Analysis of Alternatives

# OVERVIEW

The aim of this report is to analyse two different methods of implementing an application. One application is required push motion sensor data to a database and the other is required to listen to the database and send emails based on motion sensor data. The Node.js framework is used for development and motion sensing is performed using an Arduino Uno board.

# ALTERNATIVES

## On-Premise Applications

On-Premise applications are installed and run on computers on the premises of the organisation using the software rather than at a remote facility.

## Firebase Functions

Firebase functions are cloud functions that allow developers to run backend code triggered by firebase events. The code is stored on Google’s cloud and runs on managed servers, there is no need to manage or scale private servers.

# CRITERIA

The analysis of alternatives will focus on the following criteria:

## Manageability

* How easy is to configure, modify, deploy and control the system?

## Ease of Programming

* How easy is it to program and develop in the framework?

## Support

* How much documentation, examples, online forums, etc. are available?

## Performance

# 

# RESULTS

The two alternatives were compared through two spikes that were designed to push data from the motion sensor to the firebase and listen for motion and send emails when they occur. Their evaluation is summarised below.

## On-Premise Applications

**Manageability**

As the applications are run on private servers, managing these servers can be time consuming and expensive. It must be ensured that these servers stay up be and are able to handle the load of user traffic in order to keep the applications working.

**Ease of Programming**

The On-Premise applications only require the user know how to code in JavaScript using the NodeJS framework and interact with the firebase. Most web developers have extensive experience in JavaScript and so it should be easy for them to develop on-premise applications.

**Support**

The NodeJS framework and JavaScript language are both very well-documented, information for either is highly accessible. As the user-base is large support community for developing using NodeJS is extensive.

**Performance**

## Firebase Functions

**Manageability**

Firebase provides an incredibly easy to manage system. As it is cloud-based, and server management is handled by Google, scaling occurs without any intervention required from the application developers. Code can be deployed to firebase very simply, by using one command in the command line. In addition, all server configuration, provisioning, and decommissioning is done by Google.

**Ease of Programming**

Firebase functions are written in JavaScript running in the NodeJS environment with the ability to interact with Firebase triggers using the Firebase API. The Firebase API is fairly simple and very well explained, any experienced programmer should not have an issue with it.

**Support**

The website for Firebase provides a vast amount of information and reference on all of its API. Each class and method is documented with examples provided. There are also an increasing number of firebase projects appearing on GitHub that can be used as examples.

**Performance**

# Recommendation

Our recommendation is that Firebase functions are used over On-Premise applications. Firebase functions are much easier to manage, automatically scale and have an abundance of support provided by Google’s documentation. We believe this is worth the trade-off of having to program with the Firebase API instead of pure NodeJS as it is very easy to work with.

Firebase Functions were shown to be vastly superior in:

* Manageability

and on-par with On-Premise applications in:

* Support

Given that On-Premise applications provide almost no benefit over Firebase cloud functions to an experienced developer, we can see no reason to use this method other than if inexperienced developers are unable to learn/use the Firebase API.

On-Premise applications were shown to be marginally superior in:

* Ease of Programming