NUS-ISS

MASTER OF TECHNOLOGY IN SOFTWARE ENGINEERING

Graduate Certificate Examination

Sample Case Study & Question Paper

Subject: ***Engineering Big Data***

Froggy FinTech Case Study

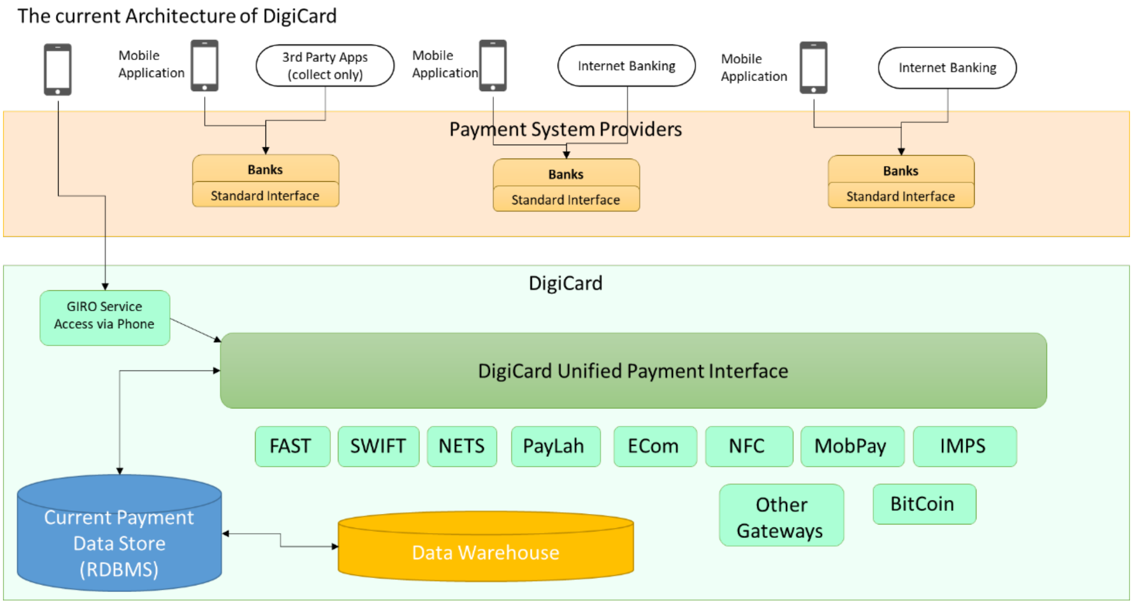
Froggy is a fast growing FinTech company which commenced operations about 8 years ago. FinTech refers to a set of technologies that focus on new ways of delivering banking and financial services to consumers.

The company has gained a world-wide customer base with their largest base (about 40%) of the customers in the USA and the reminder of their customers are spread across Asia, Europe and Pacific. They have very less presence in the African and South American nations.

Their DigiCard is a fully integrated, digital financial commerce platform. DigiCard offers worldwide services for innovative digital payments such as online, mobile, and at the POS. DigiCard provides a comprehensive payment solution that covers a wide spectrum of payment processing channels including interbank transfers, electronic funds transfers such as NETS, internet based payments, debit/credit cards, bitcoins, NFC payments, mobile payments etc. Such payment transactions expose *Froggy* to various kinds of frauds causing financial loss for the organization. Most of these frauds happen during transactions involving internet based online payment.

