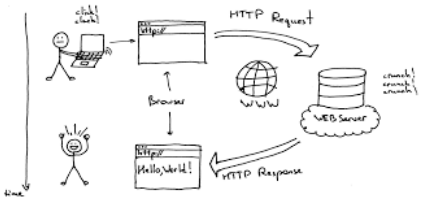
# Worksheet 2: Networking with Unreal and HTTP Servers

Version 1.0
Creative Computing
COMP280

Gareth Lewis




*The problem with troubleshooting is that trouble shoots back.*

*-Author Unknown*

*A computer lets you make more mistakes faster than any invention in human history - with the possible exceptions of handguns and tequila.*

*-Mitch Ratcliffe*

## Introduction

For this worksheet, you are going to develop HTTPServer applications that can be integrated into Unreal Engine, allowing you store persistent game data on a server such that it can be shared between multiple users. This is an ideal architecture for managing shared data systems such as player accounts, high score tables, portable players progression data (to allow player progress to be maintained across multiple machines) and other player and game related data.

These activities should be undertaken in pairs: one BA and one BSc per pair.

## Worksheet Activities

The goal of the activity is to create a body of knowledge that you can all reference for cybersecurity related information and references for academic writing.

To do this:

(A) Follow the workshop in week 2 to create a python-based HTTP Client/Server application
(B) Follow the workshop in week 3 to create a python-based HTTP Server application that will successful communicate with an Unreal client.

## Additional Guidance

Creating an HTTPServer for the Unreal Engine sounds like a big task that is both hard and complex. In reality it is actually comprised of several hard and complex tasks that are fairly small, and we will discover that by breaking large tasks down into smaller tasks they become soluble.
Much of the work in this worksheet is geared around proving concepts through the use of sandbox code to break a large problem into a many smaller problems which can all be solved in isolation. Once these smaller problems have been solved, the knowledge gained from solving them can be used to create the required large and complex solution.

As a pair of programmers, you have the choice to solve problems through pair programming (with a driver and a navigator) or to break into sub teams to solve individual problems and report back to each other. The approaches you use are likely to depend on the nature of the problems you are looking to solve and how you want to work with each other. Don't forget, if the worst comes to the worst, you can still ask for help.