Section A — Answer ALL questions

1. Identify TWO existing distributed systems that illustrate two extremes of **openness**. In other words, one of these systems is very open and the other one is not open at all. Explain your answer.

   (6 marks)

2. Consider a distributed system that has to be developed and then deployed on a network. Networks can be classified according to a number of parameters. Some of these parameters are very likely to be of importance to distributed system developers and other parameters are less likely to matter to such developers. List SIX network parameters that are very likely to matter to distributed system developers.

   (6 marks)

3. Explain what the following Erlang code fragment does in all possible circumstances:

```erlang
receive
    { newMessage } -> silenceWatcher(EnquirerID)
after 500 -> ok
end,
EnquirerID ! { silence }.
```

   Focus on the occurrences of messages but do not explain the purpose of these messages.

   (6 marks)

4. Give an example of a client-server application that benefits from the use of a **cache**. ALSO, explain why a **cache** benefits this application.

   (6 marks)

5. Define the concept of a peer-to-peer (P2P) distributed system, stating all its important characteristics.

   (6 marks)

6. Explain the role of the Simple Object Access Protocol (SOAP) in Web services.

   (6 marks)

7. Briefly describe a real-life computational problem for which it is beneficial to use a **cluster**. Justify your answer.

   (4 marks)

END OF SECTION A

Section B — Answer TWO questions

8. Consider the following specification of a **RESTful service**. This service allows clients to make enquiries at a specific hotel.

   The server exposes instances of the following types of resource:

   - a collection of user accounts on the URL `"http://my.hotel/users"` supporting method POST

   - a user account on URL matching `"http://my.hotel/users/{userId}"` supporting method GET and also PUT for existing accounts; represented in XML

   - a collection of enquiries on URL matching `"http://my.hotel/users/{userId}/enquiries"` supporting method POST

   - an enquiry on URL matching `"http://my.hotel/users/{userId}/enquiries/{enqId}"` supporting method DELETE; represented in XML

   - a response to an enquiry on URL matching `"http://my.hotel/users/{userId}/enquiries/{enqId}/response"` supporting method GET; represented in XML

   a) Describe the design of a Remote Procedure Call (RPC) Web service that provides a functionality equivalent to the RESTful service above.

   All functionality should be covered using NO MORE THAN FOUR *procedures.*
   Use a Java-like notation to describe your remote procedures.   (11 marks)

   b) Describe the design of a Web Service Resource Framework (WSRF) Web service that provides a functionality equivalent to the RESTful service above.

   Make sure all functionality is covered and a substantial use is made of WSRF features.
   List all URLs that expose services and all available procedures on each URL.
   You will need some degree of creativity to complete this task.   (11 marks)

   c) Describe how a service equivalent to the one described at the beginning of this question could be implemented in Erlang.

   Assume that all clients are also Erlang programs.
   Focus on the messages that are sent between client and server in typical interactions.
   You are not expected to write Erlang code.   (8 marks)

8(a)

userId addUser(userInfo)

userInfo getUserInfo(userId)

enqId submitEnquiry(userId, enquiry)

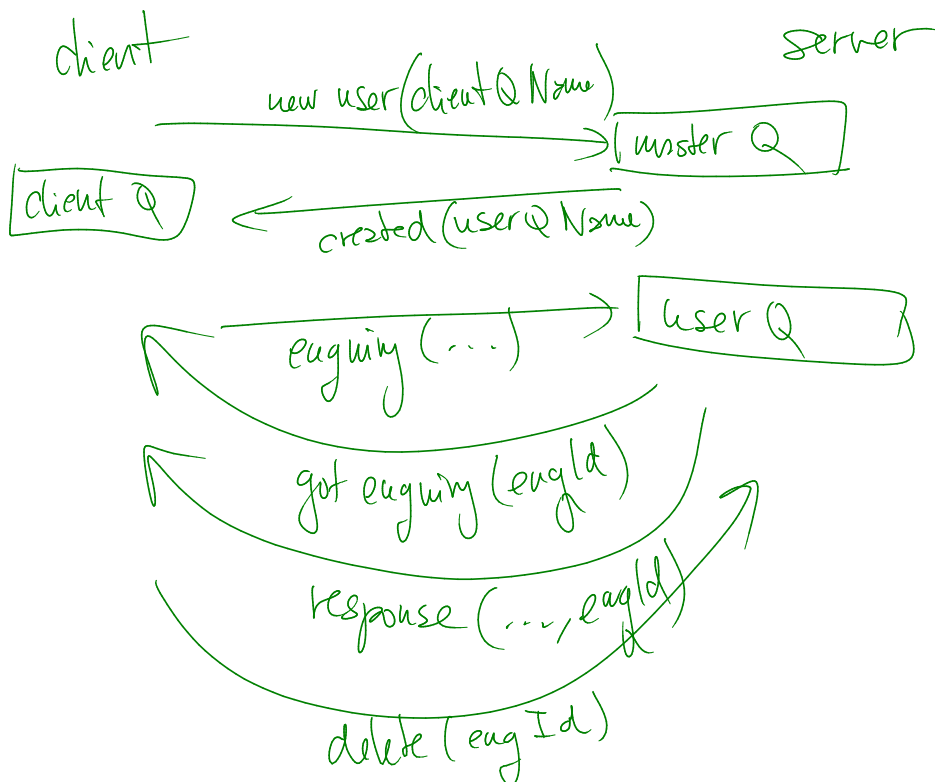response getResponse(enqId)

