Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Summer 2021

Raytheon Team
WiDROS

Daniel Tam
Joell Soriano
Joshua Pearson
Renato Cruel Amado
Tiffany Frias

Revision History

Revision	Date	Author(s)	Description
0.1	08.12.2021	RCA	document creation
0.2	08.13.2021	DT, JS, JP, RCA, TF	complete draft

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

WiDROS is a system that allows users to see real-world coverage range of all wireless network present at a building in the University of Texas at Arlington, through a virtual reality environment. Users of WiDROS software will be able to see and experience an accurate virtual reality environment of the building. They will also be able to see information and coverage range of all wireless network present on the building through this virtual reality environment.

1.1 PURPOSE

The system accepts real-world wireless signal data collected by a Raspberry Pi placed on a drone. Then, the signal information is processed and packaged into visual domes that indicate the coverage range of each network. Our product will map these visual domes on to an accurate virtual reality environment of a building in the University of Texas at Arlington. The product will allow users use a virtual reality goggles to experience the virtual reality environment of the building along with the network coverage domes. It can be used to accurately view the coverage of each wireless signal on the building and optimize router placement to ensure better coverage.

1.2 SCOPE

The scope of this project will be limited to creating a virtual reality environment of only a building at the University of Texas at Arlington, and to only processed and display data retrieved from the Raspberry Pi. Any other locations or signals will not be addressed in this project, but could be considered for future projects.

Note: Due to COVID-19 and social distancing measures, the environments have been changed to locations in or around our homes. However, since we may be returning to campus in the near future,we are leaving the original environments listed above.

1.3 KEY CONSTRAINTS

There are about a dozen documented constraints for WiDROS, but here we will list a few of the more significant ones. We believe this will help the reader understand certain architectural design choices.

- The application will pull the Raspberry Pi data from a data file instead of a AWS database.
- The application will have an interface that will allow users to see and select information about an environment using data from the Raspberry Pi.
- The Raspberry Pi will be able to push the RF signals to a data file.

2 SYSTEM OVERVIEW

This section provides the reader with an overview of team WiDROS' Virtual Reality portion. The system produces a virtual environment that makes wireless signals visible to the user occupying the environment. The environment can be rendered from data pulled from a data file pushed there by the Raspberry Pi signal scanner. The principle parts/components of this system are the VR Headset and a data file.

Figure 1: WiDROS architectural layer diagram

2.1 RASPBERRY PI DESCRIPTION

The Raspberry Pi has units that uses sensors to collect data from the real world, for this project, WiFi signals. All of these signals are recorded and sent to a data file. In order for the VR to communicate with this layer, the data file will have to be extracted from the Raspberry Pi and then uploaded into the Unity Environment.

2.2 DATA FILE

The Raspberry Pi shall export data onto a data file, that data file will then be exported into the Unity Environment. Data file will be a JSON file.

2.3 VR DESCRIPTION

The Virtual Reality environment is in charge of interpreting the data from the data file and rendering an overlay to the VR Headset for the user to see. The simulation consists of the following subsystems: an interface to connect with data file, a controller to receive and process information, user input from the VR headset, an overlay to present the data acquired by the Raspberry Pi in a way that is easily understood by the user and is visually pleasing, models that appear in the overlay, and a display for the user to see.