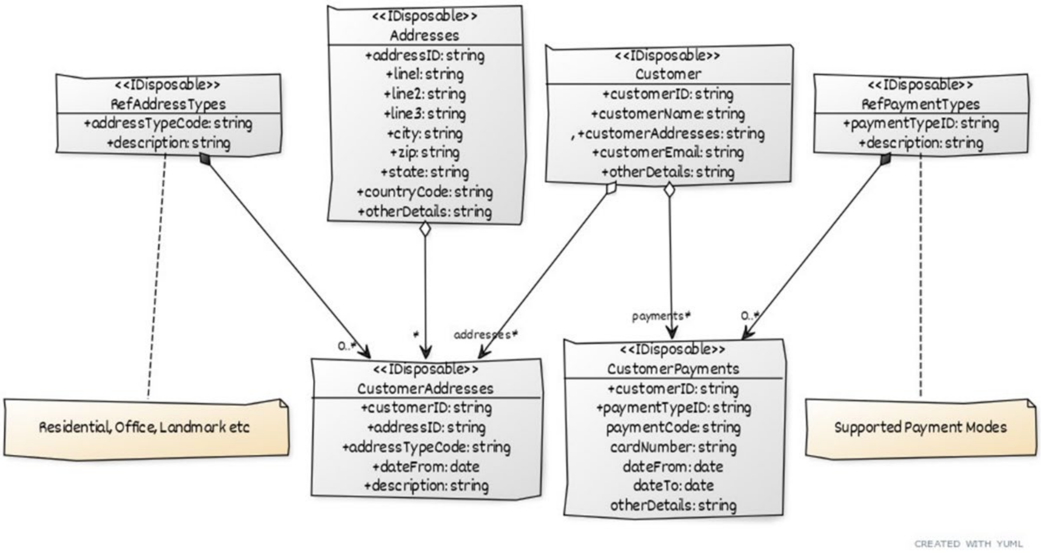
Current Architecture

DigiCard has very well designed IT systems that were predominantly built on an architecture that focussed on transactional processing. The system acts as a payment integration technology for many online ecommerce merchants. Main functionalities include multiple payment processing using web and mobile applications. The architecture is processing transactional structured. For their decision making needs and data insights they have traditional data warehouse systems which they use for their analytics needs. The architecture is depicted below:



Sample Schema in the Current Data Store

To showcase one example of how data stores are architected and designed the storage of Customer Data is described in this section. In DigiCard Database Schema, customer could have a Billing Address, a Residence Address and more than one Delivery Address. Therefore, the current system stores Addresses in a separate normalised Table, instead of repeating Address details many times in Customer Tables. This is typically transaction focussed and allows validation of the Address against files of recognised Addresses. In addition, maintaining a separate Address Table helps in tracking changes of Customer Addresses. Here is a number of Payment Methods, which are stored as Reference Data. A Customer can have many Payment Methods, such as Credit Cards, NFC, Bit Coin etc. The current system has been designed to resolve the many-to-many Customer Payment Relationship with an intermediate table CustomerPayments. The sample schema with the above focus is provided in the diagram below.



The current IT infrastructure and data store architecture has been meeting its needs over the years. However, with the growth in business, the intense competitions in the market and increased customer expectations, the company has a felt need to review its IT systems with particular focus on enlarging their data collection, repository and big data analytics. They had carried out an initial analysis and have arrived at some key findings.

The major limitations in the current data store are:

(i) Much of structured data need to be extracted into features ready for analytics modelling.

(ii) For each type of payment transactions there are many kinds of fraud where each fraud exhibits certain behaviour. Currently there is no study performed to evolve patterns or grouping of similar happenstances.

(iii) Data is not ingested on real time, cleansed, pre-processed and ready to consumption. The data stores do not cater to semi-structured and unstructured formats.

(iv) Collected demographic data is not complete. Social media information is not explored. Very less information is known about non-frequenting customers.

The above findings have prompted the Froggy management to carry out an architectural review and upgrade to its current IT systems.

Proposed Data Lake Architecture

***DigiCard** Interface is expected to a play a major role in achieving goals of universal electronic payments, a less-cash society, and financial inclusion.*

