

Cloud Microservices Orchestration Framework

Overview

This Senior Design Project (SDP) deals with the development of a Microservices Orchestration Framework. Given a Cloud based microservices Enterprise Service Bus (ESB) framework that will include a microservices catalog with automated service registration, your team will develop domain specific language establishing an enterprise integration framework and microservices standard, i.e. The MOSL - Microservices Orchestration Service Language. MOSL would/could facilitate service registration, advertisement of the microservices catalog, a web-based user interface orchestration service and scripting of micro services workflow.

Background

ESB implements a communications system between mutually interacting software applications in this case the applications being microservices. ESBs are designed to provide a uniform need of moving work offering applications to connect to the bus and describe messages based on simple structure or business policies. The advent of microservices and their interaction was an evolution of Service Oriented Architecture (SOA) implementation beyond the initial use of ESBs. With the use of microservices, ESBs have almost been forgotten, left for dead. However, the growing number of microservices, their numerous endpoint configurations (input/output), manual marketplace selection and configuration requirements creates challenges in determining appropriate microservices for specific uses in real-time.

The Project

General - Create a Framework that allows you to orchestrate microservices. This idea merges the concept of cloud microservices with SOA processes implemented through an ESB to create a more flexible microservice infrastructure. The product should leverage an enterprise service bus for microservice registration and orchestration. Services advertise their input and output as well as service availability. The user uses MOSL to query the service catalog for the service they need, and the catalog responds with usable microservice, all communicated across the ESB. The MOSL protocol helps manage the growing number of microservices and their interfaces by creating one domain communications standard and establishing an ESB to use it and communicate on.

First Semester

- Design a domain specific language that allows user to message between and string microservices together using the enterprise service bus.
- Define the features, inputs, and outputs.
- Design a user interface / query engine.
- Design the “wrapper” to manage microservice components and allow user to assign microservices to a specific input or output. The microservice acts as a software agent.

Second Semester

- Build the architecture of MOSL that integrates with the ESB.
- Develop the user interface that can string components together. The user must be able to drag and drop services around to orchestrate a workflow.
- Develop the query engine that allows the user to discover microservices.
- Build the “Wrapper” to manage microservice components

Project Sponsor: AnaVation LLC