3 Subsystem Definitions & Data Flow

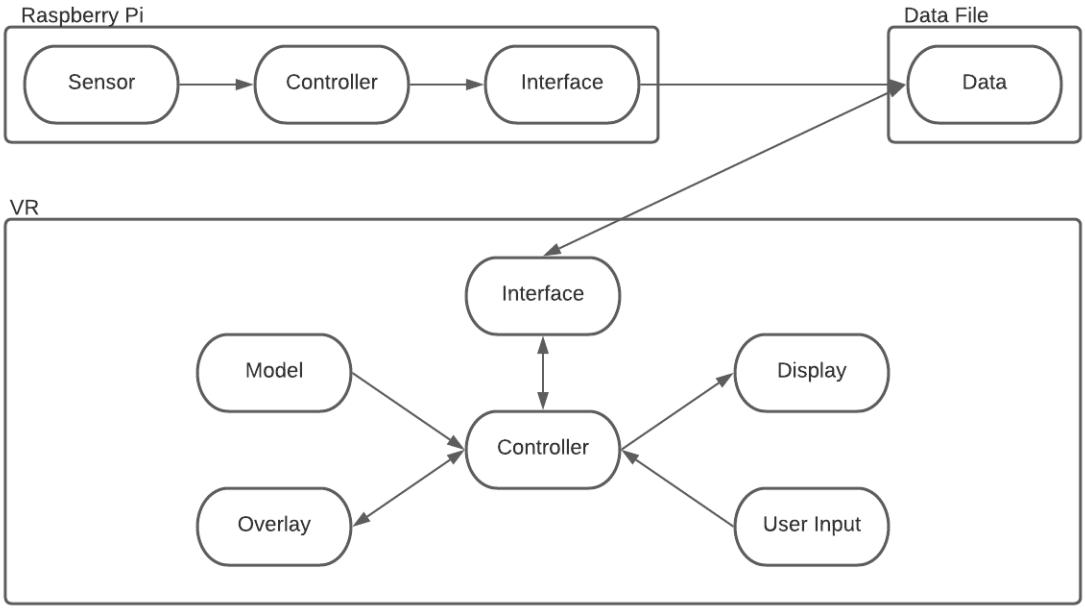

Figure 2: WiDROS Data Flow Diagram

4 RASPBERRY PI SUBSYSTEM

The Raspberry Pi will gather data from the user environment through its sensors, process the data, and transmit the data to a data file.

4.1 SENSORS

The Raspberry Pi has several sensors that will detect the following properties: WiFi SSID, WiFi Strength and global position.

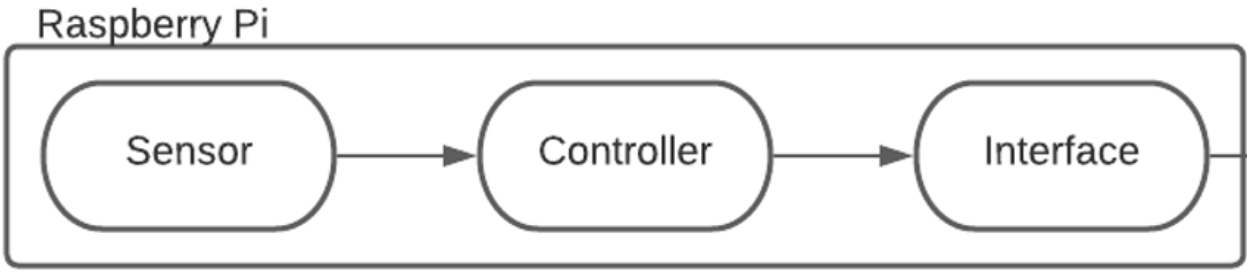

Figure 3: Raspberry Pi subsystem description diagram

4.1.1 ASSUMPTIONS

We will receive the data that we need to create our simulation, such as GPS data, signal strength, and time.

4.1.2 RESPONSIBILITIES

The sensors have the responsibility of collecting the necessary data in a reliable fashion.

4.1.3 SUBSYSTEM INTERFACES

Table 2: Sensor interfaces

ID	Description	Inputs	Outputs
#01	Wired Connection	Sensor Data	Sensor Data

4.2 CONTROLLER

The controller component is the main set of scripts on the Raspberry Pi that will process the sensor data and post it to the data file. The Pi contains multiple python scripts that are able to read, process, and post the data collected by the sensors so that it will be able to be sent to a data file through the interface.

4.2.1 ASSUMPTIONS

The controller will receive data from the hardwired sensors.

4.2.2 RESPONSIBILITIES

The controller has the responsibility of converting the data received into a JSON file and sending it tothe interface.

4.2.3 SUBSYSTEM INTERFACES

Table 3: Controller interfaces

ID	Description	Inputs	Outputs
#01	Wired Connection	Sensor Data	JSON Data

4.3 INTERFACE

The interface receives the JSON file that contains the data gathered by the sensors, and sends this extracts data file.

4.3.1 ASSUMPTIONS

A JSON file will be received.

4.3.2 RESPONSIBILITIES

The interface is responsible for extracting the JSON file.

