Load Balancing on an IoT Fog

Christopher Gonzales
Sebastian Correa

INTERFACE CONTROL DOCUMENT

REVISION – 1
29 April 2023

INTERFACE CONTROL DOCUMENT
FOR
Load Balancing on an IoT Fog

PREPARED BY:

Author Date

APPROVED BY:

Project Leader Date

John Lusher II, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	2/22/2023	Load Balancing on an IoT Fog		Revision 1
2	12/4/2023	Sebastian Correa		Revision 2
3	12/4/2023	Christopher Gonzales		Revision 3

Table of Contents

List of Tables

List of Figures

1. Overview

The ICD document will detail how the cloud, RPI, and data collection unit interact. In addition, it will list all inputs, outputs, and how each subsystem manages them—first, the IDC documents how inputs from an IoT fog will be transferred between the devices, edge nodes, and server nodes. A description of how each subsystem interacts with the other follows.

2. References and Definitions

2.1. References

Refer to section 2.2 of the Functional System Requirements document.

2.2. Definitions

mA	Milliamp
mW	Milliwatt
MHz	Megahertz (1,000,000 Hz)
TBD	To Be Determined
RPI	Raspberry Pi (4)
V	Volts
A	Amps
AWS	Amazon Web Services
In	Inch
mm	Millimeter
OS	Operating System

3. Physical Interface

3.1. Weight

3.1.1 Edge Node

Component	Weight	Number of Item	Total Weight
RPI 4	TBD	1	TBD
SD Card	TBD	1	TBD
5V, 3A, Type-C Power Supply	TBD	1	TBD
Heat Sink	TBD	1	TBD

Table 1: Edge Node Weight

3.1.2 Server Node

Component	Weight	Number of Item	Total Weight
AWS Cloud Server	TBD	1	TBD

Table 2: Server Node Weight

3.2. Dimensions

3.2.1. Dimension of Edge Node

Component	Length	Width	Height
Raspberry Pi 4	85 mm	56 mm	21 mm
5V, 3A, Type-C Power Supply	TBD	TBD	TBD
Heat Sink	TBD	TBD	TBD

Table 3: Edge Node Dimensions

3.2.2. Dimension of Server Node

Component	Length (in)	Width (in)	Height (in)

Table 4: Server Node Dimensions

3.3. Mounting Locations

The Raspberry Pi edge node will be mounted somewhere easily accessible and visible, such as a desk or a shelf. The RPI should be placed in a secure, flat, stable location. The AWS instance will be accessed via the cloud, so no physical mounting location is required.

4. Thermal Interface

The edge node subsystem will require a heatsink. As the Raspberry Pi will perform lots of computations it will experience high traffic. Therefore, to handle this workload a passive heatsink should suffice. As an RPI automatically has some cooling features it should not need a cold wall heat sink.

5. Electrical Interface

5.1. Primary Input Power

The only item we have that needs a power supply is our Raspberry Pi 4. We need to have a 5V 5A power supply that contains a switch and a micro USB to supply the Pi. We will connect the Pi in a location where the power will not be disrupted to make sure the load-balancing system will not be interrupted.

6. Communications / Device Interface Protocols

6.1. Wireless Communications (WiFi)

The Raspberry Pi 4 has a Wireless Module found on the top left corner of the board, next to its GPIO pins. The Wi-Fi connection needs to first be connected with an ethernet cable to install the OS. Once that's done, we'll be able to connect the pi wherever we would like.

6.2. Host Device

The only Host Device present in this project will be our Raspberry Pi 4 device. It will need a monitor, keyboard, mouse, and power supply to operate.

6.3. Video Interface

We will need an HDMI cable to connect the Raspberry Pi 4 to a monitor.

6.4. Netmaker Network

The Netmaker network lets us connect the Raspberry Pi 4 to the cloud to establish the Pi as a worker node. It will use the Master Node in the cloud as a host for every connection that needs to be made over the network.

