University of Central Lancashire UCLan Department of Computer Science	**UCLan Coursework Assessment Brief**	2022-2023
	Module Title: Distributed Systems	
	Module Code: CO3404	Level 6
	Microservices Development	This assessment is worth 50% of the overall module mark

OVERVIEW

This assignment will assess your ability to implement and test a distributed application as a service, across a distributed system of microservices. You will gain experience of delivering a solution using a modern software architecture utilising modern hosting approaches and tools used in industry by some and in the future by many, including: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), DBaaS (Database as a Service), Containerisation, Virtual Machines, Cloud Hosting, Object Data Modelling, application development in HTML, CSS and JavaScript utilising a NodeJS runtime environment, Express middleware, documentation of API (Application Programming Interface) contracts using the OpenAPI standard, relational and document-based NoSQL databases.

NOTE: develop and test using local host before cloud deployment, so you do not exhaust your Azure credit. Shut down or delete virtual machines and any PaaS services when not in use. It is your responsibility to monitor your cloud usage which is something a developer needs to be aware of in industry.

HIGH LEVEL REQUIREMENTS

You will develop, test, deploy and document a Jokes Service. Users can request a random joke, submit a new joke and a moderator can moderate the new jokes. The full service will consist of three microservices using the architecture illustrated in Figure 3 at the end of this document. Figures 1 and 2 illustrate simpler architectures for the lower grades.

Your microservices should be loosely coupled, independent and resilient to failure of any of the other microservices. The requirements for each of the three microservices, numbered 1 to 3 below, is:

1. Deliver a random joke on request by a user from a web page. The user can select the type of joke from an up-to-date list of available joke types. This will be deployed using Infrastructure as a Service on Microsoft Azure using a single virtual machine. The jokes will be stored in a MySQL relational database.

2. Enable users to submit new jokes. They will submit their joke using a web page and be able to choose from an up-to-date list of joke types to categorise their submission. This will be deployed into a combination of IaaS and Database as a service. The application will be deployed into an Azure VM, and the application will communicate with a MongoDB database. The database will utilise the free cloud service offering, MongoDB Atlas. You will use an Object Data Model (ODM) approach in schema design and implement it using the Mongoose NodeJS package.

3. Enable a moderator to access a submitted joke. The moderator needs to authenticate to this microservice. They will be able to request a new joke from the “Submit Jokes” microservice, edit the joke, change the “type” the submitter has chosen if necessary or add a new type based on the joke content. They can then either submit the new joke to the “Deliver Jokes” microservice, or not, if it is either not considered funny or is inappropriate. The moderator will then delete the joke from the “Submit Jokes” microservice and the next one is presented for moderation.

ASSIGNMENT DELIVERABLES:

- Source code and any shell or docker scripts you have created
- A narrated video, no more than 10 minutes, showing all the functionality of your service
- A short report of no more than 2000 words

MODULE LEARNING OUTCOMES ASSESSED (in part)

1.	Critically evaluate patterns, technologies, and frameworks
2.	Compare potential technologies for the development of a distributed enterprise system
3.	Implement a distributed application using appropriate technology and frameworks

ASSESSMENT CRITERIA

for 40%

1. See figure 1 for the minimal solution architecture required for this grade band
2. Implement the “Deliver Jokes” microservice
3. Implement the ”Submit Jokes” microservice to submit new jokes into the new jokes database
4. The two microservices should run on local hosts using different port numbers
5. Provide a ‘/doc’ path to serve up your open API standard contract documentation for each microservice
6. A video illustrating service operation, testing and API documentation
7. A report, no more than 2000 words, containing the following:
    a. Self-assessment / critical evaluation of how well you have satisfied the requirements and what areas could be improved. Using the assessment criteria, propose an overall grade justifying the mark you are awarding yourself. This will help you to carefully review the requirements and critically assess your own work. Your mark will not count but will be used as a measure of your ability to self-assess. i.e., it should be close to my awarded mark as I will be following the same specification
    b. A discussion of the design patterns used and their appropriateness for a service of this size and complexity. Propose other potential approaches and compare them to your solution
    c. What alternative technologies, deployment models and frameworks could you have used? Compare those to the ones you have used