4.3.3 SUBSYSTEM INTERFACES

Table 4: Interface Interfaces

ID	Description	Inputs	Outputs
#01	Ethernet Connection	JSON Data	JSON Data

5 Data File Subsystem

The Data File subsystem will host all the data processed by the Raspberry Pi sensors. The data file will also contain data for heavier calculations that may be necessary to create the AR simulation and triangulation of WiFi domes.

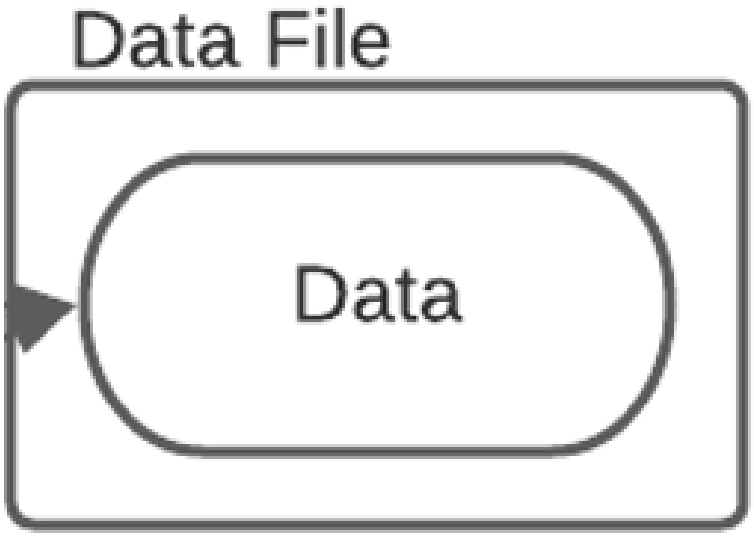

Figure 4: Overview of Data File subsystem

5.1 DATA

5.1.1 Assumptions

A stable and uninterrupted internet connection for the Raspberry Pi to push data to the data file.

5.1.2 Responsibilities

The data file is responsible for the information needed to successfully load the WiFi domes within the Unity Environment.

5.1.3 Subsystem Interfaces

Table 5: IoT Core Interfaces

ID	Description	Inputs	Outputs
#01	Incoming Sensorium JSON Data	JSON-formatted input	JSON Dictionary

6 VR SUBSYSTEM

The VR (Virtual Reality) Simulation subsystem contains an API to interact with and retrieve data from the data file, a controller to handle the information from the data file, models of the VR simulation, an overlay that will visualize fetched data, a user interface, and a layer for user input to interact with display.

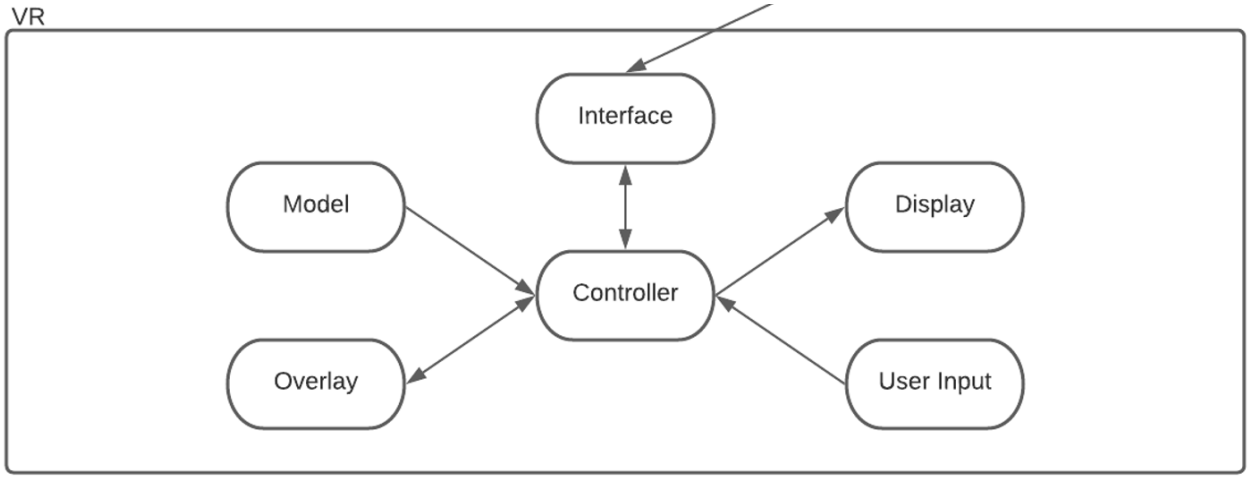

Figure 5: Overview of the VR layer

6.1 INTERFACE

The interface will specifically take in data from the data file and send it to the controller subsystem within Unity.

