Utilising Edge Computing in a 3-Tier system between Client Devices and a Data Centre

A dissertation submitted in partial fulfilment of the requirements for the degree of BACHELOR OF ENGINEERING in Computer Science

in

The Queens University of Belfast

by

Connor Dickson

9th May 2017

# Acknowledgements

* Hans
* Automated Intelligence
* Cassio

# Abstract

* 100 word outline of subject matter/findings

Table of Contents

[Acknowledgements 2](#_Toc474236032)

[Abstract 2](#_Toc474236033)

[Introduction and Problem Specification 2](#_Toc474236034)

[System Requirements Specification 3](#_Toc474236035)

[Design 3](#_Toc474236036)

[Architectural Description 3](#_Toc474236037)

[User Interface Design 3](#_Toc474236038)

[Software System Design 3](#_Toc474236039)

[Implementation and Testing 3](#_Toc474236040)

[Choice of implementation languages/dev environments 3](#_Toc474236041)

[Use of software libraries 3](#_Toc474236042)

[Key implementation details 3](#_Toc474236043)

[How each component was implemented 3](#_Toc474236044)

[Discussion of test approach 3](#_Toc474236045)

[System Evaluation and Experimentation 4](#_Toc474236046)

[Conclusion 4](#_Toc474236047)

[References 4](#_Toc474236048)

[Copyright 4](#_Toc474236049)

[Appendices 4](#_Toc474236050)

[User manual 4](#_Toc474236051)

[Test Results 4](#_Toc474236052)

[Minutes of meetings 4](#_Toc474236053)

# Introduction and Problem Specification

Background material which introduces the problem area, context and background.

Identify my problem

Systematically researched and fully analysed the problem

# System Requirements Specification

Precise description of the system developed. Should be updated for the final system delivered. List assumptions made about the problem and any system constraints(RAM?)

Functional and non-functional requirements should be complete, clear, accurate, feasible and objectively verifiable.

Read req’s for list of what can be included in this section

# Design

## Architectural Description

## 

The entry point for the system is the Electron Application which runs on the Client Raspberry Pi

The Edge is comprised of 3 Raspberry Pi’s which form a Docker swarm. The applications run in containers that can be running on any of the Pi’s but the entry point is always the master who directs traffic. Traffic being directed is colour coded.

There are multiple data centres depending on what application is making the request. The WebAPI project is hosted in IIS

## User Interface Design

* Sketches or screenshots

## Software System Design

* Role of each component and the interface between the components
* Clear correlation between design and specs

Design should be linked to requirements and give a critical discussion of key design decisions/styles/patterns. Read more in notes

# Implementation and Testing

## Choice of implementation languages/dev environments

* Electron that utalises nodejs for Client
* Docker/Nodejs for edge node
* WebAPI for Windows Server

## Use of software libraries

* Pocketsphinx for voice recognition
* Nodejs packages
* Testing suite

## Key implementation details

## How each component was implemented

* Client
* Edge
* Data Centre

## Discussion of test approach

* Because of the nature of the code it could prove difficult to test as it has to run in a nodejs server and made actual requests
* Manual Testing
* Unit tests

# System Evaluation and Experimentation

A lot of the emphasis in this project will be here. It could even be split into two sections.

Provide empirical results and draw conclusions.

Describe methodology (use experiment methodology like goals and hypothesis, what’s measured/controlled)

Read more in notes

# Conclusion

General summary of the success of the project with respect to criteria identified in the intro

Discussion of significance of experimental results. Agree with others work?

Strengths and weaknesses

Evaluation of hardware/software

Looking for critical appraisal and significance of contribution in the context of wider work.

# References

* First ref should be URL to code gitlab

# Copyright

* Electron
* Nodejs/ Nodejs libraries
  + Node JS proxy
* Pocketsphinx
* Testing Suites
* Redis/Docker src images

# Appendices

## User manual

## Test Results

## Minutes of meetings