**Service Oriented Architecture**

Joshua Harrison

838252

Febuary 2018

**Discussion Points**

**Access control**

When a logged in user requests a resource, should a database call be made to validate the user has access to that resource (e.g. they’ve enrolled?)

OR on login should a

**Introduction**

For this project a MooC service in which universities can provide course material as both standalone units and as part of a wider qualification has been designed. The platform *‘MightyMooC’* has been designed with fundamental service orientated design principles. Erl(2010) defined eight core Service Oriented Design Principles as:

* ***Standard Service Contract****: Services must adhere to a standardized shared contract in order to promote standardization of design and ease of adoption.*
* ***Service Loose Coupling –*** *The edges between services should be minimally weighted, with no one service too heavily reliant or any other in order to perform its function.*
* ***Service Abstraction –*** *Service should be coarse grained with details of the internal logic of the abstracted from external consumers.*
* ***Service Reusability –*** *Services should be significantly agnostic so that they are candidates from multiple disparate business processes.*
* ***Service Autonomy –*** *Services should have significant control over their runtime environments and resources required for perform their function. This is in line with containerization and if taken to its logic limit, a microservices architecture.*
* ***Service Statelessness –*** *State information should not be handled by the services runtime in order to enhance resource scalability.*
* ***Service Discoverability –*** *Services should be discoverable through defined discovery mechanisms in order to maximize service reuse and thus avoid rewriting existing service logic*
* ***Service Composability –*** *Individual services can be grouped into ‘compositions’ with other services in a service inventory to solve high level business problems.*

Erl reasons that interoperability is the fundamental concept of Service Orientated Architecture as achieving interoperability leads to demonstrable return on investment (RoI) through the continual application and reuse of services in multiple business processes, both internal and external to the service owners. The eight principles highlighted above serve to drive the overall architecture towards an interoperability. The MightMooC project has been design with these principles in mind, and their application shall be discussed in greater detail in the follow sections.