AD-Fraud Bot Traffic Detection

Goal - Increase the bot traffic detection by 5%

- Rajeev Joshi

The problem

Big Picture

Ad fraud When a fraudster tries to deceive the advertisers with fake ads/traffic.

Impressions is a collective measure of how many times an ad is displayed regardless of whether it was viewed

Context

Impression fraud is when an ad is not viewable to the human eye, but still counted(Advertiser pays money for this).

Impression fraud

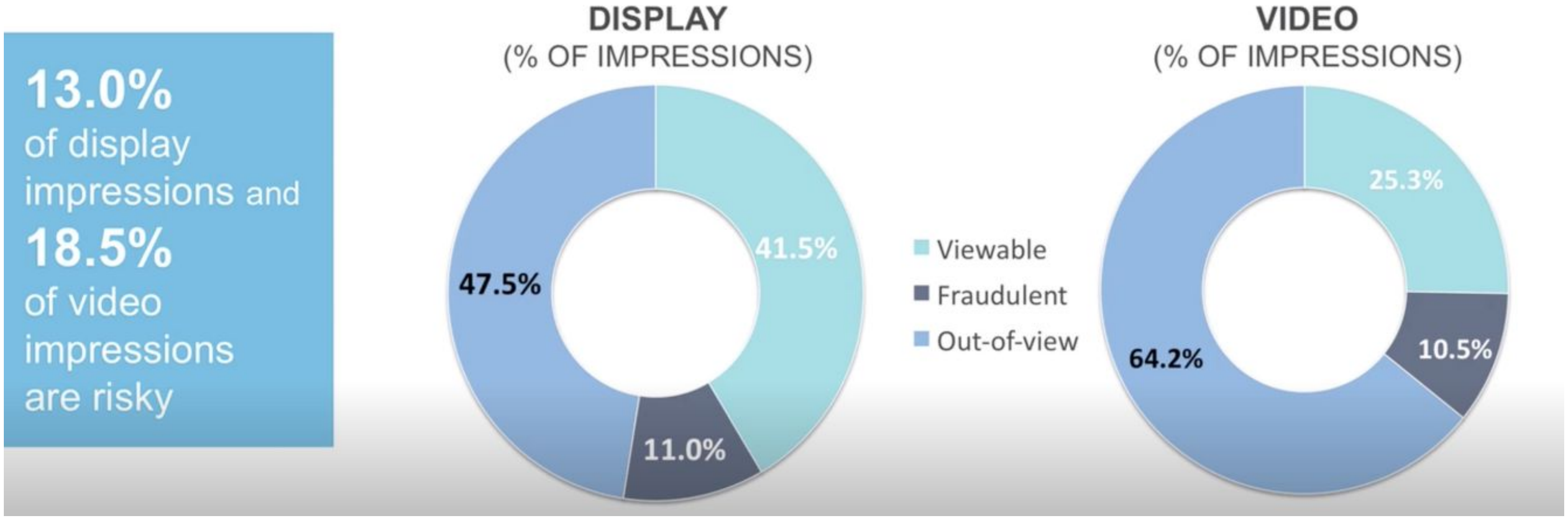
- Pixel Stuffing
- Ad Stacking
- Fake Websites

Problem statement

Huge boom in Bot traffic due to programmatic advertising

Fake Ad-traffic generated by bots burning through the advertisers budgets and eroding trust in **Digital advertising**