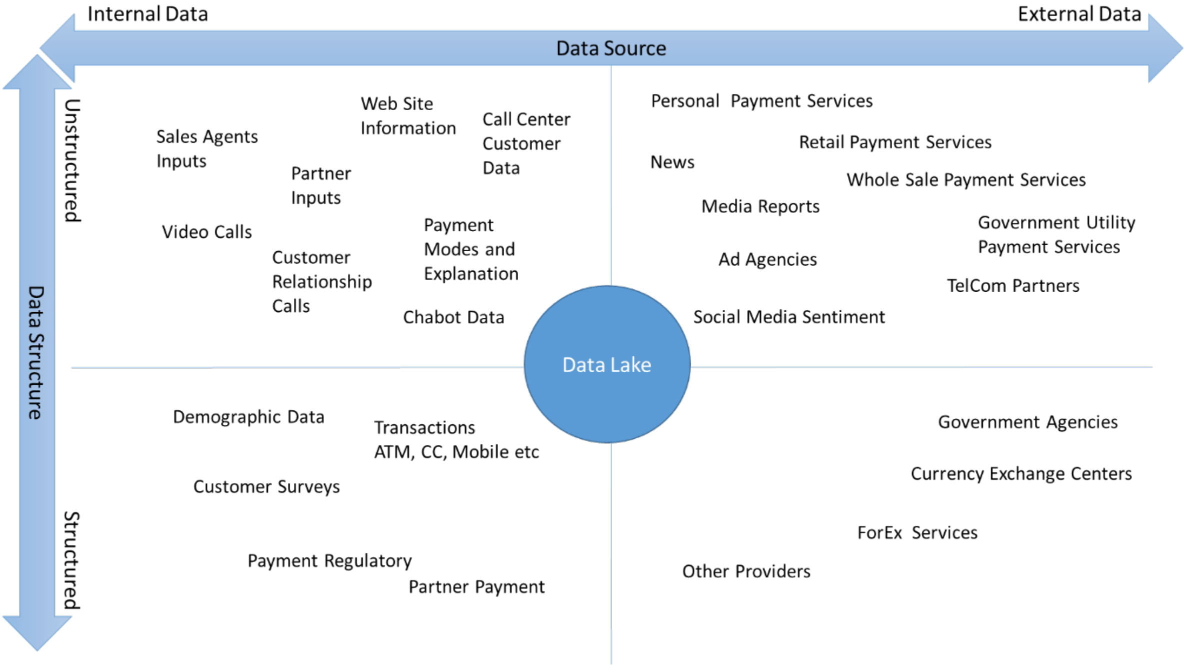
As a preventive measure against fraud incidents and risk scoring, *Froggy* is looking at building a proactive new-generation analytics processing that is data driven. In this direction, *Froggy* proposes to evolve a data lake and has identified new data sources that augment existing enterprise data to find customer's behavioural patterns and device details. Data includes customer information, device information, transaction patterns, usage patterns and other useful permissible data from third parties. *Froggy* wants to employ the new platform to aid better risk scores, insurance products, chat-bots, log analytics and fraud pattern detection.

In *Froggy*'s approach to build a data lake, your team has been tasked to analyse different design options for building the various components. Essentially the proposed information architecture has to address newer big data ingestion policies, data pre-processing, storage, archival, and newer processing capabilities that would address the various analytics needs specific to identified DigiCard use cases.

- **Data Lake**: The data lake will hold the data in raw format and should support flexible schema to facilitate lucid design solutions. The Data Lake should collect both structured and unstructured data to gain real-time visibility into physical, virtual or cloud infrastructure.
- **Customer Data**: This comprises of the customer's profile and transaction history from various sources as earlier highlighted. The data is expected to be voluminous, of different transaction types, of high veracity and would also comprise of data from third parties as mentioned earlier. Your team is expected to design a data store that would support analytics modelling.
- **Decision**: Once the analytics pipeline processes a transaction, it will arrive at a decision which is binary: whether the transaction should be approved or rejected, and the reasons for the decision should be stored back into the Data Lake.
- **Processing**: The data stored must be batch processed for offline score calculations. Also the data needs to be automatically processed by a real-time processing framework and make updates to the Data Lake as necessary. Near real time processing can detect trends and events unfolding over a few minutes that you would not be able to see by inspecting each transaction in isolation (for example, suspicious activity across multiple customers in the same geographical region or same retailer may need to be flagged).
- **Querying**: The data should also be capable of being explored and analysed by humans using querying tools or frameworks such as Spark. Through this newer insight on user

behaviour can be discovered which can be incorporated into the customer transaction logic and stored in Data Lake

