

# COMP3008 Distributed Computing

## Assignment 1

**Due date: 25 September 2022 11:59pm AWST**

**Total marks: 40**

This is a group assignment. A maximum of two persons can be part of a group. It is okay if you want to submit it alone if you don't find a group partner.

The assignment has two parts:

- **Part 1:** It is a Turnitin report submission. Basically, you have to answer some questions. **[10 marks]**
- **Part 2:** Programming submission. Here, you need to build an application based on the given specification. You also have to submit a report where you should mention your individual contribution. **[30 Marks]**

Details of these two parts and marks distributions are discussed below:

### Assignment 1 (Part 1) – 10 Marks

Create a document or report by answering the following questions. Your document should have a cover page where your information (id, name, email etc.) should be provided.

Q1. Consider a Web-based newscasting system that has a server 'publish' news. This news is sent via callbacks to clients that have 'subscribed' to the newsfeed. Clients are untrusted in that they communicate over the Internet and anybody is able to write a client and connect to the service. There are three choices in the way the callback can be implemented:

- The server calls back to clients with normal (synchronous) calls
- The server calls back to clients with asynchronous calls
- The server calls back to clients with oneway calls

- (i) Discuss a major problem with using normal synchronous calls to have the server make callbacks to the (untrusted) clients and explain why asynchronous and oneway calls do not suffer from this problem. **[3 Marks]**
- (ii) Of the asynchronous and oneway approaches, which model would you choose to implement the newscasting callback? Justify your choice. **[3 marks]**

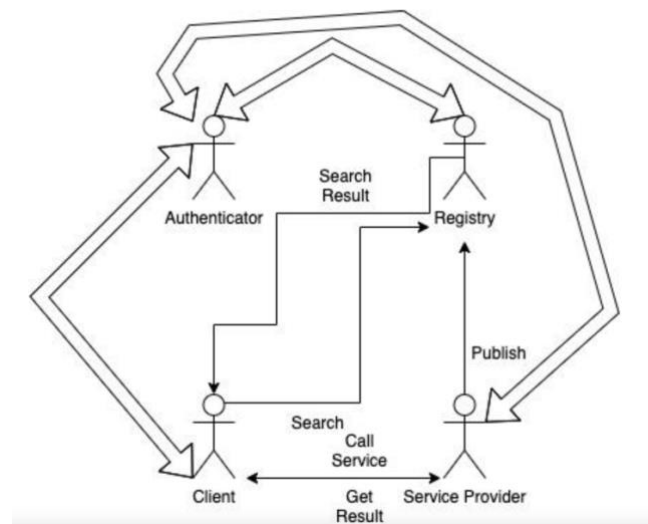
Q2. Consider that you are given the task to build a distributed system that will provide consolidated reporting across all divisions to the directors of the company. The system requires a single business-tier server to access multiple different databases from different divisions in the company. Code to access the appropriate databases can either be located in the business tier or the data tier. In which tier would you place this code? Justify your choice. **[4 Marks]**

**Part 1 Submission Guideline:** This is a Turnitin submission. Just upload the document (pdf or .docx) in the part 1 submission portal. **Your similarity index should**

be below 10% from the Turnitin report. If you submit in a group, only one member should submit the same file.

### Assignment 1 (Part 2) – 30 Marks

Your task is to build a simple SOA (service-oriented architecture) described in the following picture:



You have to create four separate projects where each project represents an actor of the SOA:

1. **Authenticator:** It provides authentication services. The other actors, i.e., client, Service provider, and registry communicate with the authenticator when they need to validate any information.
2. **Service provider:** It provides basic mathematical services, i.e., add, multiply, generate prime numbers, and check for prime numbers.
3. **Registry:** It provides basic service registration services for the providers and has searching services for the client.
4. **Client:** Its job is to invoke and test the services that are provided by the service provider.

The basics of the simple SOA are as follows: the service provider publishes its services to the registry. It means the name of the service, its description, and service endpoints are stored in the registry. The client can search for a service in the registry using the description of the service. The search result will return the actual service endpoint, and the client can use the endpoint to invoke the service. In any service invocation among the client, provider, and registry, the actors must prove that they are authenticated by the authenticator.

Every project instance will run on a single machine. So, we don't need service orientation in a single project. However, it is expected that you will design the projects using object-oriented principles. Every project may have a business layer and the data

layer. The business layer is for handling the logic and the data layer is for managing the data. You don't need to implement services between layers in a single project.