Sizing up the RTB challenge

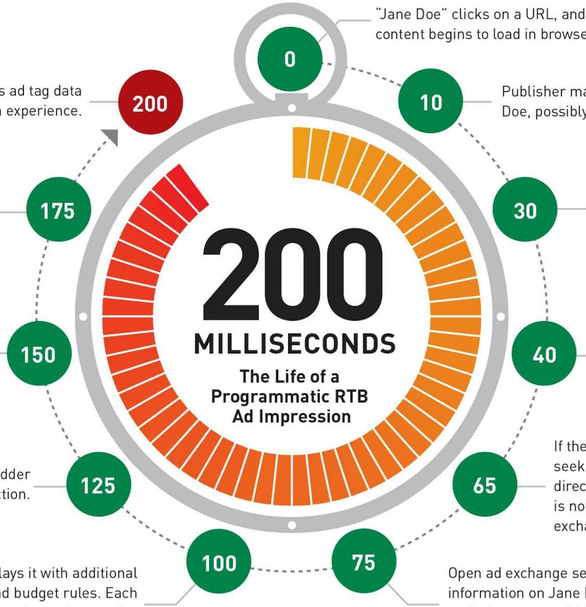


13.0% of display impressions and **18.5%** of video impressions are risky

Source: Trade Desk

MediaCrossing™

200MS: The Life of a Programmatic RTB Ad Impression



0 — "Jane Doe" clicks on a URL, and the publisher's content begins to load in browser.

10 — Publisher may find information it has stored on Jane Doe, possibly in its Data Management Platform (DMP).

30 — Publisher sends available information to its ad server asking ad server whether an ad campaign is available that would target Jane Doe. If there is a campaign matching Jane Doe's profile, an ad is served.

40 — If no campaign targets Jane Doe, the server seeks to match the impression programmatically requesting response from selected traders, ad networks and supply-side platforms (SSPs).

65 — If the impression is not cleared, the server may seek to clear the impression in a programmatic direct way via private exchanges. If the impression is not cleared, the request is sent to an open ad exchange in hopes of achieving liquidity.

75 — Open ad exchange sends a bid request containing information on Jane Doe's browser, website URL, and ad type to multiple bidders including traders, ad networks and demand-side platforms (DSPs).

100 — Each bidder processes bid request, overlays it with additional user data and marketers' targeting and budget rules. Each bidder's algorithm evaluates the request, selects the creative and sends it along with optimal bid price to ad exchange.

125 — Ad exchange selects winning bid from bidder responses through second-price auction.

150 — Ad exchange sends winning ad URL and price from winning bid to Publisher's ad server. Publisher's ad server tells Jane Doe's browser which ad to display.

175 — Jane Doe's browser pulls ad from winning bidder's ad server and sends matching ad to browser. Browser displays web page including matching ad.

200 — Winning bidder's ad server receives ad tag data on Jane Doe's initial interaction experience.

Challenges deep-dive

Challenge 1

Time Constraints

Ad exchanges often require a response within about 120 milliseconds after the bid request is sent. Timing out too often might affect the DSP's ability to bid

Challenge 2

Low Memory Footprint

A typical large scale DSP

(Trade Desk), will evaluate up to 3 Million bid requests per second

Challenge 3

Sophisticated invalid traffic (SIVT)