A rough map of different kinds of data that the Data Lake should collect is shown below:



New Processing Challenges

The power of data can be realized only if the decision makers are able to make use of the data. *Froggy* is exploring the promising opportunities that Data Lakes are evolving as a place to store the big data in the enterprise with availability for daily transactional systems, traditional data warehouses as well as the new data science directed analytics projects. On the analytics aspects, *Froggy* is planning to use the data stored in the Lakes for log analytics, marketing analytics, social media analytics and media analytics.

Moving forward *Froggy* targets to address the following challenges:

1. The company is planning to extract insights from high volume and velocity transactional data.
2. *Froggy* has a need for real-time analytics for faster/better decisions and point-of-service use. For example, *Froggy* must address security threats to financial transactions through mechanisms that will handle the fraud incidents on real-time and get insights really fast.
3. *Froggy* desires a dynamic approach for extracting customer behavioural insights from a variety of transactional data where business units can tap or purify the required information when they need it.
4. *Froggy* expects to improve its top line and wants to augment internal data (such as customer profile data) with external data (social media and non-traditional data) from a variety of sources. This can get a broader customer view and better behavioural profile of the customer, resulting in quicker customer acquisition.

Data Growth and Response time

DigiCard is growing very fast and become more complex in the volume (terabyte to petabyte), variety (structured and un-structured and hybrid), velocity (high speed in growth) in nature. This means that Froggy's infrastructure and data grows at a rapid rate.

In 2015, Froggy's payment database was under 10 terabytes. But the latest upgrade was 140 - 150 terabytes, and still growing fast. With global digital payments volumes projected to reach over 700 billion annual transactions by 2025, Froggy needs to think in terms of much bigger volumes than they currently have even now.

Following are key response time values for payment:

Response Time	**Significance**
0 to 1 Second	It is most preferred response time. If the response time is 0.1, customers always feel that the application or system is responding instantly, and do not feel any interruption. - Payment in the order of micro seconds - Clickstreams in the order of milliseconds - Event in the order of milliseconds
1 to <3 Seconds	It is the defined as the maximum limit of acceptable response time. Customers are unlikely to feel any interruption, though they may experience some delay. The response time of more than 1-second may interrupt user experience.
3 Seconds	It is a maximum limit after which response time goes beyond the acceptable limit.

Future Directions and Projects

Froggy desires to derive insights using machine learning, artificial intelligence, predictive analytics and statistical techniques to simplify financial decision making and provide superior solutions. In this initiative the company is evolving its data and processing architecture and is exploring the following projects:

Risk Analysis

Froggy offers various types of loans to its customers. It ranges from routine credit cards advancing to loans for real-estate or business capital. Froggy has classified its loans into about a dozen loan-types. Depending on the loan-type, the company applies a sequence of assessments on risks prior to sanctioning the loan. The decision process involves a product mix of structured and unstructured inputs. At present these are based on documentation provided by customers and some external rating.

Currently, *Froggy* is dependent on external credit rating agencies and credit scoring companies like FICO for borrower background information. Moving forward *Froggy* plans to to establish in-house capabilities and is setting up a Data Lake to collect and store relevant information regarding customer credit risk background. In addition, *Froggy* plans to invest on building machine learning solutions to provide instant credit risk score on borrowers rather than to wait offline for weeks to get credit rating scores from external agencies. For example, they propose to use logistic regression to predict the risk of customers and to segregate good borrowers from bad ones.