You are free to use your own messaging (input or output) formats. You don't have to follow the exact formats in the examples given in this document.

We will have a demonstration session to demonstrate each functionality. Build all the projects in a single solution. However, each project is expected to be independent (minimal code reference to other projects in the solution.) The marks are assigned to each instruction. Note, that if you cannot finish a functionality, you will be given partial marks based on your efforts or codes.

### **The Authenticator Project [2 Marks]**

It is a .NET WCF/remoting server. "net.tcp://localhost/AuthenticationService" is an example of a fixed service endpoint. It has three operations open as service functions:

1. String Register (String name, String Password): It expects two operands, i.e., name and password from an actor. It saves these values in a local text file. If successful it returns "successfully registered". **[0.5 Mark]**
2. int Login (String name, String Password): It expects two operands, i.e., name and password from an actor. It checks these values in a local text file. If a match is found, it creates a token (random integer), saves it into another local text file, and returns it to the actor who calls this function. **[0.5 Mark]**
3. String validate (int token): It expects a token and checks whether the token is already generated. If the token could be validated, the return is "validated", else "not validated". **[0.5 Mark]**
4. There is an internal function that clears the saved tokens every 'x' minutes. When you run the authentication server, it will ask for the number of minutes for the periodical clean-up in the console (using multithreading). **[0.5 Marks]**

### **The Service Provider Project [4 Marks]**

It is an ASP.NET Web API project that creates Rest services:

- a) ADDTwoNumbers: This rest service adds two input integers and returns the output in JSON **[0.5 Mark]**
- b) ADDThreeNumbers: This rest service adds three input integers and returns the output in JSON **[0.5 Mark]**
- c) MulTwoNumbers: This rest service multiplies two input integers and returns the output in JSON **[0.5 Mark]**
- d) MulThreeNumbers: This rest service multiplies three input integers and returns the output in JSON **[0.5 Mark]**

There is no data layer in this project.

**[2 Marks]** The provider has an additional business logic before providing the service. Every client should be authenticated before the service invocation. So, the service provider expects a valid token with every service call. The provider calls the validate function of the Authentication service and if validated the service is provided. Otherwise, the following JSON output is sent:

```
{
  "Status": "Denied"
  "Reason": "Authentication Error"
}
```

### **The Registry Project [6 Marks]**

It is an ASP.NET Web API project that creates the following Rest services:

a) Publish: This rest service saves the service description in a local text file. If successful it returns the status accordingly in JSON. This service expects the input in a JSON format. For example, the ADDTwoNumbers service description could be input as follows: **[1 Marks]**

```
{
  "Name": "ADDTwoNumbers"
  "Description": "Adding two Numbers"
  "API endpoint": "http://localhost:port/ADDTwoNumbers" "number of
  operands": 2
  "operand type": "integer"
}
```

b) Search: This rest service searches an input service description in a local text file and returns the service information. You are allowed to use any C# textual search library. For example, a search with the text 'add' may return the following JSON **[2 Marks]**

```
{
  [
    {
      "Name": "ADDTwoNumbers"
      "Description": "Adding two Numbers"
      "API endpoint": "http://localhost:port/ADDTwoNumbers" "number
      of operands": 2
      "operand type": "integer"
    },
    {
      "Name": "ADDThreeNumbers"
      "Description": "Adding three Numbers"
```

```

        "API endpoint": "http://localhost:port/ADDThreeNumbers"
        "number of operands": 3
        "operand type": "integer"
    },
]
}

```

c) AllServices: This rest service returns all the services saved in the local text file in JSON format. **[1 Mark]**

d) Unpublish: Given a service endpoint, this rest service will remove the service description from the local text file. **[1 Mark]**

**[1 mark]** The registry has an additional business logic before providing the service. Every client should be authenticated before the service invocation. So, the registry expects a valid token with every service call. The registry calls the validate function of the Authentication service and if validated the service is provided. Otherwise, the following JSON output is sent:

```

{
  "Status": "Denied"
  "Reason": "Authentication Error"
}

```

### **The Service Publishing Console Application [4 Marks]**

This is a C# console application to publish services. It is up to you how to design the user interface. You need to demonstrate that you can do the following operations.