for 50%

8. See figure 2 for the solution architecture required for this grade band
9. Connect the “Submit Jokes” and “Deliver Jokes” microservices via an API. i.e., No moderator service, just submit the joke into the new joke document database **and** into the “Deliver Jokes” service via an API
10. Deploy the “Deliver Jokes” and “Submit Jokes” microservices into the Azure cloud
11. “Deliver Jokes” should be deployed directly into an IaaS Linux-based Virtual Machine
12. The “Submit Jokes” should be deployed into an IaaS Linux-based Virtual Machine, (not the same one as “Submit Jokes”), and the MongoDB database should be deployed into the: MongoDB Atlas, DaaS (database-as-a-service) service

for 60%

13. See figure 3 for the solution architecture required for this and higher-grade bands
14. Create the “Moderate Jokes” microservice
15. Deploy the “Moderate Jokes” microservice into a Docker Container using the Azure Web Apps PaaS service
16. Read a joke from the “Submit Jokes” microservice into the “Moderate Jokes” microservice, submit it, unedited, to the “Submit Jokes” microservice and delete the new joke from the “Submit Jokes” microservice

for 70%

17. Enable the moderator to request, edit, change the joke, and add or change the submitted joke type. Submit the modified joke and type to the “Deliver Jokes” microservice
18. Ensure that the “Submit Jokes” microservice can operate effectively if the “Deliver Jokes” service is down
19. Provide simple authentication for the moderator, as discussed in lectures

For 85%+

20. For a high first class mark you need to build on the topics covered in the lectures, demonstrate self-directed study, and implement functionality that is related to, but extends beyond the lecture material. Some examples are:
    a. A more comprehensive authentication approach for the moderator
    b. Use of an API key for the submitter
    c. Implement one of the microservices using Java
    d. Create certificates to employ SSL/TLS, in-transit security for web clients
    e. Your own ideas
    f. Check with the module leader before implementation to determine suitability of the suggestion(s)

PREPARATION FOR THE ASSESSMENT

Before completing the assessment, you should ensure you are up to date with all the practical lab exercises on Blackboard. These exercises and provided videos are designed to give you experience working with the tools and techniques required to complete the assignment.

RELEASE DATES AND HAND IN DEADLINE

Assessment release date: Monday 05/12/2022
Assessment deadline date and time: **Friday 10/03/2023**

Please note that this is the final time you can submit – not the time to submit!
Your feedback and mark for this assessment will be provided on **Monday 17/04/2023.**

SUBMISSION DETAILS

IMPLEMENTATION SUBMISSION:

1. **Before submission make sure you have made a backup of your project.**
2. Submit only source code such that I can run the application if I need to. i.e., provide package.json, .env or whatever files are needed. Do not provide re-creatable files. You do not need to export all your data; I will use my own
3. Add the completed assignment coversheet and self-assessment into your project folder
4. Submit a .zip file of your project folder via the 'CW Implementation Submission' link on Blackboard.

NARRATED DEMO VIDEO SUBMISSION:
The video should be submitted via the 'CW Demo Video Submission' link on Blackboard. An unlisted YouTube link or Google Drive / OneDrive link is fine if the video file size is too large.

REPORT SUBMISSION:
Your report must be in Microsoft Word .docx format and submitted via the 'CW Report Submission' link on Blackboard

HELP AND SUPPORT

- Support will be provided via Microsoft Teams and email. You will also have the opportunity to ask questions during lectures / labs. You may request a one-to-one meeting with a tutor during their office hours (as published on Starfish).
- For support with using library resources, please contact our subject librarian subjectlibrarians@uclan.ac.uk. You will find links to lots of useful resources in the My Library tab on Blackboard.
- If you have not yet made the university aware of any disability, specific learning difficulty, long-term health or mental health condition, please let us know. The Inclusive Support team will then contact you to discuss reasonable adjustments and support relating to any disability. For more information, visit the Inclusive Support site.
- To access mental health and wellbeing support, please complete our online referral form. Alternatively, you can email wellbeing@uclan.ac.uk, call 01772 893020 or visit our UCLan Wellbeing Service pages for more information.
- If you have any other query or require further support, you can contact The Student Support Centre. Speak with us for advice on accessing all the University services as well as the Library Services. Whatever your query, our expert staff will be able to help and support you. For more information, how to contact us and our opening hours visit Student Support Centre.
- If you have any valid mitigating circumstances that mean you cannot meet an assessment submission deadline and you wish to request an extension, you will need to apply online prior to the deadline.

