

# Architectural Decision Record

Group 26 -

Christopher Nixon  
Davy Nolan  
Hannah Mahon  
Christopher Staunton  
John O'Doherty  
Paul O'Callaghan

This file contains information on the major technological, architectural and platform decisions made in our project. For the continuous integration continous deployment pipeline we have decided to use a range of technology including; travis CI, docker, kubernetes and Jest. For the front-end of our project we have chosen react.js and mapbox api. The reasons for such choices are outlined throughout the report.

<b>Architectural Decision:</b>	Use Travis CI as the main Continuous Integration platform.
<b>Issue or Problem Statement:</b>	There are several different CI solutions to choose from
<b>Assumptions:</b>	The code is placed in a version control repository, such as github
<b>Motivation:</b>	Support the Continuous Integration of the application.
<b>Alternatives:</b>	Jenkins, GitLab, Buildbot
<b>Justification:</b>	We chose TravisCI because it is cloud based and easy to run. It is ideal for our team project as it is quick to set up.
<b>References:</b>	<a href="https://docs.travis-ci.com/">https://docs.travis-ci.com/</a>

<b>Architectural Decision:</b>	Use GitHub as the main Version-control platform.
<b>Issue or Problem Statement:</b>	There are several different version-control solutions to choose from.
<b>Motivation:</b>	Support the Continuous Integration of the application.
<b>Alternatives:</b>	TortoiseSVN, Apache Subversion, Gitlab, Bitbucket
<b>Justification:</b>	We chose GitHub because it is accessible, open source and can be linked with TravisCI easily.
<b>References:</b>	<a href="https://github.com/">https://github.com/</a>

<b>Architectural Decision:</b>	Use Docker for Containerisation
<b>Issue or Problem Statement:</b>	We need a way to continuously deploy our application. There are different containerisation platforms.
<b>Motivation:</b>	Containerizing our software allows us to deploy it easily to all of the major cloud vendors.
<b>Alternatives:</b>	Kubernetes, Virtual Box, Vagrant
<b>Justification:</b>	We chose docker for containerization because it is the most popular open-source containerization software available, this means a large support community which is helpful if we run into problems.
<b>References:</b>	<a href="https://docs.docker.com/">https://docs.docker.com/</a>

<b>Architectural Decision:</b>	Use IBM cloud to host our application
<b>Issue or Problem Statement:</b>	We needed somewhere to host our application so that it's front-end is publicly accessible.
<b>Motivation:</b>	Having a publicly available front-end will allow us to see the end-result of our pipeline.
<b>Alternatives:</b>	Google Compute Engine
<b>Justification:</b>	We chose IBM cloud as it allows our project to be more scalable, has competitive offerings to all the other major cloud providers and our product owner expressed interest in us using it.
<b>References:</b>	<a href="https://www.ibm.com/cloud">https://www.ibm.com/cloud</a>

<b>Architectural Decision:</b>	Use Jest for our unit testing.
<b>Issue or Problem Statement:</b>	We needed software to help in testing our code and something that would integrate well with our pipeline.
<b>Motivation:</b>	We wanted to be able to test our code under unit testing and to provide some sort of code coverage aspect.
<b>Alternatives:</b>	Selenium
<b>Justification:</b>	We chose Jest as its free, simplistic and it integrates well with node.js and react framework
<b>References:</b>	<a href="https://jestjs.io/">https://jestjs.io/</a>

<b>Architectural Decision:</b>	Use React for our front-end
<b>Issue or Problem Statement:</b>	We needed a somewhere to create a front-end for our project to show that the code has been properly deployed through the pipeline
<b>Motivation:</b>	We wanted to provide some sort of front-end web application to show that our pipeline was working efficiently and properly deploying
<b>Alternatives:</b>	Inferno.js
<b>Justification:</b>	We chose react as a few of the group members have had experience with using it in the past and there is lots of nice libraries and apis that can be easily integrated that we could benefit from
<b>References:</b>	<a href="https://reactjs.org/">https://reactjs.org/</a>

# Architecture Diagram

