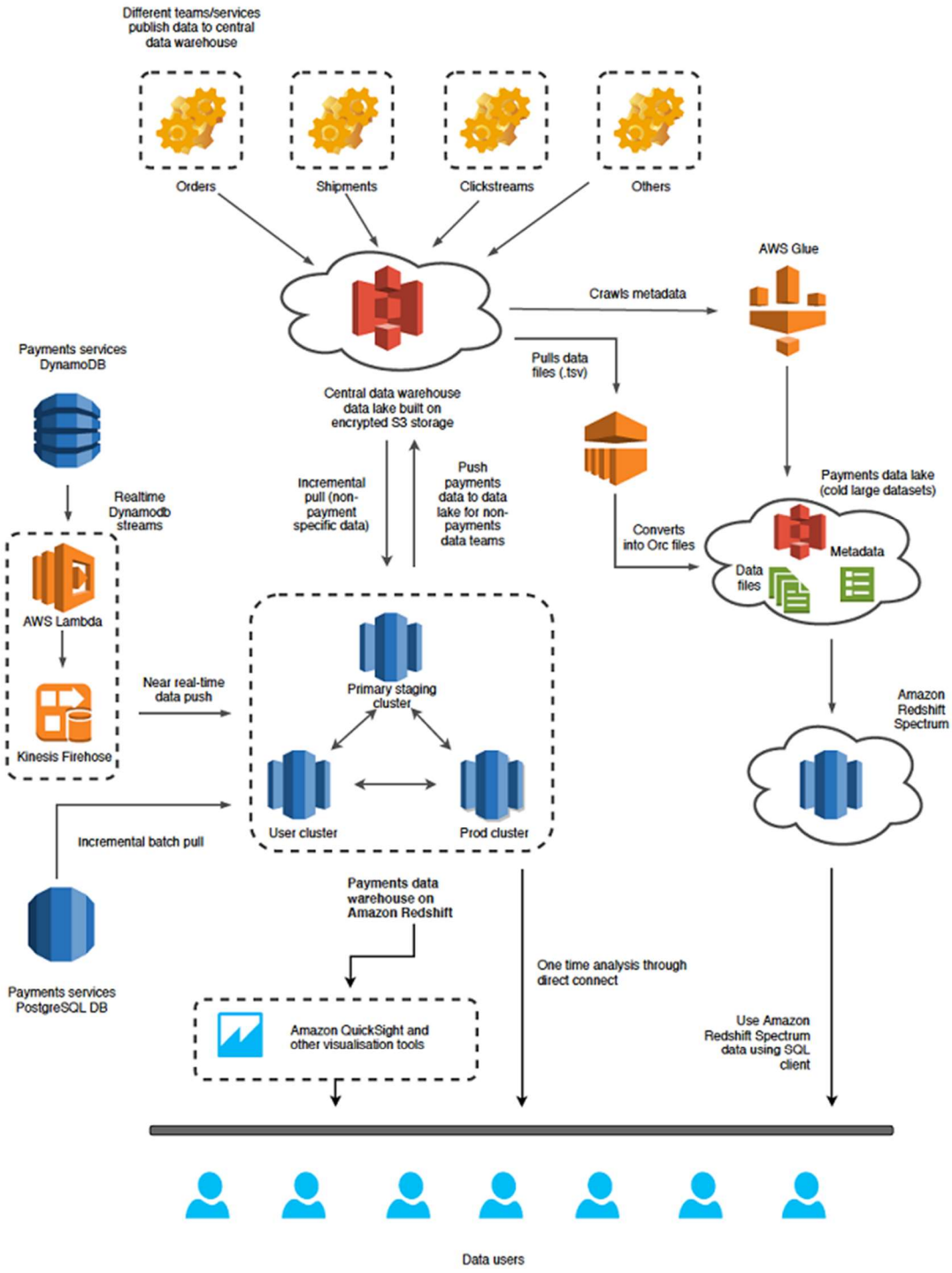
Design a cloud datare platform to process and deliver insights based on

Each store manages its own inventory of raw materials. Each store prepares pizzas, side dishes, etc. and sells them along with ready to eat products such as cookies, drinks, etc. All these different services connect to the central data warehouse which is build on encrypted S3 storage

The sale can happen by Point of Sale (POS) or Online. The online transactions would be flowing in real time whereas the transactions made by POS can be synced every 15 minutes in batches. Every time the new transaction data come to the DynamoDB AWSLambda triggers and produces to kinesis and pushes the real time data to cluster.