Disclaimer: The information provided in this assessment brief is correct at time of publication. In the unlikely event that any changes are deemed necessary, they will be communicated clearly via e-mail and a new version of this assessment brief will be circulated.	Version: 2 Updated 01/09/2022

Figure 1: Minimal Architecture

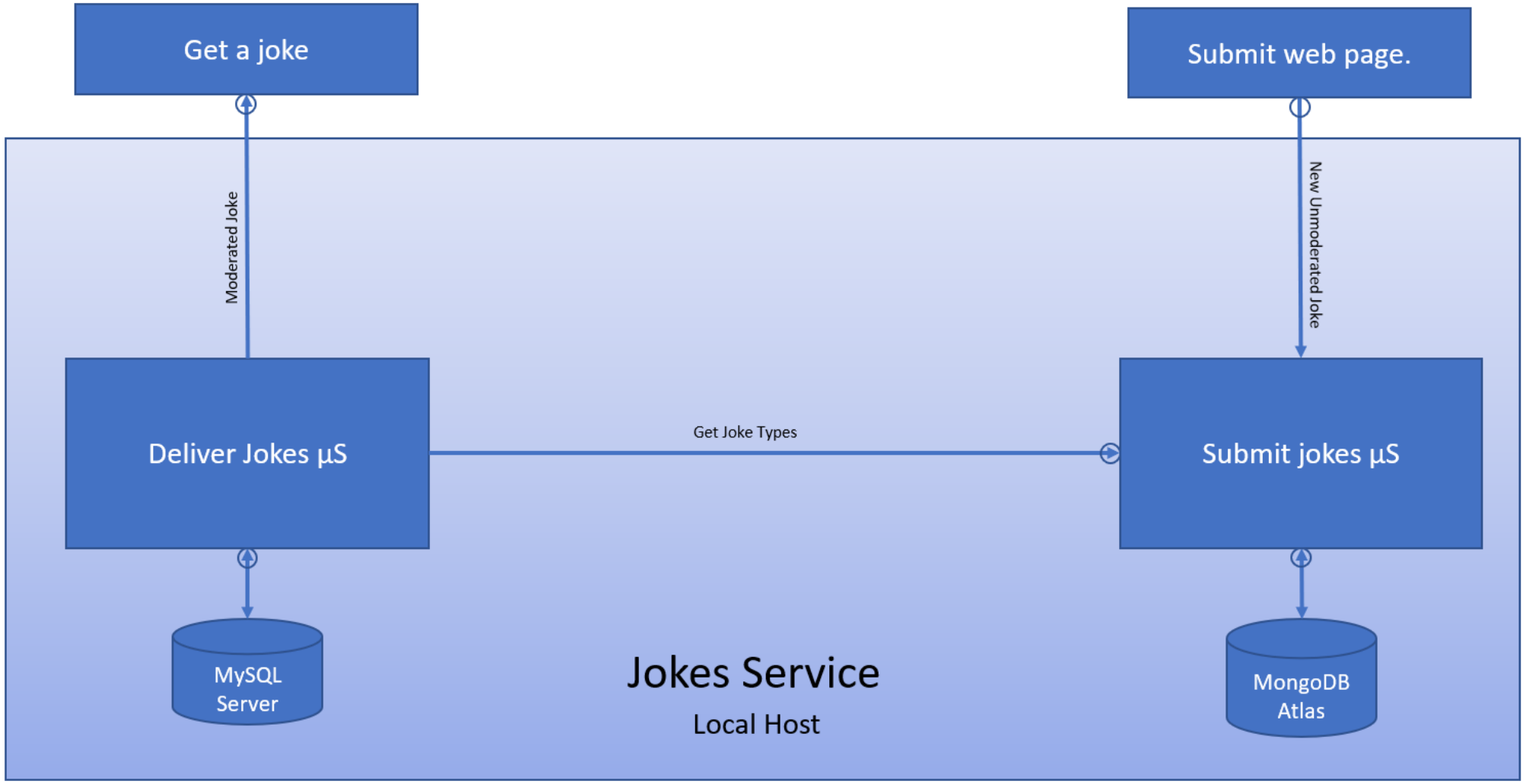

Figure 2: Intermediate Architecture

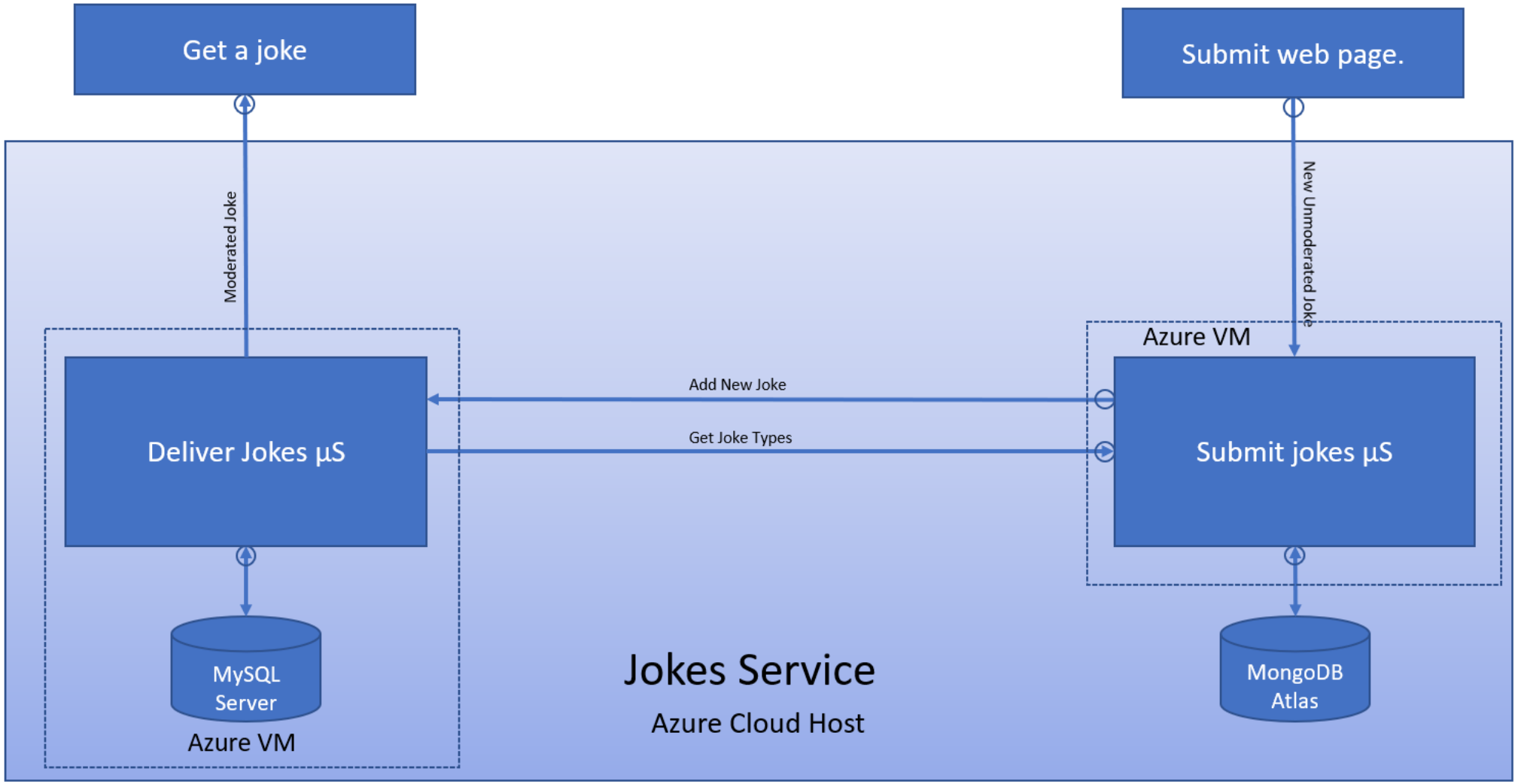

Figure 3: Full Architecture