# SOFT8026 – Data Driven Microservices
# Assignment 1
Due: 22 March 2023 @ 8pm

## Introduction

In this assignment, you will analyse a requirement using some of the approaches covered so far, e.g. bounded contexts, deciding how a system could be broken up into microservices as opposed to being developed as something monolithic. You will then use Python and Docker, with gRPC and a message queue, such as RabbitMQ, to communicated between microservices. In assignment 2, we will use Kubernetes to orchestrate the system and add some additional testing and monitoring.

## The Requirement

A new video game service is being launched and will eventually rival the Steam platform. The initial system must allow new users to sign up using their email as the login. They can do the following:

- Browse a game catalogue
- Buy games for download
  - They can use vouchers (see below about using points)
  - Games may be discounted for a period of time
- Look at their game achievements / badges
- Earn points to spend on vouchers or other digital assets
- Manage their profile and preferences
- 2 or 3 other requirements you think would be useful (e.g. check Steam, Epic store, etc. for ideas)

## Your Task

1. Analyse the requirement and describe how you would design a microservice-based system for the requirement. You must explain the decision for each microservice chosen and the integration mechanism used, i.e. why gRPC or RabbitMQ was used between two microservices. You must reference 3 reputable sources when discussing your approach to back it up. You may use sketches along the way to illustrate your approach. (1,500 words +- 100 words)
2. Put together a final diagram to show the microservices and how they connect to each other, what messages they send each other and what messaging technologies are being used. You do not have to worry too much about any particular formal notation as long as you diagram is clear and understandable.
3. Implement a portion of the system that includes:
   a. The user registration web page
   b. A datastore for the users
   c. 2 or 3 other microservices of your choice (these should be simple, minimal version of the microservices to keep your workload smaller)
4. Your implementation must include at least 1 gRPC channel between 2 microservices and 1 message queue.

## Rubric

See the rubric in Canvas for further breakdown of marks and what is expected to score well in each of the categories below:

- Requirements analysis text - 35%
- Microservice architecture diagram - 10%
- Implementation of microservice functionality - 30%
- Use of Docker - 10%
- Use of integration mechanisms (gRPC and RabbitMQ) - 15%

## What to Submit

A zip file containing:

- Word document with your analysis text and diagram
- Zipped project files
- Filled-in form (see Word document form attached to the assignment in Canvas)