- IP rotation

Hard to capture from IP blocking

- Bot farms are constantly evolving

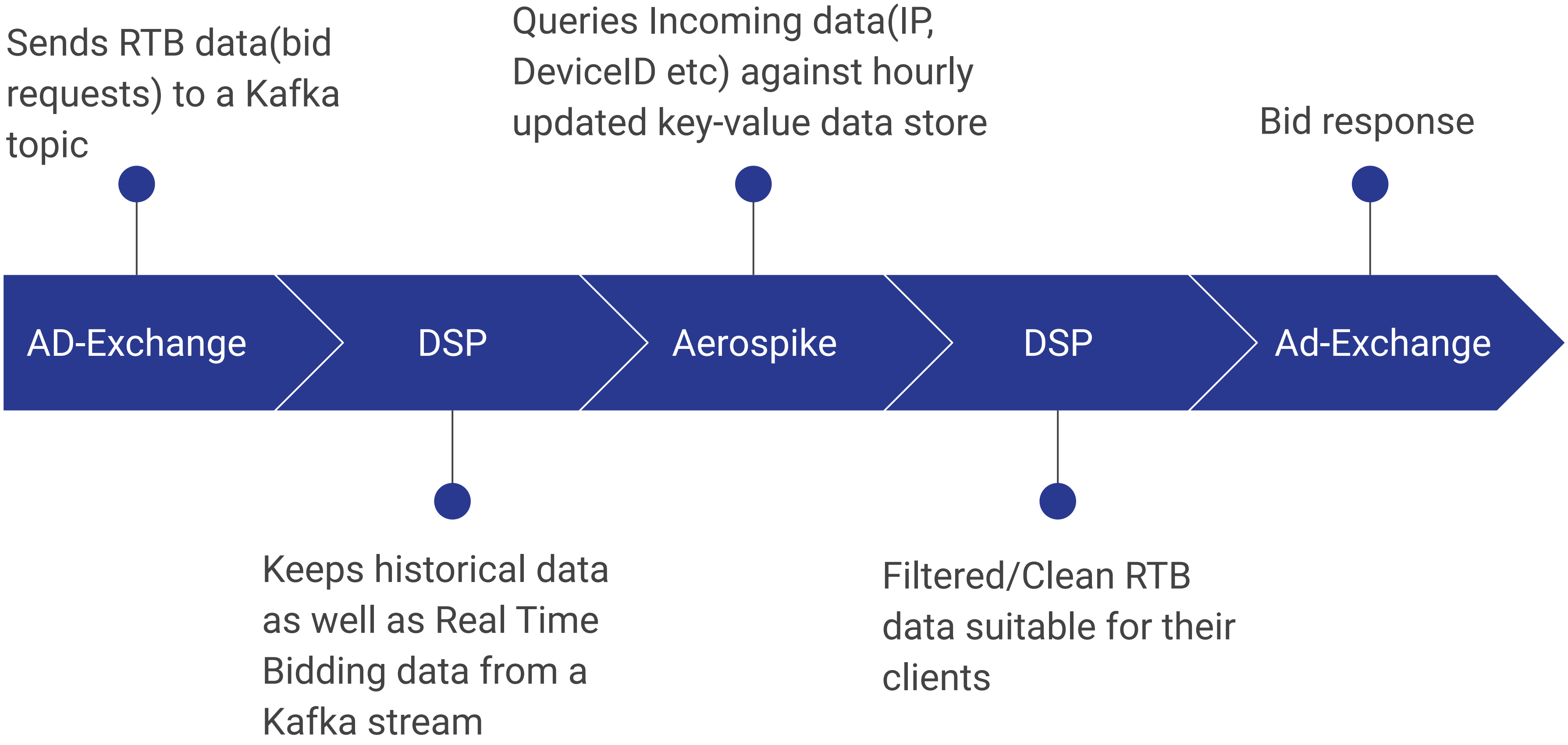
Solution

Millisecond Key-Value Lookups (AeroSpike) ?

Using Memory efficient data formats

Updating the Filtering Table as frequently as possible

Implementation



AD-Exchange

Sends RTB data(bid requests) to a Kafka topic

DSP

Keeps historical data as well as Real Time Bidding data from a Kafka stream

Aerospike

Queries Incoming data(IP, DeviceID etc) against hourly updated key-value data store

