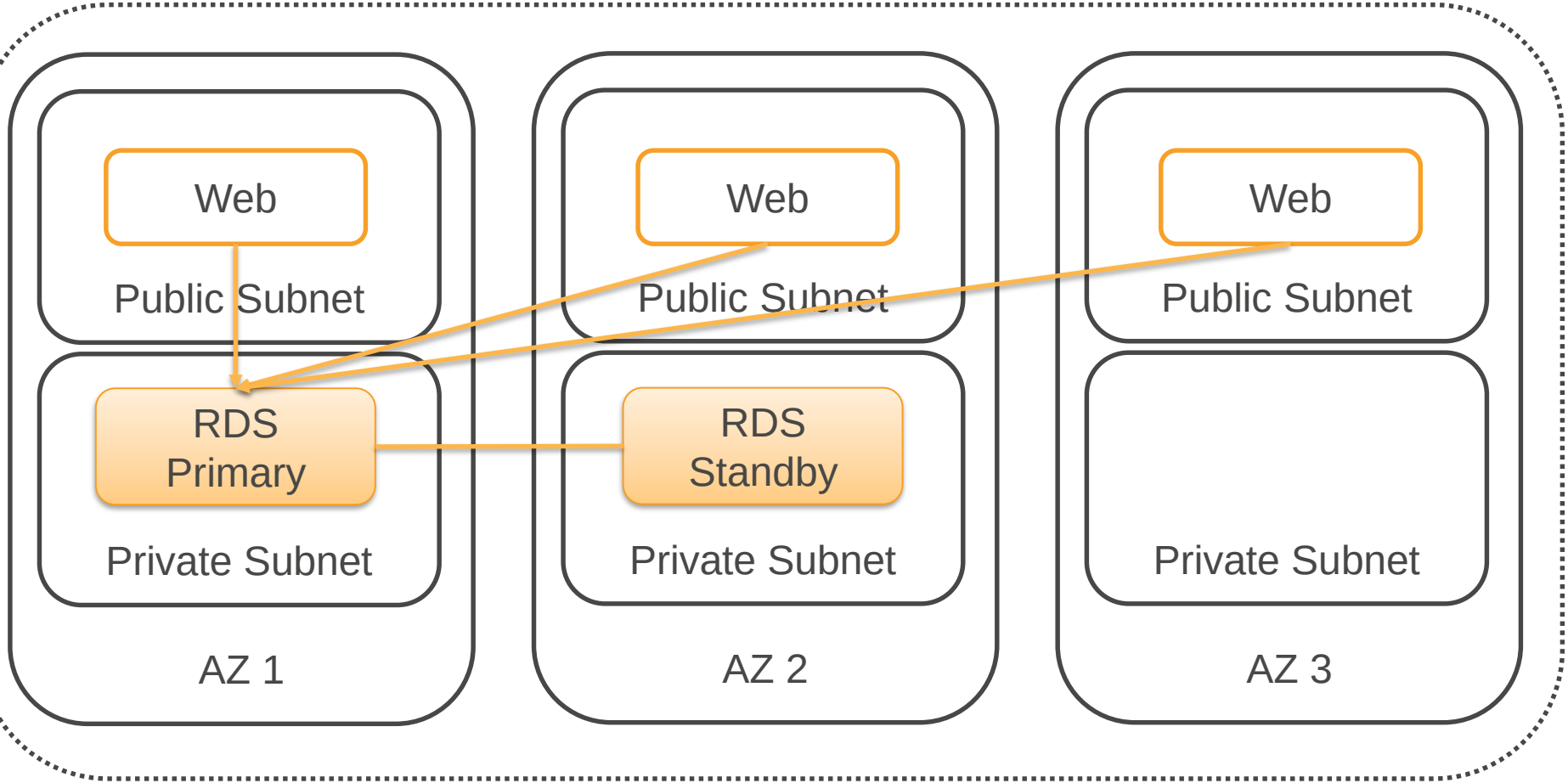
Scaling can be scheduled during next maintenance window or apply-immediate

RDS – Storage Scaling

- Storage can be scaled without interruption (zero-downtime)
- SQL Server up to 16 TB
- Aurora up to 64 TB
- MySQL, MariaDB, PostgreSQL, Oracle up to 32 TB

Scaling can be scheduled during next maintenance window or apply-immediate

RDS – Deployment



RDS – Network Security

- Deploy RDS in Private Subnet (unless your requirement is a publicly accessible RDS instance)
- Configure RDS Security Group to allow access from Web Server or Application Server Security Groups
- Assign a subnet in all Availability Zones to the DB Subnet Group
    - In case of extended AZ down or some other issue, RDS may choose to launch a replacement standby instance in a different AZ
- Connect from on-premises using Amazon DirectConnect or VPN

