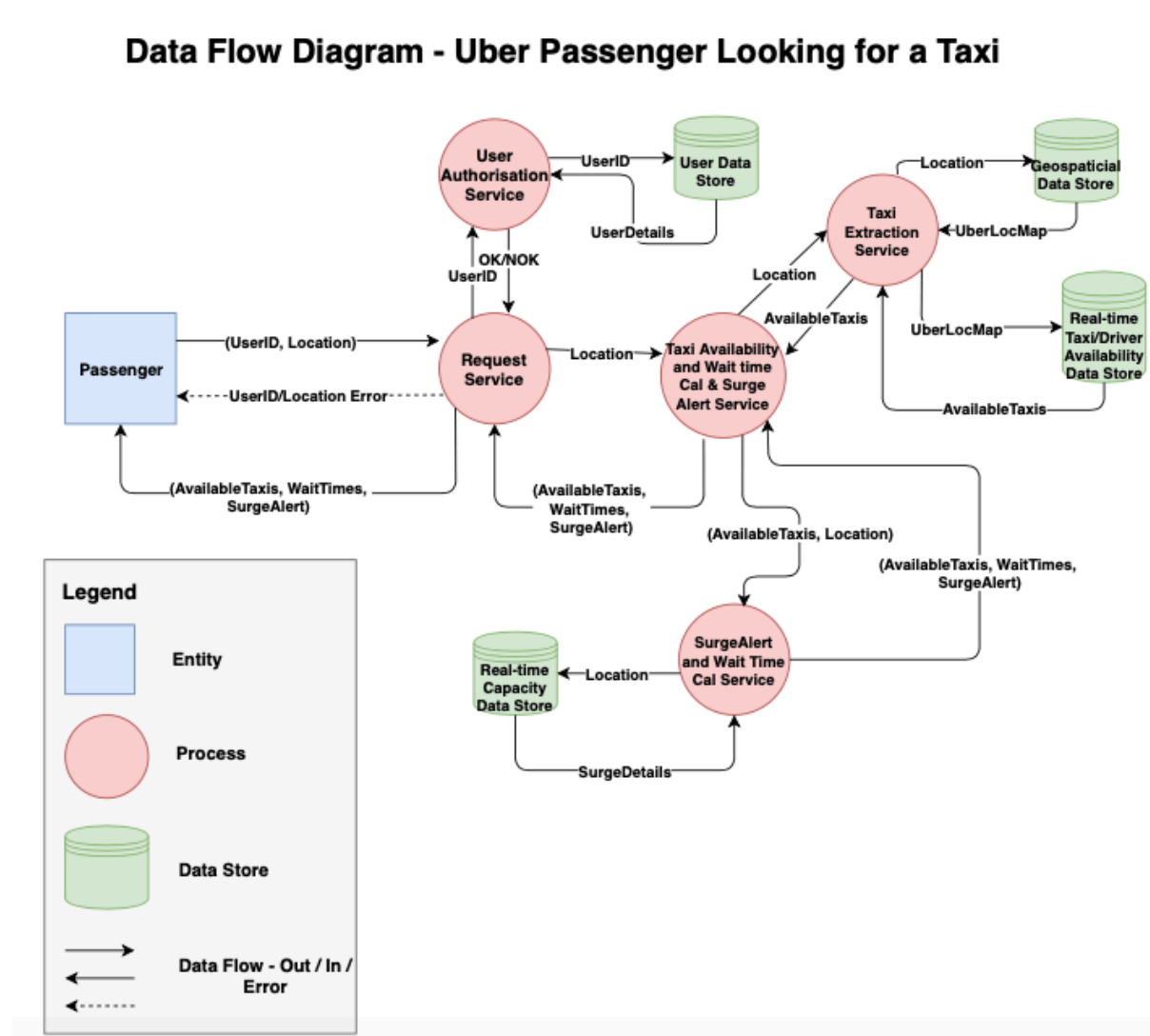T3A1.1Q1 - **Develop** a data flow diagram which shows the flow of data in an organisation's information system.

## Data Flow Diagram - Uber Passenger Looking for a Taxi



The above dataflow diagram shows the information flow through Uber systems and processes when an Uber passenger(customer) open the Uber Taxi App on their phone. The Uber app opens within a few seconds with Google Maps showing the address and taxis on the map waiting for the customer to enter the destination.

