# Application 1: Switch

### Overview:

It’s an elearning platform for internal users of our company. We have a company that exists in 3 or 4 countries. It should be performant as possible but it’s not necessary to cater to individual regions. The users can access on mobile (iOS/Android) or Website. There is also a CMS where backend users can upload all the elearning content and do user management. The eLearning platform also has videos where users can learn from. So we need a way that if we upload 1 video it can transcode into various lower resolutions. Users login through our company AD which is hosted in Azure Cloud. It can be accessed through the internet and also within our VPN. Ideally users don’t hit the internet at all when they are using our VPN.

### Application Details:

The application consists of a mobile app and a webapp for users. It’ll be written in React, Angular, and nodejs.

The devs are familiar with Postgresql for the database.

You only have enough IP for a single /24 cidr range for which you need to have Dev, QA, and Prod servers.

We have a constant load of users and we don’t mind having servers always running.

It’s about 30,000 users reading the same things regularly daily since new lessons are released weekly.

# Application 2: Navigator

### Overview:

Navigator is an ML Prediction Platform for predicting the prices of oil and gas in the future based on current and historical values as well as projected production values. Users have a series of Excel files they upload that has projected values for oil production, proposed sales, assuming currency conversions etc. The data needs to be extracted from these excels. Once that is done the users will also upload another excel file for constraints that will give the model some guide rails. You can assume the ML model is pretrained already. The users want to be able to upload the projections and the constraints and then have the model create an output to the screen in a dashboard. They also want to make sure that when they upload the excel it’s validated before the model is run so that they can fix it and re-upload. Afterwards the users should be able to tweak a few values from the projections and rerun the projections. This has to be done directly in the UI, not in excel anymore. Once the user is happy with the tweaks and model outputs, they want to save all that, so next time they have their tweaked input, constraints, and model output. Or at least be able to rerun the values.

Authentication and software development stack is up to you to decide.

### Application Details:

Application is just a webapp and dashboarding.

The Data Science team has graciously created a docker container that has the ML Model and an API written in Flask to interface with the model.

The model takes 20 mins to run to generate an output from the files. Verifying the files may take up to 5 mins.

Technology stack is up to you.

There only about 20 users who mainly use it at the end of every quarter.

# Application 3: Talent

### Overview:

Talent is an HR Platform to access various talented employees within the company. On a monthly basis all of the data from the HR Department on the top talented employees is collated into their on-premise db. The Data Science team has an algorithm to rate an employee based on this data and create a rating based on the company's leadership principles: Leadership, Communication, Technical Knowledge, Innovation. They also use this to match users to the job role. HR would like to see the results of this model one a monthly basis to help figure out potential promotion opportunities and also identify job matches across various divisions.

Security is of the utmost concern. The users and the data should never have internet access. The least amount of users as possible should have access to all the PII information like name, age, salary, etc. Only HR can see these values in Prod. However the rest of the data the Datascience team need. None of the Dev / DS team needs to see the PII information. Authentication is done by On Premise Active Directory.

You also require Dev and Prod setups.

### Application Details:

Data science team is using a platform called Databricks to train the model and do their development.

Application Team has decided on a simple React, NodeJS, SQL stack but wants to do microservices.

All of the monthly data needs to be run monthly, one step is training the model, the other step is running the results and creating an output. Aka it’s batch processed and the output it used for the dashboarding application.

HR uses this on a monthly basis but especially in Q4 to plan for next years promotions and bonuses.

\*\*Assume Dev and Prod splits for all of these, no CIDR limitation otherwise\*\*