RDS – Permissions and Encryption

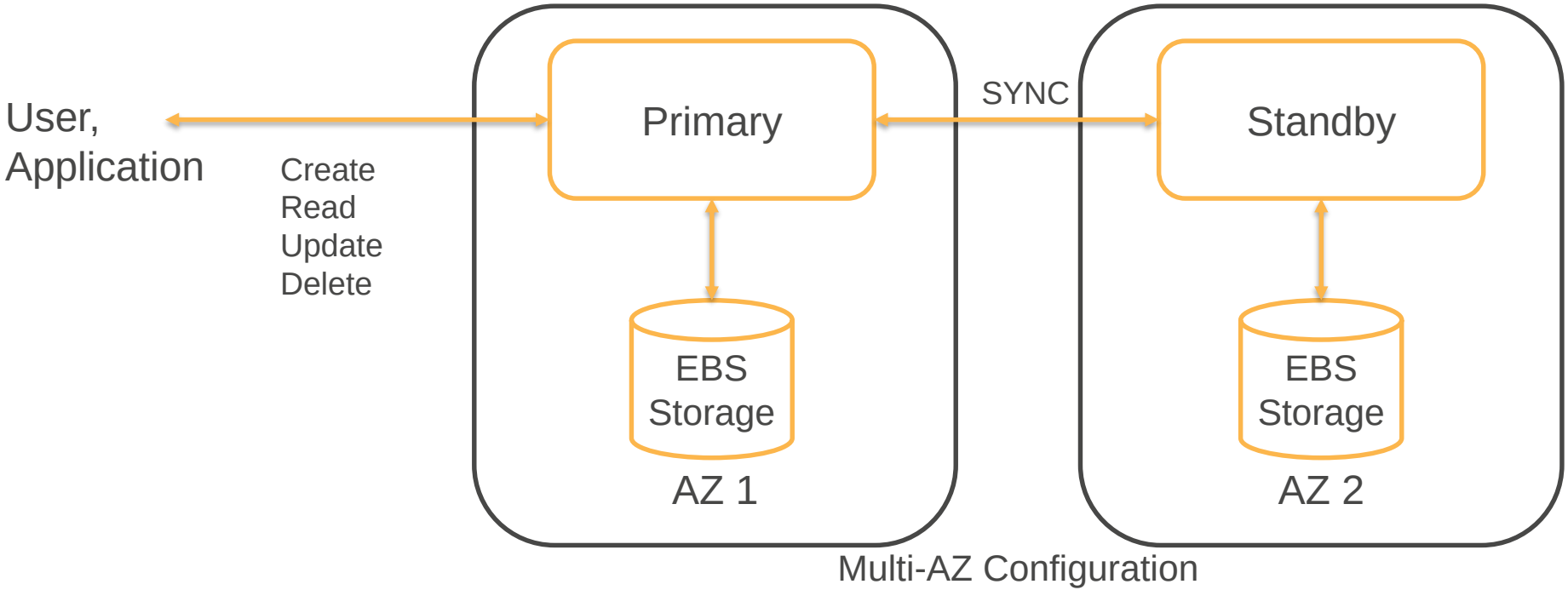
- IAM for Control Plane Access – who can create, manage, delete RDS database instances
- DB Specific User for Data Plane access – who can connect to the database, run SQL
- Optional encryption at rest using AWS Key Management Service (KMS)
- Optional encrypted connection support using SSL/TLS

RDS – Customization, Optimization

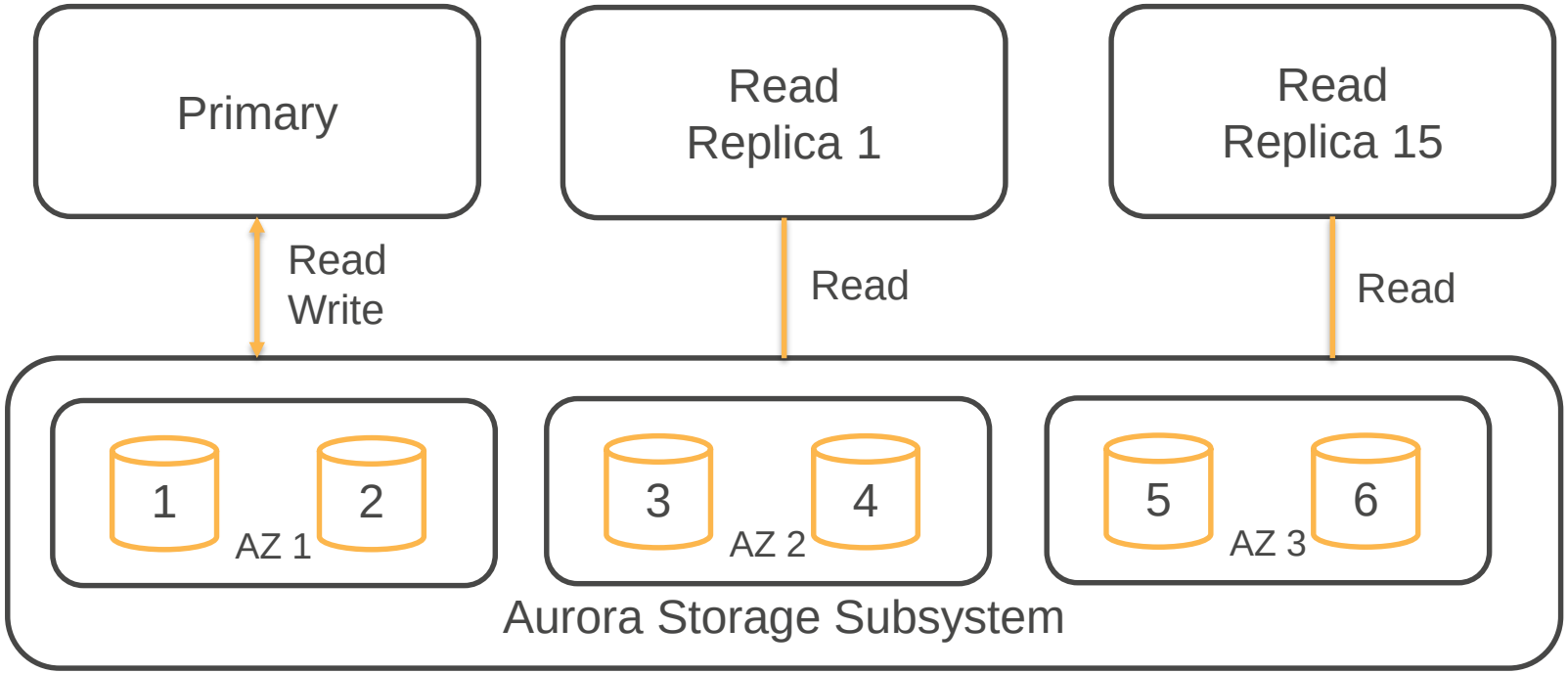
- You can customize RDS database instance and fine tune using *DB Parameter Groups*
- RDS provides best practice guidance by analyzing configuration and usage metrics
- Use Reserved Instances for long term use (1 to 3-year terms) at substantial discount
- To prevent configuration drifts, you can use AWS Config to record and audit changes to DB instance
- For monitoring, you can use CloudWatch