Passengers device sends the LocationID and UserID to Uber Request Service. Request Service send the UserID to User Authorisation Service. User Authorisation service queries the User Data store to get the User details. If the Authorisation Service identifies, the user is eligible for a ride, then the flow proceeds with the rest of activities. There are situations where the user is not authorised for a ride, like when the users credit card has expired. Then an error is sent back to the user's device via the Request Service.

Uber uses Service Oriented Architecture with lots of micro-services. If the user is authorised, then the location of the user is passed on to a service to get available taxis, wait times and any other capacity (surge) related issues. It is very rare in the cities for Uber to send back a location error. Location is passed on to Taxi Extraction Service. Taxi Extraction Service uses Geospatial data store to map location to UberLocMap.

On the customers device, the location is on Google Maps. Uber uses its own mapping solution H3 to work out taxis available close to a customer. Google has divided land into hexagon and starts the search in the hexagon and neighbouring hexagons and so on. Taxi Extraction Service uses UberLocMap to extract taxis from a Real-time Taxi availability data store. Real-time Taxi availability data store is constantly updated with the taxis by processes that are monitoring events and writing to the data store. This is as real-time as possible. The accuracy is not 100% but as close as possible.
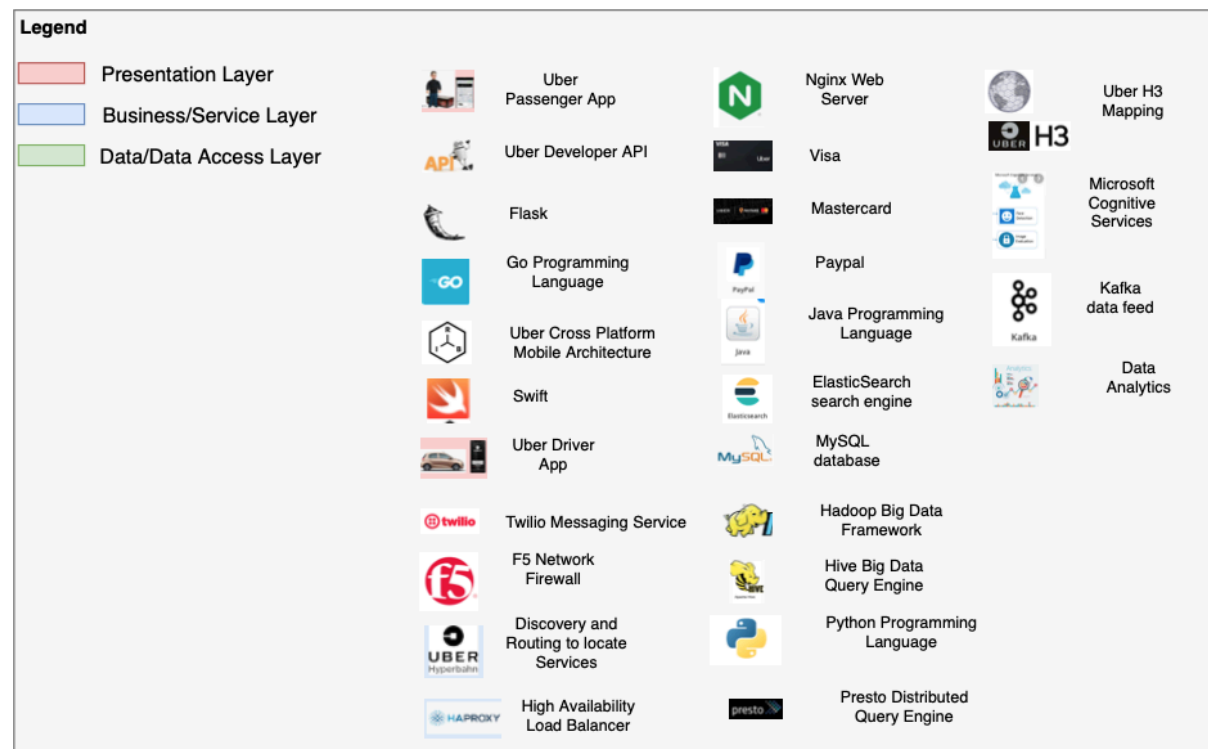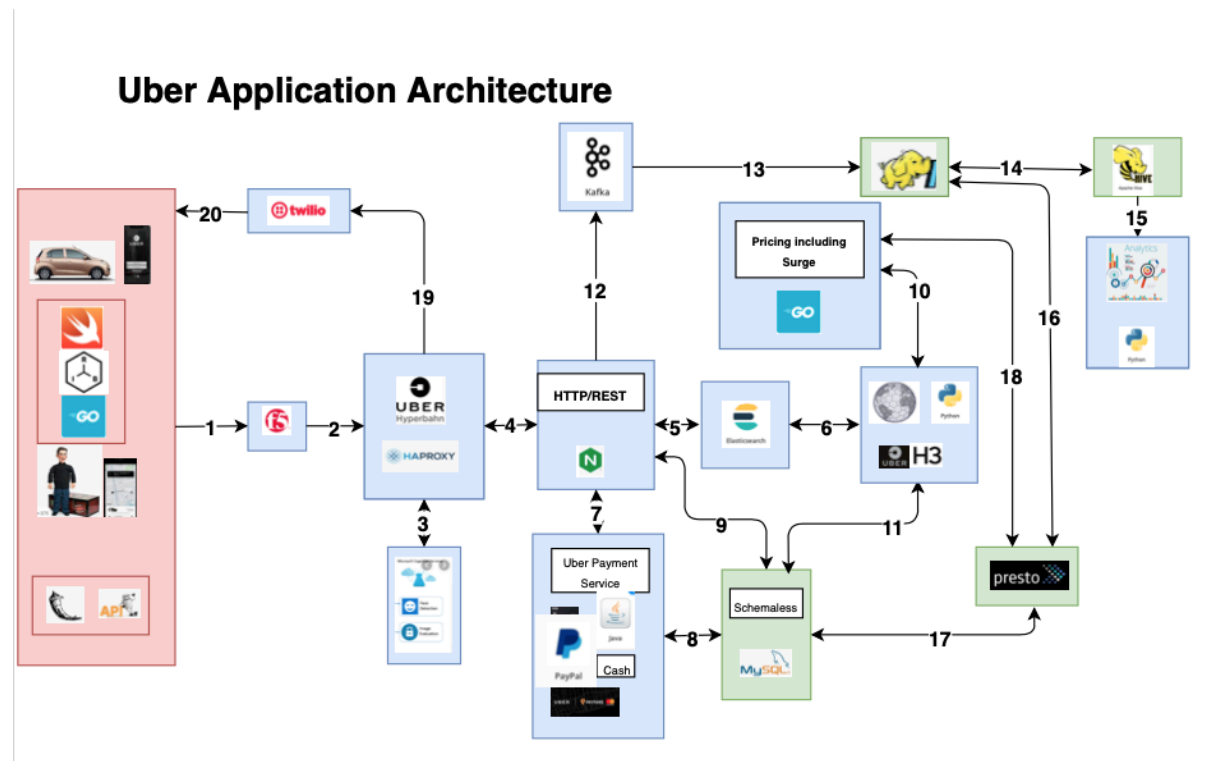
Before sending the taxis to customers device to be displayed on the map, there are other calculations required to work out the wait times and any demand related issues. If the demand is high Uber charges extra for the ride. Uber has a surge pricing model. Surge Alert and Wait time calculation service uses the Real-time Capacity data store to work out the surge pricing and the wait time. Interesting fact to notice is that when a location is classified as a surge zone, the wait times are smaller. This is because all the taxis move to that location or hexagons, as drivers are likely to make a lot more money by servicing that location. But when a location is in a surge zone, all neighbouring non-surge zones have longer wait times.