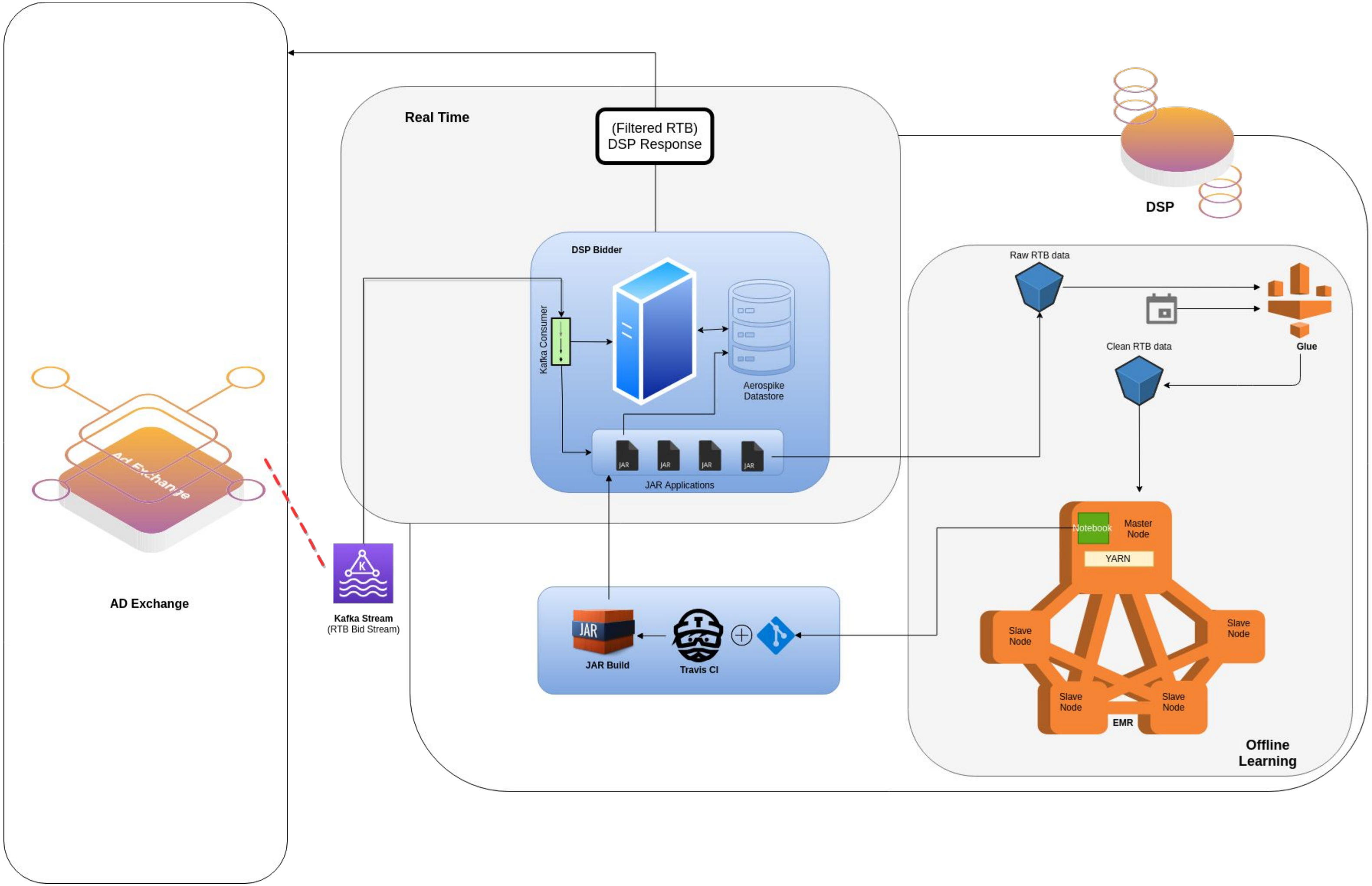
DSP

Filtered/Clean RTB data suitable for their clients

Ad-Exchange

Bid response

Architecture



Real Time

(Filtered RTB)
DSP Response

DSP

DSP Bidder

Kafka Consumer

Aerospike
Datastore

JAR Applications

AD Exchange

Kafka Stream
(RTB Bid Stream)