Amazon Aurora and Aurora Serverless

Traditional Relational Database Engine



Amazon Aurora



Aurora vs other Relational Databases

- Storage Subsystem that automatically maintains six copies of data across three availability zones
- Any changes made by Primary instance is replicated automatically
- Low latency Read Replica instances (lag time often in single digit millisecond)
- When the Primary fails,
    - A Read Replica is promoted as the new primary (typically under 60 seconds)
    - If Read Replica is not there, a new replacement primary is launched

Aurora Features

- MySQL and PostgreSQL Compatibility Modes
- Up to five times faster than standard MySQL database
- Up to three times faster than standard PostgreSQL database
- Security, Availability, Reliability of commercial databases at 1/10th cost
- Support for up to 15 low latency read replicas
- Global Database - Multi-Region Replication (fast local access, disaster recovery) for globally distributed applications

Aurora

Cluster Endpoint

- Points to Current Primary Instance
- Suitable for Writes and Reads

mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com:3306

Reader Endpoint

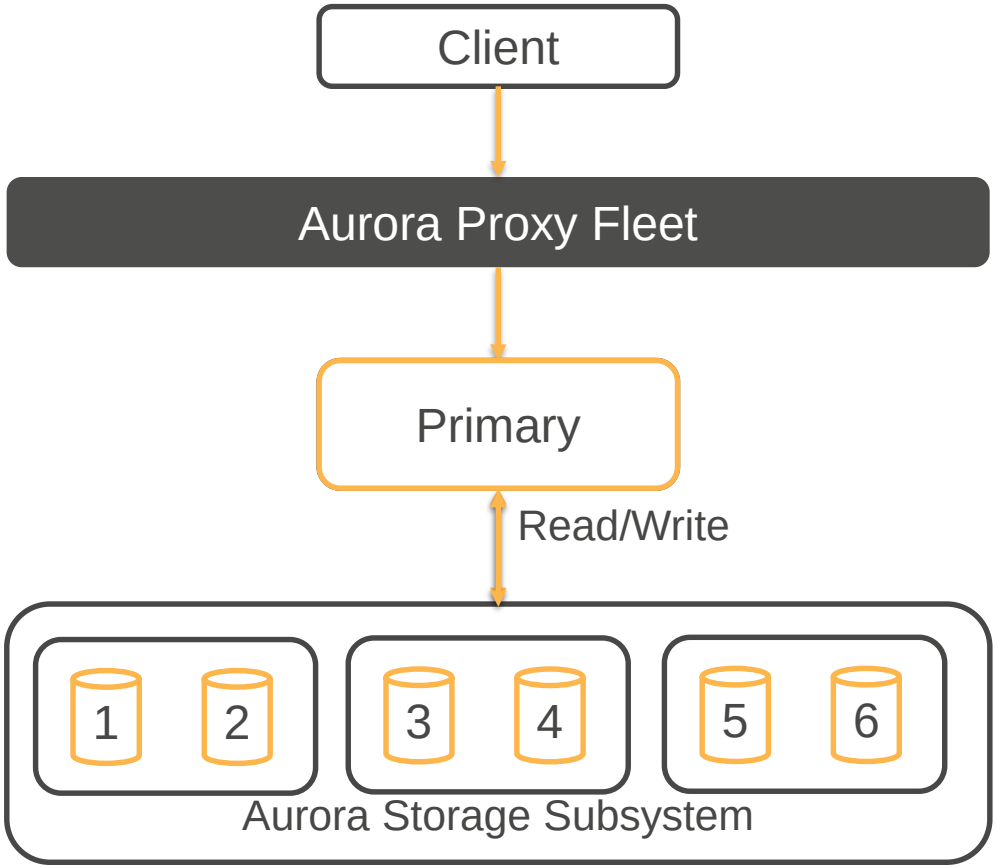
- Points to Read Replicas
- Suitable for Reads
- Multiple Read Replicas are load balanced at connection level