In essence, Froggy aspires to evolve a Credit Scoring system that prudently relies on extensive data at its disposal. A credit Score is a number used by lenders as an indicator of how likely an individual is to promptly repay his debts and the probability of going into default. *Froggy*'s desires that its credit score computation to be quite thorough and complex involving multiple criteria leading to the following five guiding factors:

1. **Customer payment history**. This shows whether customer makes payments on time, how often they miss payments, how many days past the due date they pay bills, and how recently the payments have been missed. The higher proportion of on-time payments, the higher credit score will be for the customer.
2. **How much a Customer owes on loans and credit cards**. This is based on the entire amount of customer owing, the number and types of accounts customer have, and the proportion of money owed compared to how much credit is available in the customer's account. High balances and maxed-out credit cards will lower the credit score, but smaller balances can raise it – if customer pays on time.
3. The **length of customer credit history**. The longer history of making timely payments, the higher credit score customer gets.
4. The **types of accounts a customer holds**. Having a mix of accounts, including instalment loans, home loans, and retail and credit cards may improve credit score.
5. **Recent credit activity**. If a customer has opened a lot of accounts recently or applied to open accounts, it suggests potential financial trouble and can lower customer score.

To build a credible Credit Score and consequently establish good risk-mitigation, *Froggy* requires a large set of data from various sources. The technical challenges they anticipate include the need to accept data in raw form in various types and a perineal flow of data that pertains to both existing customers and prospective customers with adequate safe-guards against data security and privacy. On acquisition of the data the company may need to have tools, policies and architectures in place for cleansing the data and using intelligent engines for identifying unreliable data and weeding these out. A good architecture is hence necessary if this project is to be reliable.

Insurance Products

Froggy is interested in expanding into the Insurance industry by offering bundled insurance products along with DigiCard on selected services. *Froggy* already stores credit profiles in their internal data lakes and hence can easily extend the insurance based on customer's credit scores. Insurance purchase data also helps *Froggy* for other purposes such as credit scoring, customer acquisition, marketing, customer retention, and designing new insurance products.

Log Analytics

Froggy plans to use log analytics to uncover patterns in customer behaviours, identify problems, audit security activities or ensure compliance with established rules, and plan for capacity or IT infrastructure changes. *Froggy* identifies few key events for performing analytics on the first phase. An event is an identifiable or significant occurrence within hardware or software, and the information about that event is recorded in the log file. A customer or a computer system can generate an event. The logs are analysed for specific patterns. The project aims to build custom dashboard(s) to share with the senior management or to create reports for specific operations managers. The project must facilitate search for distributed logs with one click and quickly gain operational insights into performance root cause.

SECTION A

All Questions in this section are based on the ***Froggy FinTech*** case study described. Your answers should pertain to the case study scenario and to earn full credit, you should site examples and/or justifications based on the case context as appropriate.

Note: Where information is inadequate, you may make appropriate assumptions. Please state such assumptions clearly in your answers.

Question 1 *(refer to case study)*

(Total: 25 Marks)

You are *Froggy's* chief data architect involved in designing a scalable data store and are now tasked to provide appropriate ***Information Architecture and Data Design***.

a. Consider the data store described in the *"Current Architecture"* section of the case study and **provide** an architectural analysis of its weakness based on the future directions and projected growth of the organisation *(refer also to 'limitations' & 'challenges' described in the case study).* Using ANY TWO limitations of your choosing, **suggest** how you would improve the data store & processing architecture.

(8 marks)

b. Consider the project entitled *"Risk Analysis"* as well as the current data schema design described in the case study. Using the six-step iterative *'Design-By-Query'* methodology, **derive & design** a new aggregate for computing the *Credit Scores* that will resolve issues present in the current architecture.

Note: For the purpose of this exam answer, it is sufficient that you identify data aggregates based on data retrieval and storage operations of Risk Analysis project ONLY.

(12 marks)

c. *Froggy* desires to build *Customer Credit Score* data sets from multiple sources such as credit rating organisations, social networks, POS transactions etc. This will necessitate their IT systems/infrastructure to cater to high volume, velocity and variety situations and hence the *Customer* related data may need to be distributed across multiple nodes.

Identify ANY TWO possible *sharding strategies* that will handle the above stated high volume/velocity/variety situation and **recommend** which you consider is the better strategy; **provide** justifications.