Once taxis, wait times and surge alerts are worked out, it is sent back to customers device that displays this information.

Reference:
https://eng.uber.com/

T3A1.1Q2 - Develop an application architecture diagram
which **describes** relationships between technologies and applications within an
organisation's information system.



## Uber Application Architecture



Legend

- Presentation Layer
- Business/Service Layer
- Data/Data Access Layer

| Icon | Description |
|------|-------------|
| | Uber Passenger App |
| | Uber Developer API |
| | Flask |
| | Go Programming Language |
| | Uber Cross Platform Mobile Architecture |
| | Swift |
| | Uber Driver App |
| | Twilio Messaging Service |
| | F5 Network Firewall |
| | Discovery and Routing to locate Services |
| | High Availability Load Balancer |
| | Nginx Web Server |
| | Visa |
| | Mastercard |
| | Paypal |
| | Java Programming Language |
| | ElasticSearch search engine |
| | MySQL database |
| | Hadoop Big Data Framework |
| | Hive Big Data Query Engine |
| | Python Programming Language |
| | Presto Distributed Query Engine |
| | Uber H3 Mapping |
| | Microsoft Cognitive Services |
| | Kafka data feed |
| | Data Analytics |

1. All requests for Uber Services from passengers, drives or developers and systems using Uber API go through F5 network firewall. The firewall has all the latest security threat intelligence updates.
2. Once the request is verified by the firewall, the request is forwarded to HAProxy, a high availability software load balancer to locate the right service to fulfil the request. There is another component called HyberBahn used for locating services. Uber is phasing out HyberBahn and moving to HAProxy. There are other functions performed like driver validation performed before using any services.
3. Validation of driver is performed using Microsoft Cognitive services. Drivers selfie is verified with the ones on record and if the selfie doesn't match, then the driver cannot pick up a passenger.
4. HAProxy Routes the request to REST api running on Nginx web server to fulfil a taxi request. There are many other services that Uber offers using REST.
5. Uber uses elastic search to optimise requests for taxis or offering cheaper delivery fee for Uber eats service when there are many other similar requests for food pick up in a location. Booking a taxi involves multiple interactions and some interactions are avoided by looking for the most recent valid data.
6. Elastic search looks for data using Uber H3 for Geospatial data related to Uber Service the customer is requesting. Some parts of this service is developed in Python. The hexagonal mapping of all the cities to work out the taxis, drivers, surge etc is developed in C. In case of Uber Taxi, the customer is in a hexagon and H3 service can work out the closest hexagon with taxis in seconds.
7. When the customer has booked a taxi or completed a ride or paying to the driver is completed using payment service. Uber uses several micro services. In some countries, Uber also provides the option of paying by cash. The working for paying by cash is different to other methods of payment. Some services that interact with banking network is written in Java.
8. To complete a payment, payment service uses customer payment details from MySQL database. Uber has developed a layer on top of MySQL to make it a schema less database which makes interaction using JSON possible.
9. There are services on the web server that also uses schema less database to insert and retrieve data. Some examples of this are adding a star rating to the driver or retrieving the favourite restaurants of a customer in case Uber eats.
10. Pricing and H3 mapping modules have processes to work out pricing based on capacity and other factors. This helps Uber to work out prices based on demand and traffic situation.
11. All location related events of customer or any other entity is stored in schema less database.
12. All activities or data feed about anything that is considered valuable for service improvement and business operations is fed into Kafka. Kafka is a data feed manager that can write to big data storage.
13. Data feed is passed on to Hadoop for aggregation and management. Hadoop is a big data framework to aggregate data from multiple sources and manage data.
14. Uber uses Apache Hive to query big data for required information.
15. Uber Analytics uses Hive to extract data for Business Analysis.

16. Uber uses Presto to query MySQL and Big Data stores for day-to-day business operations. Presto connects to Hadoop to extract data required for business operations.
17. Presto connects to Schema less MySQL database extract data for business operations.
18. Pricing uses Presto to get data that can work out prices including surge prices. Surge pricing model is charging more for a service when the service is in high demand. Some pricing modules are written in Go programming language.
19. Twilio messaging service is used for all interaction between Uber and customer or any other entity.
20. Twilio sends the message to the entity.

Reference:
https://eng.uber.com/
https://stackshare.io/uber-technologies/uber

T3A1.1Q3 - **Identify** TWO products, services or priorities of an organisation and for each:
- **explain** how their application stack contributes to the delivery of the service or priority (2 marks)
- **speculate** why they chose the particular application or combination of applications (1 mark)

Two priorities of Uber are **never run out of taxis** and **safety of passenger**.

**Never run out of taxis**

Application stacks contribution to the delivery of the service or priority

When a passenger opens the App, there is always taxi available.  Uber adjusts the price and wait times to achieve this. Uber uses a combination of applications Kafka + Hadoop to get real time feed of people looking for taxis and the supply of taxis by location.  This information in combination with the data analytics helps to identify demand areas and time windows. Higher price increases the supply of taxis to the location. Kafka data feed with Big Data framework Hadoop in combination with Uber's in-house algorithms make "Never run out of taxis" possible.

Why they chose the particular application or combination of applications (Speculation)

Uber uses a lot of open source software. Uber has changed their application stack in the last five years. Pinterest, LinkedIn, Facebook and other large organisation inspired them, and Uber used their tested and scalable products to improve service to millions of customers around the world.

**Safety**

Application stacks contribution to the delivery of the service or priority

Uber uses driver images to verify identity and ensuring the safety of the customer. Uber uses Face detection provided by Microsoft Cognitive services on Azure platform to verify drivers.

Why they chose the particular application or combination of applications (Speculation)

Storing sensitive identity data is a big corporate responsibility. To avoid additional responsibilities, it is likely to create, it is better to use it as a service. Another possibility is that the service provider may already have a license to store and collect data of this nature in many countries.