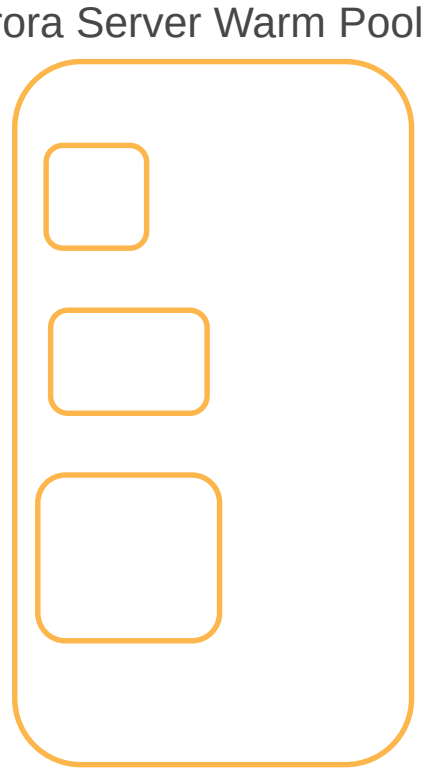
mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com:3306

Instance Endpoint

- Points to Individual Aurora Instance

Amazon Aurora Serverless





Aurora Serverless

- Storage and Processing are separate – scale down to zero processing and pay only for storage
- Automatic Pause and Resume – Configurable period of inactivity after which DB Cluster is Paused
    - Default is 5 minutes
    - When paused, you are charged only for Storage
    - Automatically Resumes when new database connections are requested

Aurora Serverless

- Aurora Serverless - Suitable for use cases that are intermittent or unpredictable
- Specify Minimum, Maximum Aurora Capacity Units (ACU)
- 1 ACU is ~2 GB of Memory with corresponding CPU/Network
- Pricing 1 ACU is $0.06 per hour + Storage + I/O
- Aurora Serverless automatically scales up and down based on load
- Scaling is rapid – uses a pool of warm resources

NoSQL Databases

DynamoDB, Cassandra, DocumentDB

DynamoDB

- Key-value NoSQL datastore
- Flexible schema - only primary key needs to be defined – all columns/attributes are flexible
- Consistent performance at any scale – single digit millisecond

Example: Movie Data

```
{
"year": 2013,
"title": "Rush",
"info": {
    "directors": ["Ron Howard"],
    "release_date": "2013-09-02T00:00:00Z",
    "rating": 8.3,
    "genres": ["Action","Biography",
               "Drama","Sport"],
    "actors": ["Daniel Bruhl","Chris Hemsworth",
               "Olivia Wilde"]
    }
}
```

Data Sample:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStarted.Python.html