(5 Marks)

Question 2 *(refer to case study)*

(Total: 15 Marks)

gWhile architecting the system for *Froggy Fintech* described in case study, you are faced with the challenge of proposing a robust and resilient ***Big Data Ingestion Architecture***.

a. Data Ingestion Recommendation: Consider the *Payment Transactions* described in the *Data Lake Architecture* section of the case study. The following data cleaning and pre-processing are necessary for ensuring reliable, query fit and analysis worthy data:

i. Segment ONLY key lifestyle indicators for each payment categories.

ii. Classify payments based on customer activities, interest, life events and other key parameters.

iii. Enrich customer payment data with DigiCard 's unique institutional needs.

iv. Profile Currency Exchange details if applicable.

v. Profile customer intent, digital engagement and journey if possible.

Recommend an appropriate ingestion tool and **identify** appropriate ingestion policy for the *Payment Transactions* considering the above data cleansing and preprocessing requirements. **State** contextual reasoning for your recommendation.

(5 Marks)

b. **Real Time Data Streaming Recommendation:** *Froggy* uses streams and logs as input for several of its projects. **Propose** a Real-Time *processing architecture* to ingest data from ***click streams*** and ***user logs***, that would help perform behavioural analytics and visualize the results via integrated dashboard. **Provide** an implementation design for the above.

Note: Your architectural proposal should include Real Time Stream artefacts recommendations such as Topics, Partitions, Window, Tools, frameworks etc.

(10 marks)

Note: *Your answers should:*

- *List design assumptions (if any) that you make.*
- *Appraise your choice in the light of the case study and demonstrate why it is suitable.*

Question 3 *(refer also to Appendix, in particular the pilot projects suggested)*

(Total: 10 Marks)

a. Referring to the *"Expected Analytics Needs"* section of the case study, answer the following questions from a ***Processing Architecture*** perspective.

Based on the requirements specified in case study, *Froggy's* current IT consultants have provided an architecture diagram which is shown overleaf.

Review the *Log Analytics on Fraud Payment Forensics* process described below and answer the questions that follow.

> **Log Analytics on Fraud Payment Forensics:** The most common types of DigiCard frauds includes, credit card fraud, counterfeit, card not present, lost/stolen card, intercepted in post, fraudulent application etc. On the basis of execution media, we can broadly classify them into three categories:
>
> i. Card Related Frauds: Credit card frauds like account takeover, lost/stolen cards, application related frauds, fake and counterfeit card, skimming etc. falls in this category.
>
> ii. Internet Related Frauds: This section includes pseudo DigiCard generator, site cloning, false merchant site, bin attack, tele-phishing, and balance transfer checks etc. falls in this category.
>
> iii. Supplier involved frauds: These type of fraud comprises supplier involvement, triangulation etc.
>
> Thus *Froggy* intends to perform log analytics to identify various fraud patterns, mine insights regarding context and indicators. By doing so, in future, Froggy is hoping to identify such occurrences in the real time payment transactions.

Identify the missing components in the existing architecture overleaf. **Propose** additional tools to meet the above *Log Analytics on Fraud Payment Forensics* and briefly **justify** your recommendation.