JAR Build

Travis CI

Raw RTB data

Glue

Clean RTB data

Notebook

Master
Node

YARN

Slave
Node

Slave
Node

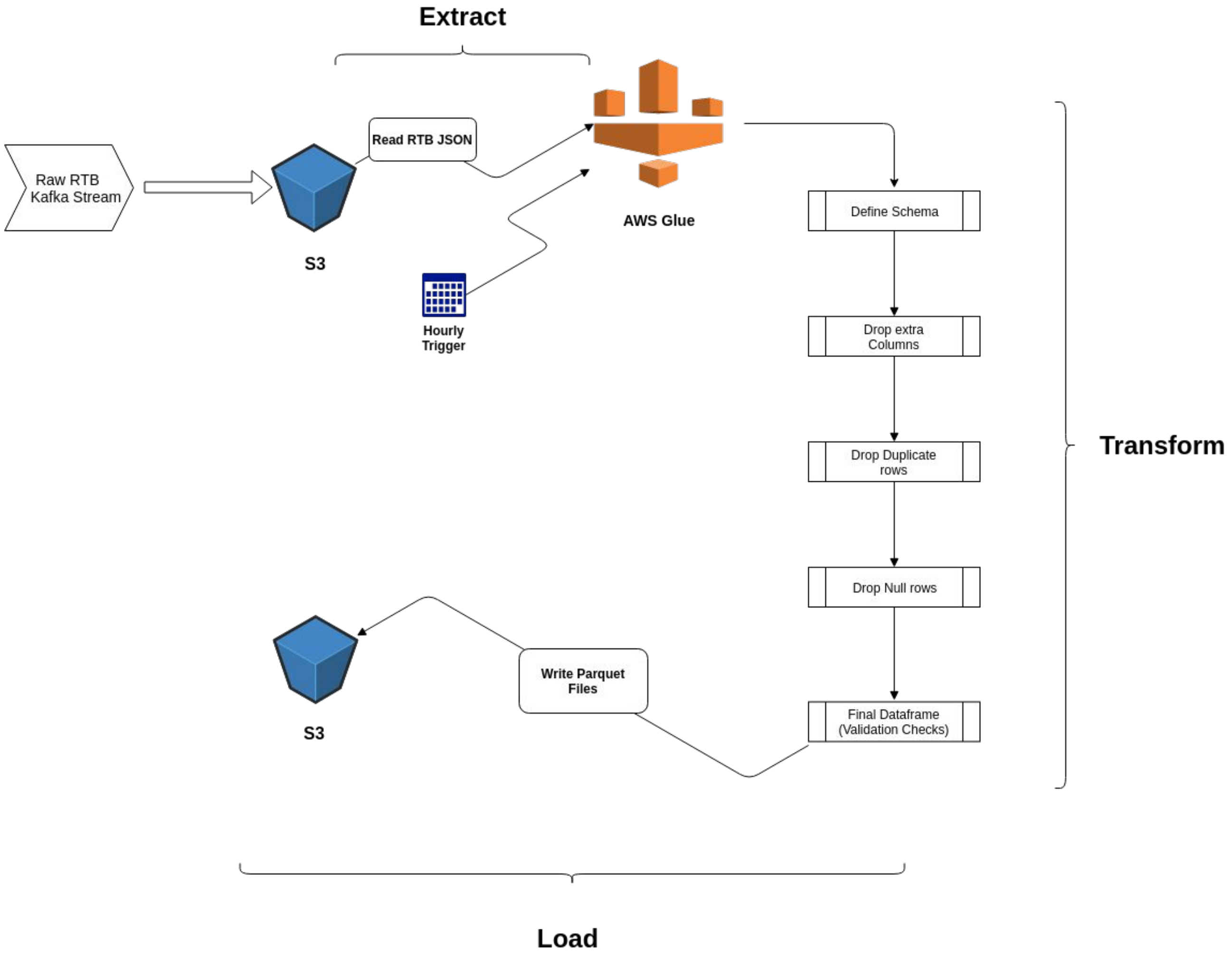
Slave
Node

Slave
Node

EMR

Offline
Learning

ETL Pipeline



Extract

Raw RTB Kafka Stream

S3

Read RTB JSON

Hourly Trigger

AWS Glue

Define Schema

Drop extra Columns

Drop Duplicate rows

Drop Null rows

Final Dataframe (Validation Checks)