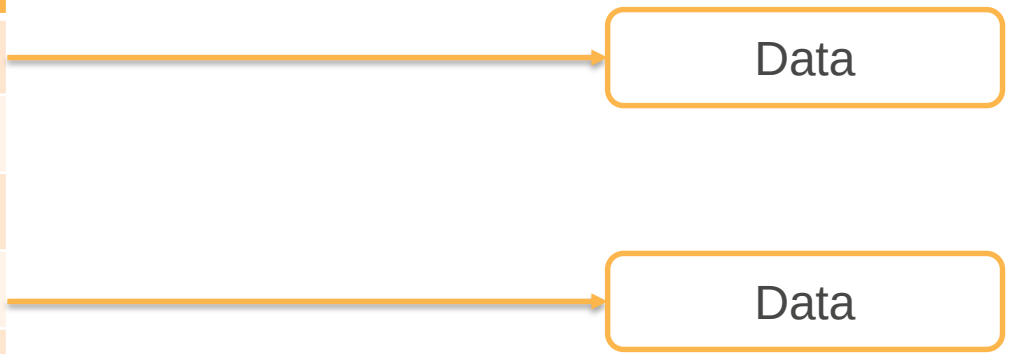
Primary Key

Simple – Single attribute

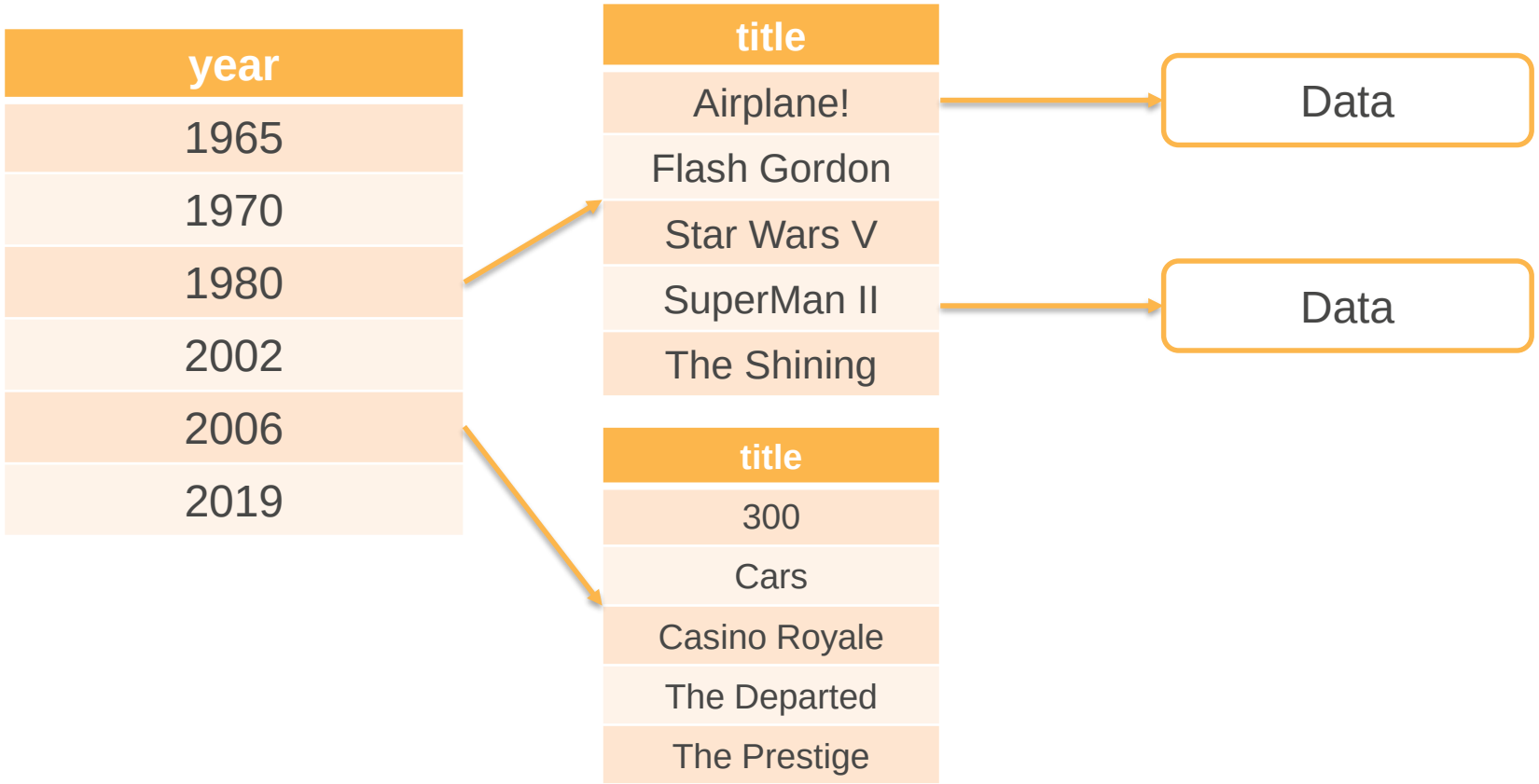
Composite – Two attributes (partition key, sort key)