Load Balancing on an IoT Fog

Christopher Gonzales
Sebastian Correa

MILESTONES AND VALIDATION PLAN

REVISION – 1
22 February 2023

403 Milestones:

Work	End Date	Owner	Status	Completion Date
ConOps	2/13	All	Done	2/13/2023
Functional System Requirements	2/22	All	Done	2/22/2023
Interface Control Document	2/22	All	Done	2/22/2023
Amazon Web Services account made	2/22	Sebastian	Done	2/20/2023
Python Refresher	2/22	Christopher	Done	2/22/2023
Project Parts Ordered	2/23	All	Done	2/23/2023
RPI Running Smoothly	2/26	Christopher	Done	2/26/2023
Midterm Presentation	3/1	All	Done	3/1/2023
Install K3s support on cloud	3/3	Sebastian	Done	3/1/2023
Install K3s on RPI	3/3	Christopher	Done	3/1/2023
Prepare Load-Balancing Algorithm	3/7	All	Done	3/7/2023
Install the Load-balancing Algorithm on RPI	3/8	Christopher	Done	3/8/2023
RPI interacts with the cloud	3/10	Sebastian	Done	3/10/2023
RPI Takes IoT Input	3/12	Christopher	Done	3/12/2023
Prepare for Progress Update 1	3/18	All	Done	3/18/2023
Progress Update 1	3/29	All	Done	3/22/2023
Configure RPI to direct traffic to the load-balancer	3/31	Christopher	Done	3/30/2023
Test the System	4/1	All	Done	4/15/2023
Monitor the System	4/1	All	Done	3/31/2023
Prepare Final Presentation	4/9	All	Done	4/9/2023
Final Presentation	4/12	All	Done	4/12/2023

Optimize the system	4/29	All	Done	3/28/2023
Project Demo	4/29	All	Done	4/29/2023

Table 1: 403 Milestones

403 Validation Plan:

Task	Specification	Result	Owner
Power Supply for Edge Device	5V 5A	Completed	Christopher
Cloud Response Time	<500ms	Completed	Sebastian
Edge Device Runtime	<500ms	Completed	Christopher
Edge and Cloud Transmission Time	<500ms	Completed	All
Number of Test Cases	50	Completed	All
Number of applications to run	25	Completed	All

Table 2: 403 Validation Plan

404 Milestones:

Work	End Date	Owner	Status	Completion Date
Check pi condition	8/24	Christopher	Complete	8/24
Choose a new cloud service	8/24	Sebastian	Complete	8/24
Convert all virtual machines to Microsoft Azure	9/7	Sebastian	Complete	9/11
Build website	9/7	Christopher	Complete	9/7
Install K3s on VMs	9/14	Sebastian	Complete	9/14
Test website	9/14	Christopher	Complete	9/11
Add more features to the website	9/21	Christopher	Complete	9/18
Create connections between VMs	9/21	Sebastian	Complete	9/21
Connect the Pi with the Cloud	9/28	Sebastian	Complete	10/1
Convert Pi connection into worker node	10/5	Sebastian	Complete	10/15
Configure containerization in Pi	10/5	Christopher	Complete	10/5
Finalize Integration	10/12	Christopher	Complete	10/15
Test Traffic transmission	10/19	Sebastian	Complete	11/5
Demonstrate movement of loads	10/26	Sebastian	Complete	11/20
Test Fault Tolerance	11/2	Christopher	Complete	11/2
Bug fix the VM’s	11/2	Sebastian	Complete	11/20
Configure cloud validation	11/9	Sebastian	Complete	11/9
Configure pi validation	11/9	Christopher	Complete	11/9
Finalize validation	11/23	Christopher	Complete	11/23

Table 3: 404 Milestones (Execution Plan)

404 Validation Plan:

Task	Specification	Summary	Result	Owner
Cloud Response Time	<500ms	Amount of time it takes the Cloud to respond to the Load Balancer	60-62 ms	Sebastian
Edge Device Runtime	<500ms	Amount of time it takes the Edge Device to respond to Load Balancer	Avg 72.7 ms	Christopher
Edge and Cloud Transmission Time	<500ms	Amount of time it takes for the edge device and cloud to respond to one another	60-62 ms	Sebastian
Local PC and Cloud Transmission Time	<500ms	Amount of time it takes for local computer(dashboard) and cloud to respond to one another	Avg 158 ms	Sebastian
Reading Traffic	<500ms	Amount of time it takes the K3s cluster to read the incoming traffic	68.6 ms	Sebastian
Minimum Number of Test Cases	50	50 test users, or traffic data, being sent to our system for testing	316 requests in 30.05s	Christopher
Minimum Number of applications to run	1	Our application we're going to use for testing	1	Christopher
Load Balancing Test	40%-60%	Ensure that the load is distributed equally between rpi website container and VM container	40%-60% (Odd # of Pods) 50%-50% (Even # of Pods)	Sebastian
Failover Test	<500 ms	Shut down one edge node and ensure traffic is redirected to working node within seconds	Min: 5 min Max: 12 min	Christopher

Table 4: 404 Validation Plan