CODING CHALLENGE

DATA INTEGRATION

Consumer engagement & analytics

#Digital IT | Consumer engagement, 2019

DATA INTEGRATION PRODUCT VISION

ENSURE EXISTENCE, QUALITY AND COMPLIANCE OF RELEVANT CONSUMER DATA, AS A FOUNDATION FOR OUR CONSUMER FOCUS; PREREQUISITE FOR PERSONALISATION AND FOR MARKETING INTELLIGENCE.

HIGH LEVEL OVERVIEW

The diagram right shows an eCommerce mobile application. In this, an embedded SDK is emitting stream of json format tagging messages (1) containing behavioural information (eg product viewed by a consumer logged into) as consumers use the app.

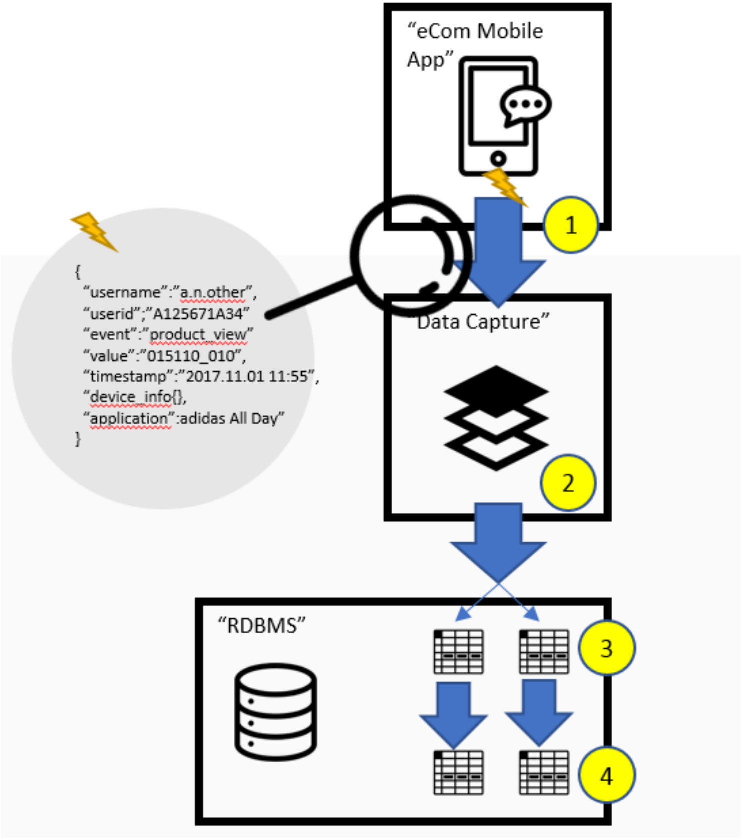
The eCommerce company owning the app would like to:

2) Reliably capture the incoming message stream where there could be large fluctuations in the number of messages being received at a given time (eg sales on Cyber Monday)

2) Queue the data for further processing

3) Parse the data to tabular structures

4) Forward the data to tables in a data warehouse for analysis and reporting teams



```
{
 “username”:”a.n.other”,
 “userid”;”A125671A34”
 “event”:”product_view”
 “value”:”015110_010”,
 “timestamp”:”2017.11.01 11:55”,
 “device_info{},
 “application”:adidas All Day”
}
```

CHALLENGE #1: THE BASICS

After the JSON message is captured, it is stored in staging XML documents. We ask you to:

- Define the structure of the XML document.
- Create a batch process using ETL tool (PDI community edition) to parse the data and store it in tabular CSV files in a normalized form.
- You need to source and destination data files and the PDI package.
- The ETL job must handle failure rolling back the changes and send a success/failure notification to an email address.
- A second ETL must move normalized data to some datawarehouse csv files using a schema optimized for reporting.

Bonus points:

- You are free to add more attributes to the JSON source to build a richer table structure.
- ETL could process correct records and derive incorrect ones to an error file

CHALLENGE #2: BONUS POINTS

You will replace the batch based architecture to go real time in data capture. For that you need to:

- Implement an API capable of recieving JSON objects and send those objets to three different queues. One to parse data into tabular data load it in a DB engine, another one to store the same data in csv format, and a third one to log messages in text files. All of them should be able to respond to an eventual big message load.
- Incorrect values should be handled sepparately to go to an error file.

WHAT WE EXPECT FROM YOU

- Develop this application in an 8 hour time
- Provide ETL packages, code used, DB structures and a README explaining how to run/build/use it.
- Name every framework/library/tool you use in your README
- BONUS: Create 1 slide with a CI/CD pipeline proposal for the app.
- When you are done, check in your soltion into any public GIT repo hoster (github, bitbucket, etc) and send us the link and any other documentation you want by email.