Simple Primary Key - title

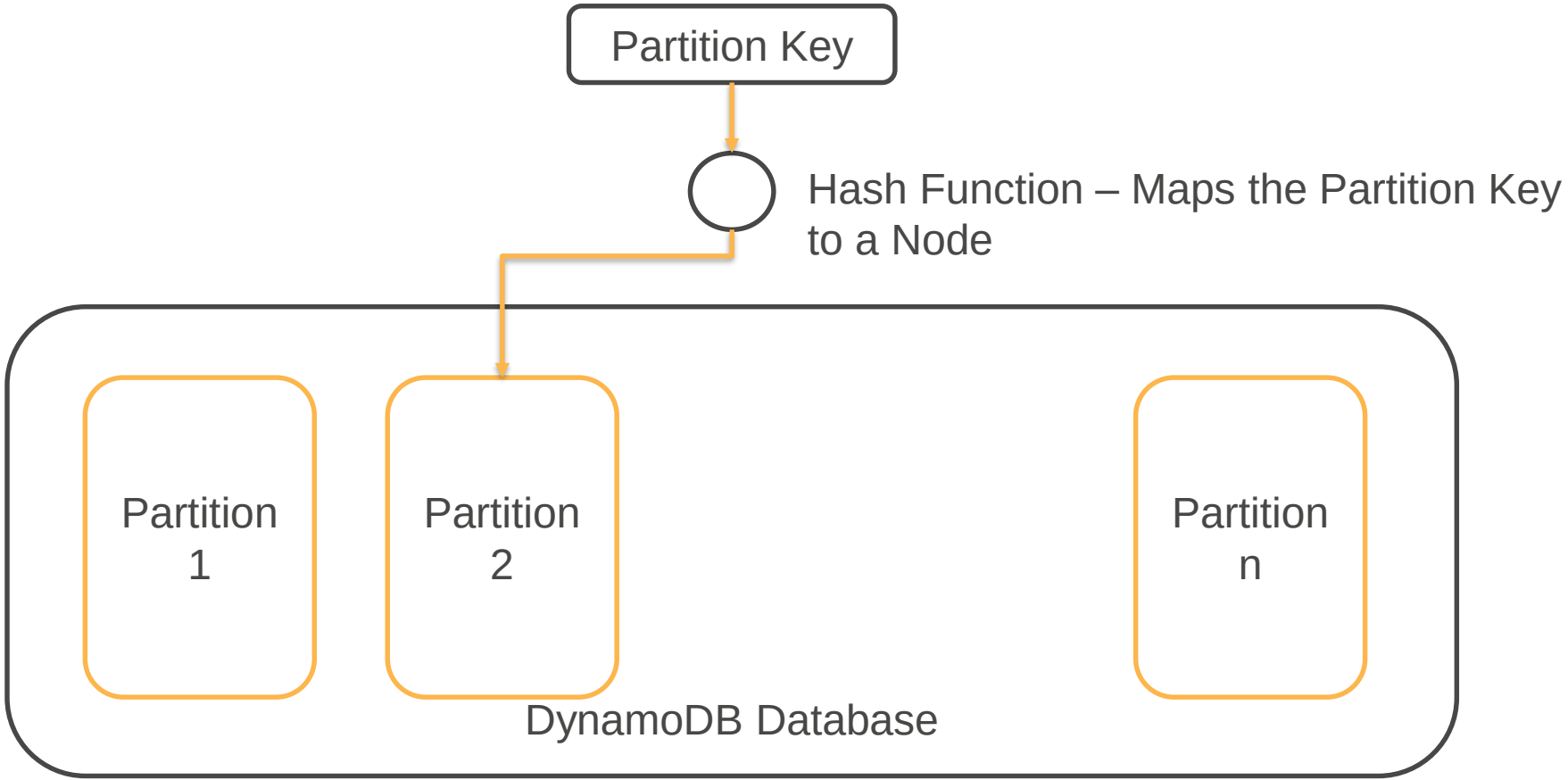
title
Avatar
Citizen Kane
JurassicPark
Titanic
Up
X-Men



Composite Primary Key – year, title



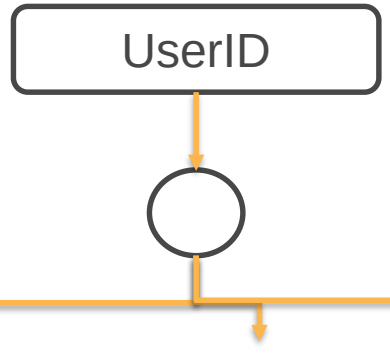
Scale-Out Processing



Game Score Table

UserID	GameTitle	Country	Other attributes
101	G1	USA	
101	G2	USA	
102	G1	USA	
102	G2	USA	
103	G1	USA	
103	G2	USA	
104	G1	USA	
104	G2	USA	

