# Project Research Document

# Rules Based Auth

X00130180 - Jordan Williams

## Section 1 Detailed Discussion x 1 Page

I was tasked with creating an application that would ultimately enforced rules based authorization/authentication between microservices in the cloud. To do this I will be using Docker to wrap my services, and then deploy them on Kubernetes, but, I will also be using Istio to install a sidecar alongside my pods (services) to monitor/restrict traffic to other pods. Istio allows you to place rules on the different services, such as verifying the scope or claim of a JWT token matches what Istio says it should contain, and if it doesn’t, Istio will restrict access to the pod and you will be denied access.

The main way of enforcing this rules based auth is via JWT tokens that are propagated throughout the application via HTTP headers when sending a request to another service. Istio will check the “Authorization” header and verify that the token corresponds to the JWK I told Istio to cross check against, and if all is good, you will be granted access.

This is an infrastructure project, so the only two services you need are the frontend and utils backend. These two services manage the handling and issuance of the JWT tokens. The JWT token is stored in the frontend until a user is logged out, and once a user is logged in a specific JWT token is granted from Auth0 API for that specific user containing a unique claim depending on the user that logged in.

There is three different users that are used, these are Doctor’s, Admin, and Patient. The backend performs read only operations on the database, and the admin-backend performs write only operations on the database. Only Doctors and Admin can access the admin-backend whilst every other user can access the backend.

I have also extended the current functionality of Istio to add my own rule that allows me to add a rule that enforces that the users with the claim Doctor cannot access the backend or admin-backend services once it is passed the time of 9-5. This service is deployed in the istio-system namespace and is written in Go, as is Istio.

Take the idea you have selected and develop it providing a detailed discussion on the

system functionality and what you propose it should do. Who are the users?

## Section 2 Existing Applications in this domain x 1/2 Page

The only application I found that was remotely similar was the Bookinfo application from Istio which is their version of hello world, but it was also very different. I looked at it in the beginning of my research and thought I could take bits from it to help with my project, but it was too different in the end for me to even try use anything from it.

<https://istio.io/docs/examples/bookinfo/>

## Section 3 Platform, Technologies and Libraries x 1/2 Page

* Python
* Flask
* Go
* Docker
* Kubernetes
* Istio
* JWT
* Auth0
* Shell scripting
* YAML
* Grafana
* Prometheus
* Google cloud (google kubernetes engine)
* Minikube
* Docker-compose
* N-tier architecture

## Section 4 The risks x 1 Page

The main risks throughout the project were if I could get it complete in time as the project was pretty complex and encompassed a lot of technologies I had no prior knowledge of. It is also a cutting edge project, as there is very little projects out there like it. These tools I have been using have only been in existence as of recently so there is not a lot of tutorials (pretty much none) regarding the features of Istio, Auth0 etc. that I wanted to take a look at.

I spent a lot of time having to trawl through open github issues searching for bug fixes etc. as these technologies are still in their infancy and have some known issues, albeit small, they can take you days to figure out. I had one of these issues with a mutual tls issue when I was using Istio that I couldn’t figure out for a day or two and I was constantly thinking about what it could be, but I later figured it out and fixed the bug in my YAML file that corrected the issue.

I began to really enjoy doing the project and seeing the progress I was making. It really made me enjoy programming again, as before I felt a little burned out as I was creating a lot of boring web apps etc for some of my modules.

I kind of knew I would get the project complete from January as I was putting a lot of thought into it and how some of my features would be completed. I was constantly planning my next feature and had an idea in my head of how it was going to be implemented, so every night after work I would go home and implement it.

I just did it all one feature at a time, like a bunch of constant little sprints, and luckily I never got stuck on a feature for more than a day or two. My mentor was also extremely helpful from IBM, he would give me clear instructions of features that need to be implemented and a timeframe of where we need to be in 6 weeks etc. I never really asked for his assistance when I was stuck, as I felt stackoverflow was a good resource to use and faster as he was a busy guy and I didn’t want to bother him, but we constantly talked throughout the weeks as I was updating him. The most important thing was he had a clear idea in his head of where the app should be come the end of the year and you could see some of the other mentors didn’t and that kind of showcased during the meetings.