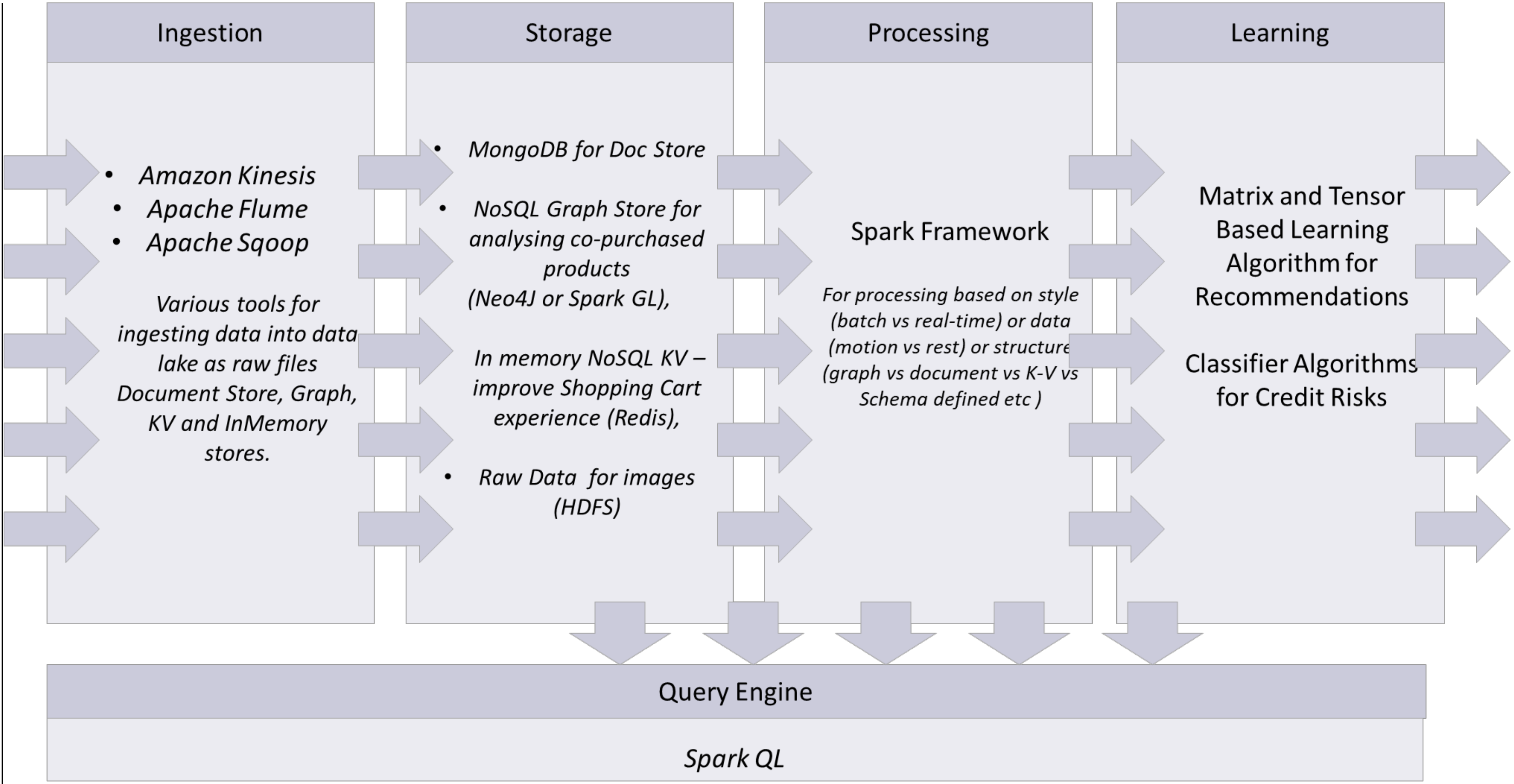
For the additional tools you recommended, **design** processing stages, functionalities and pipelines.

(10 Marks)

Note: *Your answers should:*

(1) List design assumptions (if any) that you make.

(2) Appraise your choice in the light of the two above mentioned scenarios and demonstrate why it is suitable.



Ingestion

- *Amazon Kinesis*
- *Apache Flume*
- *Apache Sqoop*

Various tools for ingesting data into data lake as raw files Document Store, Graph, KV and InMemory stores.

Storage

- *MongoDB for Doc Store*
- *NoSQL Graph Store for analysing co-purchased products (Neo4J or Spark GL),*

In memory NoSQL KV – improve Shopping Cart experience (Redis),

- *Raw Data for images (HDFS)*

Processing

Spark Framework

For processing based on style (batch vs real-time) or data (motion vs rest) or structure (graph vs document vs K-V vs Schema defined etc)

Learning

Matrix and Tensor Based Learning Algorithm for Recommendations

Classifier Algorithms for Credit Risks

Query Engine

Spark QL