SE 339 Software Architecture and Design

Final Exam - Fall 2019

Instructor: Lotfi ben Othmane

Student: Stamatios Morellas

Section 1 (30 points): Understanding

Question 1.1 (10) What architecture problem does client/service style solve?

1- Ensure good performance of the system.
2- Guarantee that data exchanged between the system's components are not lost.
3- Allow the clients to share resources such as data.
4- Enable project managers to assign developers to components based on their skills.

Answer:

Benefits of using the client-server style are higher security, ease of data management, and ease of maintenance. Disadvantages are scalability and reliability, since the system relies on a centralized server. Therefore, option **(3)** is the most correct.

Question 1.2 (10) Give two conditions that makes a design decision an architectural decision?

Answer:

A decision is architectural if it has non-local consequences and/or the consequences matter to the achievement of the architectural drivers.

Question 1.3 (10) Suppose that you are assigned to design the architecture of a complex Web application. The customer requested that the system should support modification **and** deployment of one or a set of components at runtime without impacting the behavior of the overall system. Give one architecture patterns, tactics, or styles that you may use to address this requirement.

Answer:

I would use a **component-based style** to address this requirement. By using a component-based style, we can create a "pluggable" or modular architecture, which can easily be customized to include/exclude various components for different runtime configurations.

Section 2 (50 pts) Read the paper "On the Definition of Microservice Bad Smells" by Davide Taibi and Valentina Lenarduzzi and published by the IEEE software magazine, which is provided in Canvas and answer the questions below.

Question 2.1 (5 pts): Enumerate five of the characteristics of microservices listed in the paper.

Answer:

(1) Small

(2) Autonomous

(3) Have a single, clearly defined purpose

(4) Allow for independent deployment

(5) Newly developed architectural style

Question 2.2 (10 pts): Give three harmful practices that creates issues for the microservices developers.

Answer:

(1) Hardcoded IPs and Ports – makes it harder to change the service location in the future

(2) Shared Persistence – using shared data among services that access the same database

(3) Static Contract Pitfall – using microservice APIs that are not versioned, possibly causing people to connect to an older version of the microservice

Question 2.3 (10 pts): Enumerate three smells that **do not** have straightforward solutions.

Answer:

(1) Wrong Cuts

(2) Microservice Greedy

(3) Too Many Standards

Question 2.4 (10 pts): What **is** the main lesson that developers should practice when they migrate a monolithic application to microservices-based application?

Answer:

Lesson **(3)** is the main lesson that developers should practice when migrating a monolithic application to a microservices-based application.

Question 2.5 (15 pts):

(a) What is **API** gateway?

Answer: An API gateway gathers all of the API calls from clients and routes them to the appropriate microservice. It essentially acts as a "middle-man" between the microservices and the client(s).

(b) What problem does it address? and

Answer: Using an API gateway will make it easier to add and remove microservices since it directly manages the relationship between them and the clients. This means that when there is an increase in the number of microservices, the difficulty of managing all of the microservices can be more controlled.

(c) How can it be used to address smells?

Answer: If used, an API gateway could potentially help remove some of the other risks (or smells), since it has to do with how the microservices communicate with each other.

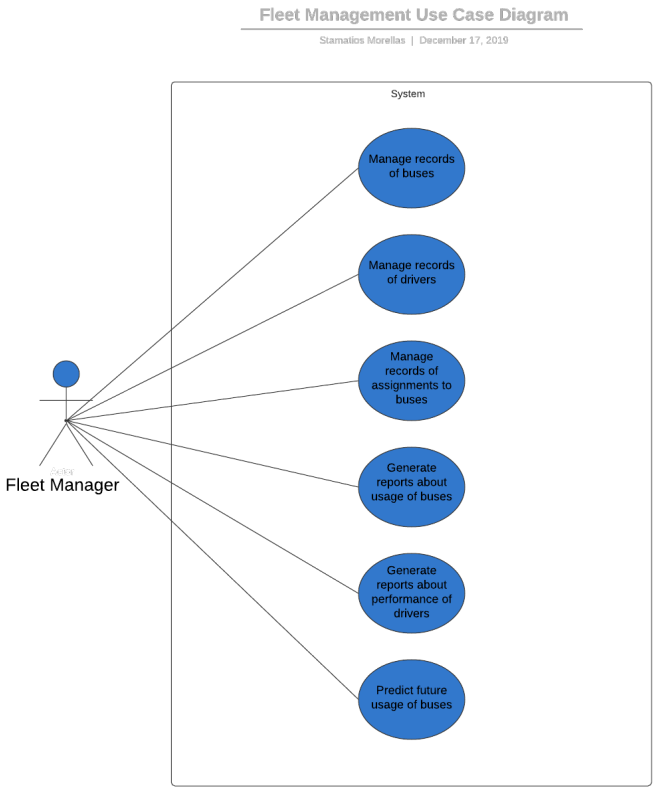
Exercise (60 points):

You are assigned to manage a project for developing a software to manage the city busses. The goal of the software is to provide the fleet managers with synthesized reports that they can use to manage their fleet. The main functional requirements of the system are:

1. Manage the records of all the busses. Each record includes the vehicle identification, the license plate number, the date of start service, the date of acquisition, and the acquisition amount.
2. Manage the records of the drivers, including driver license, date of start service, name, phone number, and address of each driver.
3. Manage the records of assignments of drivers to busses.
4. Generate reports about the usage of the busses, the performance of drivers, and the book value of the fleet.
5. Predict the future usage of buses by lines and income for the future months and years.

The enterprise architecture team at the company decided early in the year of the following architecture policy: front-end components shall be developed using AngularJS, backend components shall be developed using Node.js, libraries should be developed using Java language, and the enterprise architecture team decides on the open source software that could be used in the company's products.