Game Score Table – UserID, GameTitle



Partition 1

UserID	Title
101	G1
101	G2
104	G1
104	G2

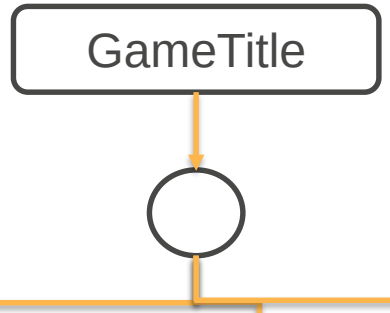
Partition 2

UserID	Title
102	G1
102	G2

Partition 3

UserID	Title
103	G1
103	G2

Game Score Table – GameTitle, UserID



Partition 1

Title	UserID
G1	101
G1	102
G1	103
G1	104

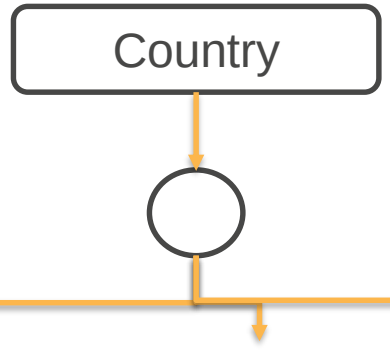
Partition 2

Title	UserID
G2	101
G2	102
G2	103
G2	104

Partition 3

Title	UserID

Game Score Table – Country, UserID



Partition 1

Country	UserID
USA	101
USA	102
USA	103
USA	104

Partition 2

Country	UserID

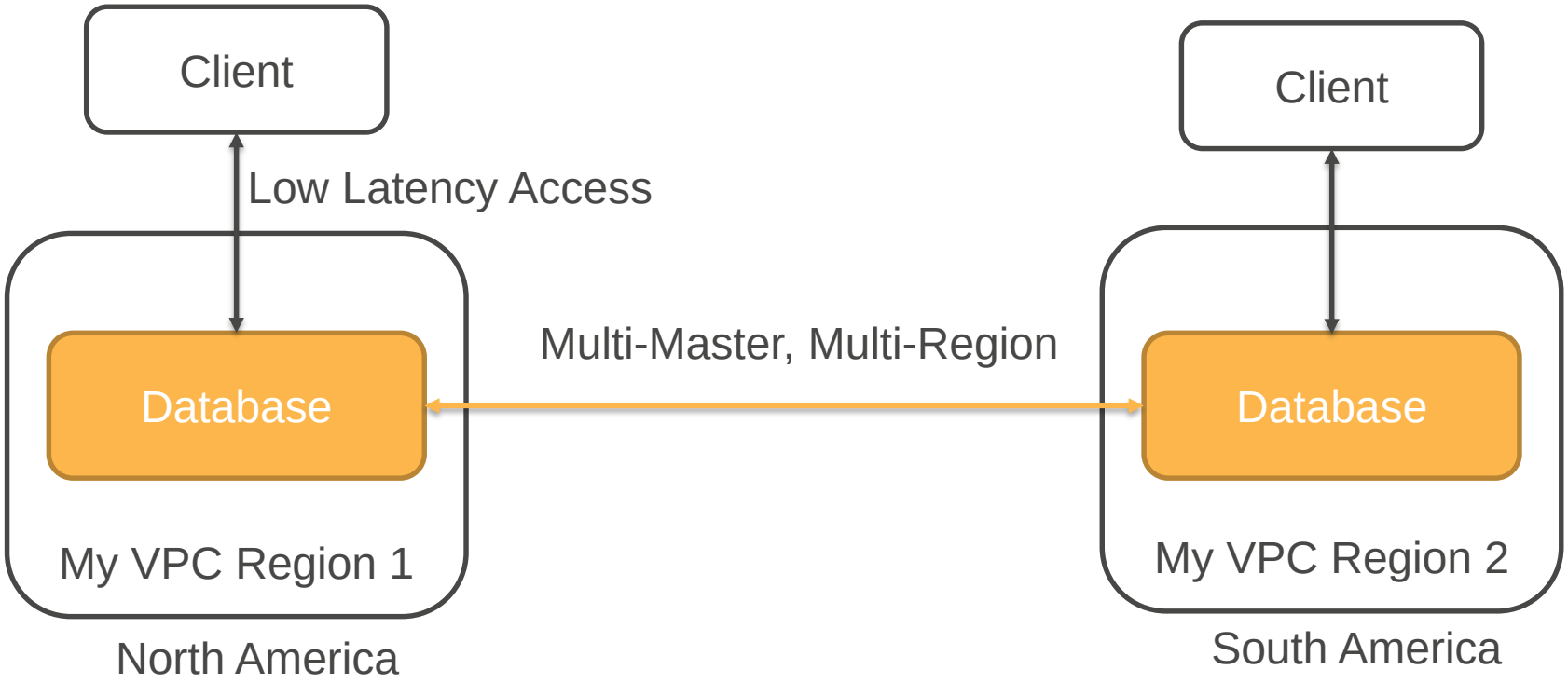
Partition 3

Country	UserID

DynamoDB Features

- Automatic replication of data across multiple-availability zones in a region
- Global Tables – multi-master, multi-region replication -Fast local access across different regions
- ACID Transaction Support
- Point-in-Time Recovery – Automated Continuous Backup (35 days retention)
- On-Demand Backup/Snapshot for long term retention
- Automatic deletion of expired items – Time To Live
- Limits - Item size cannot exceed 400 KB

DynamoDB Global Table - Multi-Region, Multi-Master



Transactions

DynamoDB supports ACID Transactions - Atomicity, Consistency, Isolation, Durability

Transactions are useful when you want to insert, delete or update multiple items as a single logical operation

"DynamoDB provides native, server-side support for transactions, simplifying the developer experience of making coordinated, all-or-nothing changes to multiple items both within and across tables"

https://aws.amazon.com/dynamodb/features/

Cassandra, DocumentDB

Amazon Managed Cassandra

AWS managed open source Apache Cassandra

Move Cassandra workloads to AWS Cloud

Performance Benefits are comparable to DynamoDB

AWS recommends Cassandra for: industrial equipment data collection, and other use cases that require high performance and large number of columns

Cassandra versus DynamoDB

- DynamoDB primary key is made up of single attribute partition key and optional single attribute sort key. Cassandra supports multi-column partition and sort keys
- DynamoDB max item size is 400KB – Cassandra has a theoretical limit of 2GB per column. However, general practice is not to exceed few MBs.
- Cassandra also supports large number of columns – DynamoDB even though supports large number of attributes, it is constrained by 400KB size limit per item

Amazon DocumentDB

“Amazon DocumentDB (with MongoDB compatibility) is a fast, scalable, highly available, and fully managed document database service that supports MongoDB workloads.”

DocumentDB emulates MongoDB API and it is not true port of open source code. Currently, there is a drift in the direction of MongoDB and DocumentDB.