- e) Registration: the app asks for the username and password in the console and sends them to an appropriate Authentication service. **[1 Mark]**
- f) Log in: the app asks for the username and password in the console and sends them to an appropriate Authentication service to verify. If successful, the returned token is saved in its program memory. This token will be sent as an additional parameter for every subsequent service call. **[1 Mark]**
- g) Publish service: the app asks for the service name, description, API endpoint, and number of operands and operand types in the console and sends them to an appropriate Registry service to publish. **[1 Mark]**
- h) Unpublish service: the app asks for an API endpoint in the console and sends them to an appropriate Registry service to unpublish. **[1 Mark]**

It is expected that you display the result of each operation in a meaningful way. You are allowed to use the RestSharp library.

### **The Client GUI Application Project [10 Marks]**

You are free to use any GUI framework, i.e., WPF or Web or anything. It is up to you how to design the user interface. You need to demonstrate that you can do the following operations:

- a) Registration: the app asks for the username and password in the GUI and sends them to an appropriate Authentication service. **[0.5 Mark]**
- b) Log in: the app asks for the username and password in the GUI and sends them to an appropriate Authentication service to verify. If successful, the returned token is saved in its program memory. This token will be sent as an additional parameter for every subsequent service call. **[0.5 Mark]**
- c) Show all available services: The GUI will call the appropriate Registry service to retrieve all the available services. The list of available services will be displayed in a manner so that they will be selectable. **[1.5 Marks]**
- d) Search a service: the app asks for the service description in the GUI and sends them to an appropriate Registry service. The list of search results will be displayed in a manner so that they will be selectable. **[0.5 mark]**
- e) Testing a service: The user will select a service graphically. Let us assume the user selects the ADDTwoNumbers service. The GUI app knows the API endpoint and number of operands from its search results. Next, it will create the input boxes for the service testing automatically in the GUI. As ADDTwoNumbers needs two operands two input boxes will be shown for the input and a 'test' button. When the button is pressed the GUI will call the service using the API endpoint and display the result. **[4 Marks]**
- f) You should use multithreading when calling the functions and display a progress bar in the GUI **[3 Mark]**

### **Documentation/report [3 Marks]**

1. This is for maintenance and other developers who will look after your code. Let us assume that a new service (e.g., adding four integers service). Provide a step-by-step guide on updating the code to the new service. You will be asked to do the same in your demonstrations. It is expected that you do not need to change your GUI application project when you want to access the new service **[2 marks]**
2. If you work in a group, it will be your report on individual contribution – overall contribution percentage (ideally 50%), which parts you have worked on, how long you have worked, how you communicate with your partner etc. **[1 mark]**

### **Part 2 Submission Guidelines:**

Submit your assignment electronically, via Blackboard, before the deadline. **If you submit in a group, all the members must submit the same files plus the individual report.**

To submit, do the following:

1. Fill out and sign a declaration of originality. A photo, scan or electronically- filled out form is fine. Whatever you do, ensure the form is complete and readable! Place it (as a .pdf, .jpg or .png) inside your project directory.
2. Attach your Assignment report to your project directory.
3. Zip your entire project directory (as a .zip or .tar.gz file). Leave nothing out.
4. Submit your zip/tar.gz file to the assignment area on Blackboard.
5. Re-download, open, and run your submitted work to ensure it has been submitted correctly.

You are responsible for ensuring that your submission is correct and not corrupted. You may make multiple submissions, but only your newest submission will be marked. The late submission policy (see the Unit Outline) will be strictly enforced. Please note:

- DO NOT use WinRar.
- DO NOT have nested zip/tar.gz files. One is enough!
- DO NOT try to email your submission as an attachment. Curtin's email filters are configured to discard emails with potentially executable attachments silently. In an emergency, if you cannot upload your work to Blackboard, please instead upload it to Google Drive, a private GitHub repository, or another online service that preserves immutable timestamps and is not publicly accessible.

**Marking Demonstration:** You will be required to demonstrate and discuss your application with a marker in a one-to-one online/in-person session. Most of the marks for your assignment will be derived from this demonstration. The demonstrator will ask you to rebuild and run your application (provided by the demonstrator) and demonstrate its major features. They may ask you about any aspect of your submission.

The demonstration schedule and policy will be published later (Check blackboard announcements). We may also cancel the demonstration-based marking due to unavoidable circumstances. In that case, it will be a full inspection-based marking.

**Academic Integrity:** This is an assessable task. If you use someone else's work or obtain someone else's assistance to help complete part of the assignment that is intended for you to complete yourself, you will have compromised the assessment. You will not receive marks for any parts of your submission that are not your original work.