Different teams/services publish data to central data warehouse
Orders
Shipments
Clickstreams
Others
AWS Glue
Crawls metadata
Pulls data files (.tsv)
Payments services DynamoDB
Central data warehouse data lake built on encrypted S3 storage
Incremental pull (non-payment specific data)
Push payments data to data lake for non-payments data teams
Payments data lake (cold large datasets)
Converts into Orc files
Metadata
Data files
Realtime Dynamodb streams
AWS Lambda
Kinesis Firehose
Near real-time data push
Primary staging cluster
User cluster
Prod cluster
Amazon Redshift Spectrum
Incremental batch pull
Payments data warehouse on Amazon Redshift
Payments services PostgreSQL DB
Amazon QuickSight and other visualisation tools
One time analysis through direct connect
Use Amazon Redshift Spectrum data using SQL client
Data users

Amazon Redshift clusters: Instead of the over-provisioned, monolithic model of the data-warehousing era, we now employ three Amazon Redshift clusters. Each cluster has a separate purpose and can be scaled independently. This model opens the door to additional cost savings through dynamic cluster size management.

    - Staging cluster:
        - Dynamic data sources are in a transition state, moving from relational to non-relational data sources.
        - The mechanism to pull data from the central data lake to Amazon Redshift storage also continues to evolve and remains resource intensive.
        - The Payment Analytics team naturally leans heavily on the payment-oriented datasets. Some transformational needs are unique to the team's mission. This specialization leads the team to perform additional core dataset processing, particularly for data science and in-depth business analysis tasks.
    - User cluster: Internal Business users wanted to create the tables in the public schema for their analysis. They also needed direct access for their ad hoc analysis. Although the SQL proficiency is high among the Payment Analytics users, we apply workload management (WLM) and query monitoring rules (QMR) in a unique way on this cluster, to allow broad but reasonable use of the system.
    - Data-platform cluster: We execute transformations for datasets with a critical SLA on this cluster, loading the result of this work into both the User and Prod clusters.
2. Near real-time data ingestion: Few reports need real-time data collection from different services. Many of these services store the data in DynamoDB, with DynamoDB Streams enabled. We consume the data from these streams through an AWS Lambda function and Amazon Kinesis Data Firehose. Kinesis Data Firehose delivers the data to Amazon S3 and submits the copy command to load the data into Amazon Redshift. The Payment Analytics customers can consume this data as it's loaded in 15-minute batches throughout the day.
3. Alternate compute on Amazon EMR: We receive website clicks data through clickstream feeds, which can run into billions of records per day for each marketplace. Although large datasets are critical, costumers access

them less frequently on Amazon Redshift. We chose Amazon S3 as a storage option and applied the transformations using Amazon EMR. This approach ensures that we do not fill up the database with massive cold data. At the same time, we enable data access on Amazon S3 using Amazon Redshift Spectrum, which provides similar query performance. As part of the Payment Analytics processing, the team converts native TSV data from Amazon EMR into either ORC or Parquet file formats. These columnar formats allow for faster and more efficient processing, especially when only a subset of the columns is required. To further improve performance, the column-oriented data layout on Amazon S3 renders with daily partitions. Amazon Redshift can then automatically pick which files need to be read, saving time and expense.

Requirements

1. Handle large write volume: Billions of write events per day.
2. Handle large read/query volume: Millions of merchants wish to gain insight into their business. Read/Query patterns are time-series related metrics.
3. Provide metrics to customers with at most one hour delay.
4. Run with minimum downtime.
5. Have the ability to reprocess historical data in case of bugs in the processing logic.

So I have designed a cloud data platform to process and deliver insights as per requirements