ElastiCache

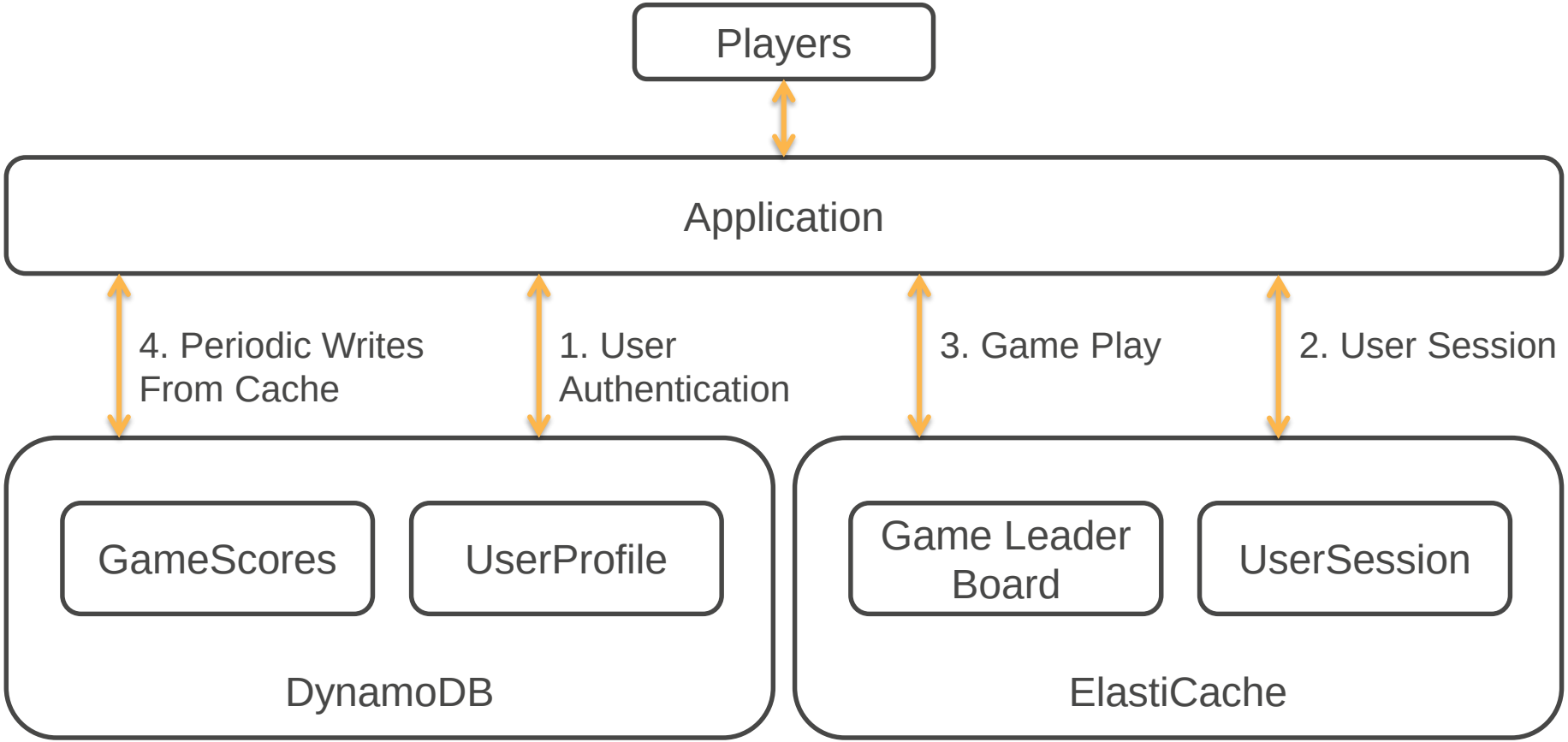
In-memory data store

Amazon ElastiCache

- In-memory datastore with sub-millisecond latency
- Ideal for frequently read data, reduce read-traffic going to database, buffer high-frequency writes and periodically reconcile with backend database

  Uses: Product reviews and rating, Caching, Session Management, Gaming leaderboards, geospatial applications
- Deploy in your VPC – Network isolation and security
- Choice of engines: Memcached, Redis

Game Leader Boards (READs/WRITEs)



MemCached Features

- Key-value store
- Scales up to 20 nodes and 12.7 TB
- Sub milli-second latency

Redis Features

- In-memory datastore with advanced data structures: Strings, Lists, Sorted Sets, Hash, Bit Arrays

  Sorted Sets can be used to easily Game Leader Boards – keep a list of players sorted by rank.
- Built-in commands for Geospatial data
  - Distance between two places or persons
  - Find all places within a given distance from a point
- Sub milli-second latency
- Scales up to 250 nodes and 170 TB

Redis High Availability Features

- Pub-Sub and Messaging

  For example, High performance chat rooms, server to server communication, social media feeds
- Read Replica across multiple Availability Zones
- Detects primary node failure and automatically promotes replica as primary
- Backup, Restore
- Export to another region
- Lua scripting support

Amazon Redshift

Data Warehouse - Redshift

- Peta Byte Scale Massively Parallel Relational Database
- Cluster consists of Leader Node and Multiple Compute Nodes

  Available Storage = Storage per Compute Node X Number of Compute nodes
- Columnar Storage
- Targeted Data Compression
- Powerful SQL based Analytics
- With Redshift Spectrum - query can span tables in Redshift and files stored in S3 Data Lake