Question 3.1 (15 pts): Develop a use case diagram that captures the six features of the system. Provide also a table for the use cases with two columns: use case name and use case description.

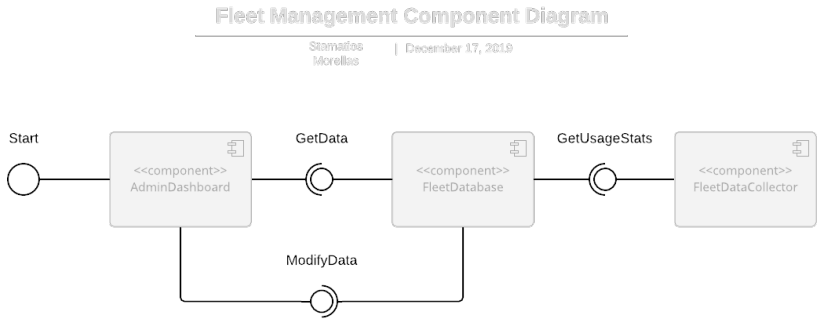


The use-case diagram (shown above) can be constructed based on the functional requirements listed in the project description. The table for the use cases is shown below.

Use Case	**Use Case Description**
Manage records of buses	View and modify the records of each bus in the fleet. This includes vehicle ID, license plate number, date of start service, date of acquisition, acquisition amount
Manage records of drivers	View and modify the records of each driver in the fleet. This includes driver license, date of start service, name, phone number, and address of driver.
Manage records of assignments to buses	View and modify the records of driver assignments to buses.

Generate reports about current usage of buses	Generate a report from the data provided by each bus's usage data collector.
Generate reports about performance of drivers	Generate a report from the performance data of each driver
Predict future usage of buses	Given all current usage data, generate a prediction of what the future usage data of the fleet could look like.

Question 3.2 (15 pts): Develop a component diagram that captures the capabilities of the expected software. Provide also a table for the components with two columns: use component name and component description.



The component diagram (shown above) can be constructed based on the various components needed in order to satisfy the use cases. The table for the components is shown below.

Component Name	**Component Description**
Data Management Component (AdminDashboard)	A web-app (or online interface) that can be used by the fleet manager (actor) to display and interact with the fleet management service records and data.
Data Collection Component (FleetDataCollector)	A device installed in every bus that is involved with this service to collect on-board usage data and send the data to the server.

Data Storage Component (FleetDatabase)	A server that stores all the collected usage data of the fleet, in addition to all of the vehicle, driver, and assignment records.

Question 3.3 (30 pts): Assume that you presented the diagrams of Question 3.1 and 3.2 to the customer in the project kick-off meeting. The representative of the customer questioned if the architecture addresses the following requirements:

1- Support scaling the data analytics component(s) independently of the other system components,
2- Permission to access the data analytics components should be managed based a Centralized Access Control List.

You are asked to extend your previous architecture to address the two requirements.

a. Identify one or a set of architecture approaches (styles, patterns, tactics) that addresses quality attribute 1. (10 points)

Answer:

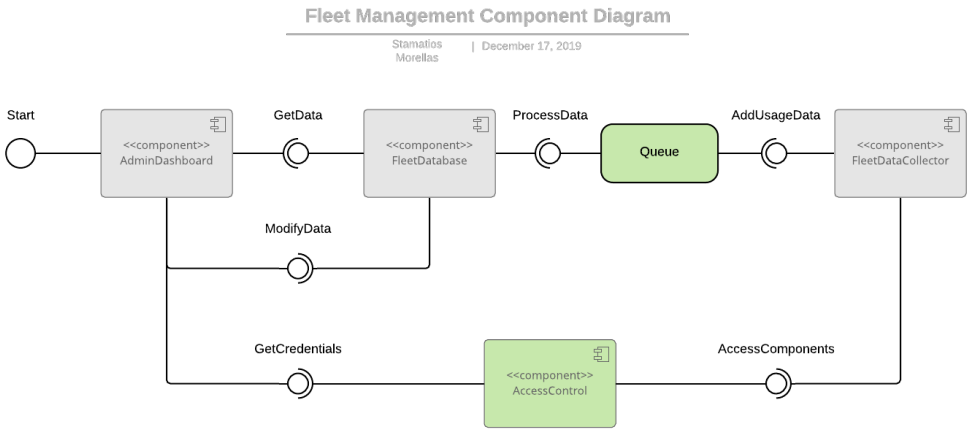
I would use the queue management pattern to address this quality attribute. Since scaling the data collection components could cause a heavy load on the server to process and store the data, implementing a queue management pattern could help mitigate the processing time for the collected usage data.

b. Identify one or a set of architecture approaches (styles, patterns, tactics) that addresses quality attribute 2. (10 points)

Answer:

I would use the gateway pattern to address this quality attribute. Since permission is required to access the data collection component directly, a gateway component can be introduced to determine whether or not the data collection component may be accessed by the current actor.

c. Extend the components diagram that you developed in Question 3.2 to support these solutions. You should also provide a components description table like in Question 3.2. (10 points)



Component Name	**Component Description**
Data Management Component (AdminDashboard)	A web-app (or online interface) that can be used by the fleet manager (actor) to display and interact with the fleet management service records and data.
Data Collection Component (FleetDataCollector)	A device installed in every bus that is involved with this service to collect on-board usage data and send the data to the server.
Data Storage Component (FleetDatabase)	A server that stores all the collected usage data of the fleet, in addition to all of the vehicle, driver, and assignment records.
Queue	A component introduced in order to ensure performance, reliability, and minimal data loss when the system is scaled.
Centralized Access Control List (AccessControl)	A component introduced in order to determine whether or not access can be granted to an actor requesting direct access to the data collection component.