removeEnquiry(enqId)

(b)

user factory: EPR CreateResource()

user service: getInfo()

　　　　　　　EPR submitEnquiry(enquiry)

enquiry factory (hidden)

enquiry service:

　　　　　　　response getResponse()

　　　　　　　remove()

Java RMI
would work
very similarly
to WSRF

(c) s/Erlang/JMS/g

client

new user(client Q Name)

→ master Q

client Q

created (user Q Name)

enquiry (...)

→ user Q

got enquiry (engId)

response (..., engId)

delete (eng Id)

server

9. a) Expand the acronym SOA as used in the context of distributed systems.

    (2 marks)

    b) Briefly define the concept of SOA.   (4 marks)

    c) State THREE characteristics of services that are suitable for participating in a SOA distributed system. ALSO, for each characteristic state why it is desirable.

    (9 marks)

    d) State THREE of the benefits that SOA is believed to give businesses that make use of it. ALSO, for each benefit explain why it is a benefit.

    (6 marks)

    e) Give an example how nodes in a SOA distributed system interact.   (3 marks)

    f) Explain the need for standards in THREE different aspects of SOA in general. Illustrate each aspect with a suitable example.

    (6 marks)

10. a) A **factory of remote objects** is often used in distributed systems, which are built using a **distributed objects** middleware such as Java RMI. Define the concept of **factory of remote objects** AND state its main purpose.

(4 marks)

b) State THREE typical features of a factory of remote objects that enhance the way in which the factory fulfils its main purpose.

(6 marks)

c) Consider a system where clients can use a server to create and move a number of players around a virtual grid. The server updates a drawing of where the players are currently located. The server relies on the services of a factory of remote player objects.

The following collaboration diagram illustrates the first two steps taken in this system when a client asks the server to create a new player:
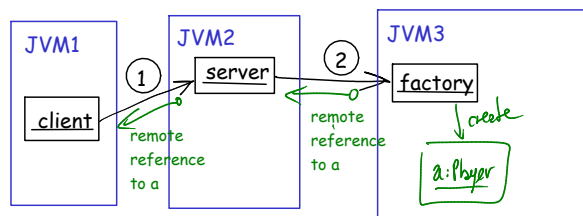
i) Draw a copy of the diagram in your answerbook and add further calls to indicate what happens inside the factory and how it responds to the call marked (2) in the diagram.

You may need to add one or more new objects to the drawing.
Number all calls to indicate in what order they are executed.
Briefly state for each call which method it invokes and what parameters are passed.
Also explain what value each call returns if any. Describe also calls (1) and (2) that are already shown in the diagram above.

(9 marks)

ii) Draw a diagram that illustrates what happens when the client calls a player's method to make the player move. Include calls until the point when the server redraws the player in its new position. Assume that the player notifies the server of its move.

You can reuse the drawing that you made in task (i) or make a new one.
You may need to add one or more new objects to the drawing.
Number all calls to indicate in what order they are executed.
Briefly state for each call which method it invokes and what parameters are passed.
Also explain what value each call returns if any.

(11 marks)

END OF EXAMINATION PAPER

Jvm1