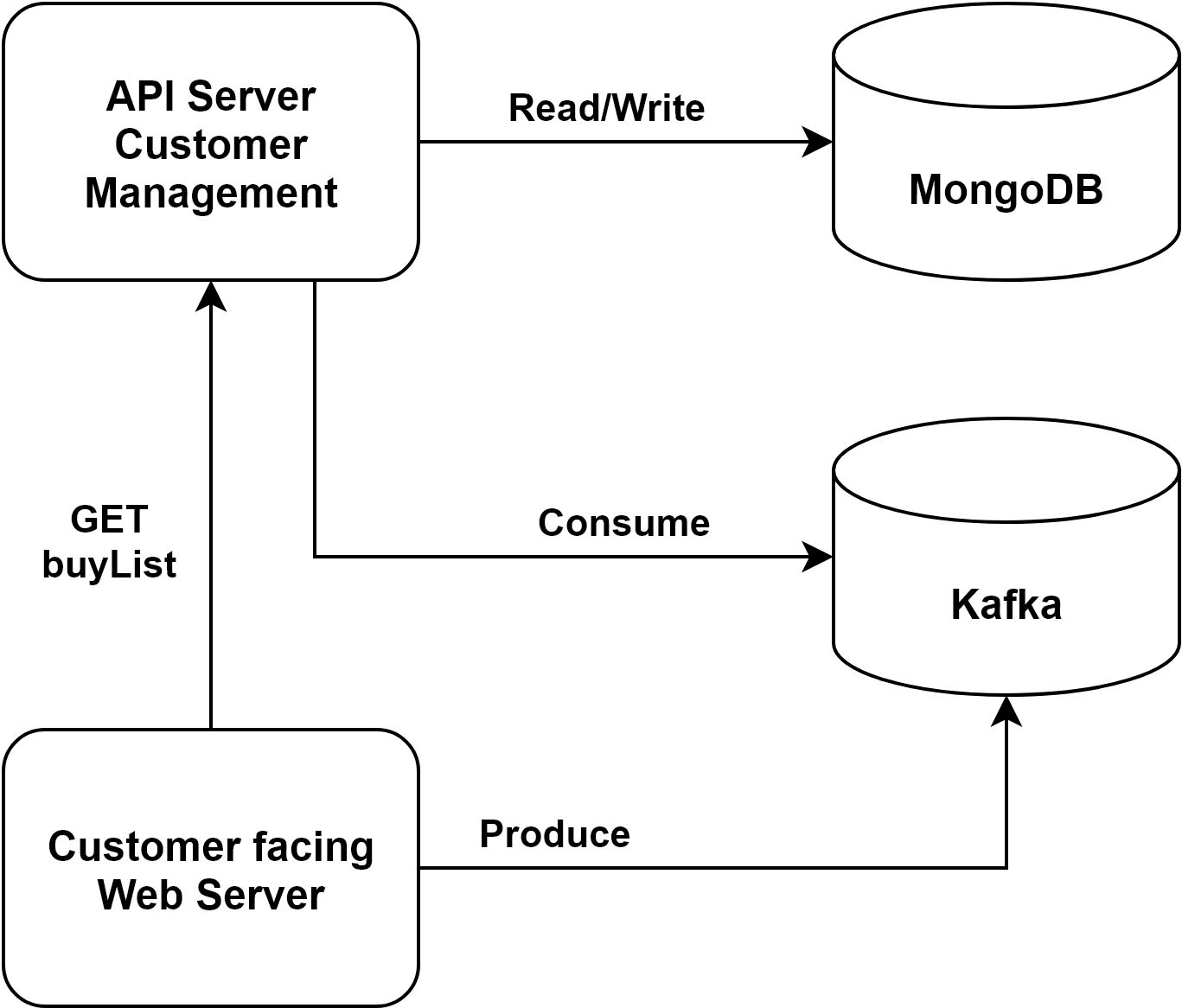
# **Home Assignment**

# Design

[](https://app.diagrams.net/?page-id=kUYQotXFGYtgS2nCEFAh&scale=auto#G1ufg4CNogX-qtlJNekFBJw4lfb0IiHNrx)

# 

# **Background**

The above diagram describes a simple system for a certain user to buy random items and get a list of all the items that he bought. You need to implement the entire system in a way that you see fit.

# 

# **Architecture**

## MongoDB

1. Store the user’s purchases in MongoDB as you see fit

## Customer Management API

1. Read & Write data into MongoDB
2. Consume messages from Kafka
3. GET route - Return all customer purchases

## Customer Facing Web Server

1. Handle a “buy” request and publish the data object to Kafka. (username, userid, price, timestamp)
2. Handle a “getAllUserBuys” and send a GET request to Customer Management service and present the response

## \* Bonus 1 - Autoscaling

Add autoscaling resources for the software components based on metrics that are relevant to the use case

\* hint - CPU and memory are not the only metrics you can use

## \* Bonus 2 - Frontend

Add a UI frontend component that will use the customer-facing web service

1. Button 1 named Buy - Send a purchase request
2. Button 2 name getAllUserBuys - display all purchase requests for the user

# **General Guidelines**

* Write the components in JavaScript/TypeScript or Python (JS/TS is preferred but not a must)
* The setup needs to run on Kubernetes
* Provide clear instructions on how to get your setup up and running
* Add clear remarks with explanations of what's been done in your code