

**University of Leeds
School of Computing
COMP3900 Distributed Systems**

**Artur Oliveira Rodrigues
SID: 200622082**

Coursework 3

All the code of this coursework is available as a public GIT repository at: https://github.com/arturhoo/ds_cw3

Question 0:

The two chosen web services for this coursework were:

1. CDYNE Weather – FREE. This web service provides weather information based upon a US zip. On this particular task, it was used the current temperature in Fahrenheit units of a given zip. Its publisher is CDYNE.
<http://www.xmethods.net/ve2/ViewListing.po?key=425811>
2. Temperature Conversions. This web service provides conversion for temperatures between the units Celsius and Fahrenheit. Relevant for this task is the conversion from Fahrenheit to Celsius.

Question 1:

This new paradigm of communication between two entities represents a more straightforward approach when compared to Java RMI. The way information is produced or implemented is not relevant anymore, but the way these pieces of information are made available through common standards such as XML. In a nutshell, it requires much less work to build a distributed system with JAX-RPC and JAX-WS than with Java RMI.

Question 2:

As discussed in question 1, the platforms, programming languages or operating systems behind the web services used were not important for the implementation on this coursework. When compared to conventional distributed object technology, it would certainly be needed a reasonable amount of knowledge on the platforms behind the objects going through the systems.

Question 3:

It is reasonable to say that the factors that affect the performance of my application are all related to the network or the servers itself, from factors such as network availability, connection speed, geo location to physical power state of the servers.

Provided the freedoms to restructure the servers layout, it would make sense to bring all the service providers and consumers into the same network. No breaks are good hardware additions in order to guarantee server availability. Further improvements could be replication and load balancing.