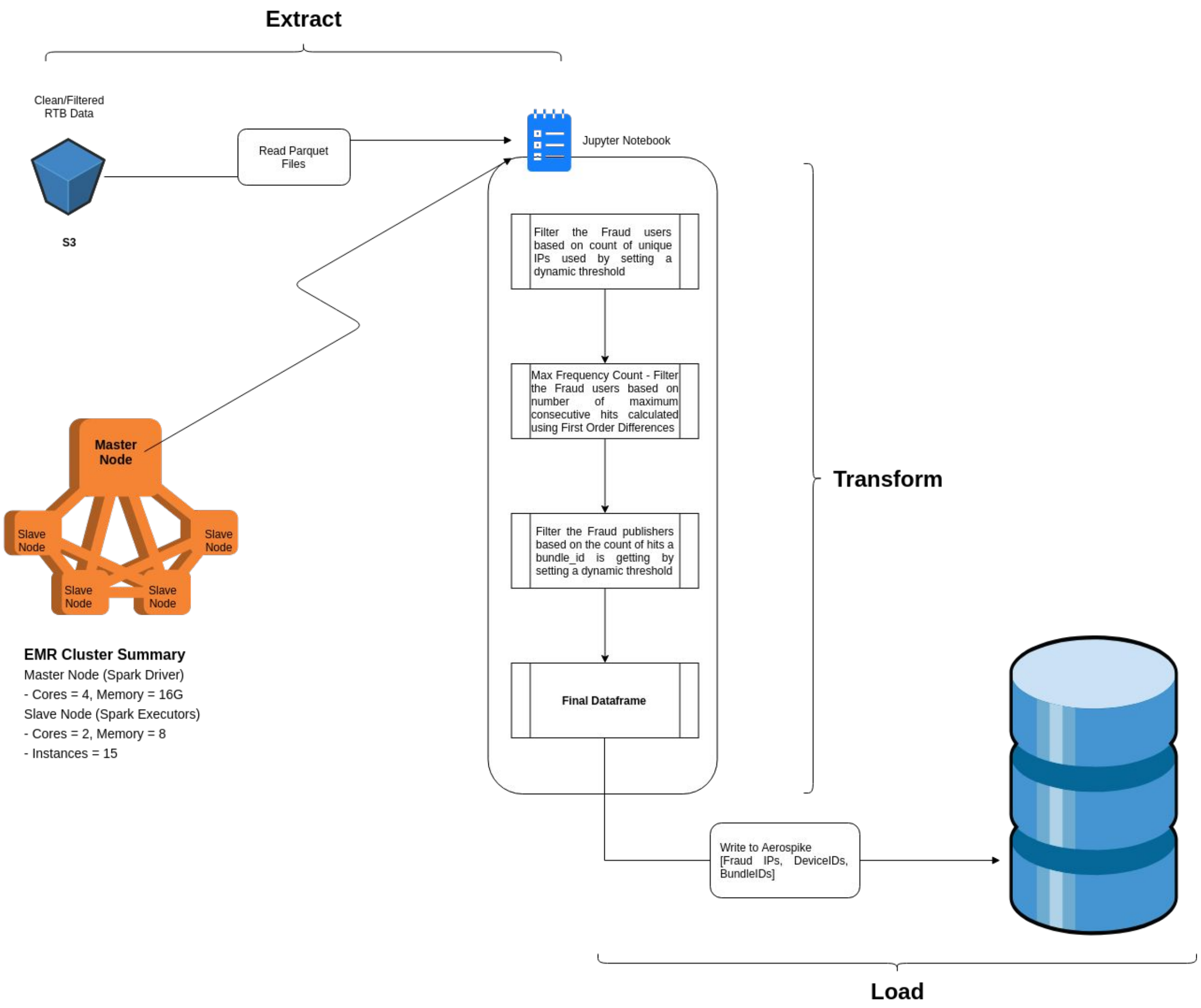
Transform

Write Parquet Files

S3

Load

Ad-Fraud ETL Pipeline



Extract

Clean/Filtered RTB Data

S3

Read Parquet Files

Jupyter Notebook

Filter the Fraud users based on count of unique IPs used by setting a dynamic threshold

Max Frequency Count - Filter the Fraud users based on number of maximum consecutive hits calculated using First Order Differences

Filter the Fraud publishers based on the count of hits a bundle_id is getting by setting a dynamic threshold

Final Dataframe

Transform

Write to Aerospike [Fraud IPs, DeviceIDs, BundleIDs]

Load

EMR Cluster Summary

Master Node (Spark Driver)

- Cores = 4, Memory = 16G

Slave Node (Spark Executors)

- Cores = 2, Memory = 8
- Instances = 15

Impact

Expected to have 5% SIVT capture rate increase