6.1.1 ASSUMPTIONS

Data will be received in JSON format.

6.1.2 RESPONSIBILITIES

The responsibilities of this subsystem are to interface the simulation with the data extracted from the Raspberry Pi so that the subsystems can receive the data needed to do their function.

6.1.3 SUBSYSTEM INTERFACES

Table 6: Interface Interfaces

ID	Description	Inputs	Outputs
#01	Internet Connection	Data File fetched from Sensor	Processed Data

6.2 CONTROLLER

The controller component will be the Unity graphics engine. This component will handle all of the graphics calculations and will combine the model and the overlay to create the display as well as handle any sort of user input.

6.2.1 ASSUMPTIONS

The description of the controller is based on the assumption that the data required to render the graphics are injected into the graphics engine and that a stable Internet connection was available if running in real-time, or a local copy of the data is stored if not running in real-time.

6.2.2 RESPONSIBILITIES

The controller component is responsible for rendering the 3D model of the environment and the graphical data point clouds. The Unity engine will process the data points received from the data file and render the model on correct GPS location. It will also map the GPS marker to the 3D model of the building on campus.

6.2.3 SUBSYSTEM INTERFACES

Table 7: Controller Interfaces

ID	Description	Inputs	Outputs
#01	Data File	Raspberry Pi JSON formatted Data	Sensor Data to Display
#02	Data Stored Locally	.obj file	3D WiFi Domes

6.3 MODEL

The model component will contain all of the models and other Unity assets that will make up the scene for the simulation. This will include the 3D models generated by Mapbox as well as any other 3D models necessary for the simulation. It is not expected to change often.

6.3.1 ASSUMPTIONS

It is assumed that the Model subsystem will have the necessary models stored locally on the machine in order to create the simulation.

6.3.2 RESPONSIBILITIES

The Model subsystem has the responsibility of containing and rendering the overall environment before any overlays are added.

6.3.3 SUBSYSTEM INTERFACES

Table 8: Model Interfaces

ID	Description	Inputs	Outputs
#01	Local Storage	.obj files	3D Model

6.4 OVERLAY

The overlay component will consist of the wireless dome models and the display of other data from the sensors. While the wireless data will be displayed in either point clouds, some sort of 3D map of wireless signals, or a sort of sphere that dynamically re-sizes based on intensity, all of the other data will be displayed in a heads up display that the user will see and can toggle on and off.

6.4.1 Assumptions

It is assumed that the Overlay subsystem will have received the data necessary to render the wireless signals before it is requested to render the signals.

6.4.2 Responsibilities

The Overlay subsystem will have the responsibility of rendering the wireless signals and other relevant data to the environment and userâs heads up display.

6.4.3 Subsystem Interfaces

Table 9: Overlay Interfaces

ID	Description	Inputs	Outputs
#01	Server	Raspberry Pi JSON formatted Data	WiFi Domes and Signal Meta Data

6.5 DISPLAY

The display will be the combined product of what the user will see. This will be the scene that the user moves through that has the combined overlay with the models of the environment.

6.5.1 Assumptions

It is assumed that the Display subsystem will have received the models from the Model subsystem and overlays from the Overlay subsystem so that they can be shown to the user.

6.5.2 Responsibilities

The Display subsystem has the responsibility of displaying the models from the Model subsystem, as well as the overlay from the Overlay subsystem.

6.5.3 Subsystem Interfaces

Table 10: Display Interfaces

ID	Description	Inputs	Outputs
#01	Unity Graphics Engine	Building 3D Model, WiFi Domes, Signal Meta Data	VR Environment

6.6 USER INPUT

The user input component will consist of the VR controllers and headset that the user will be using to interact with the environment. The user will be able to give directions to the controller to move the view and the player-model throughout the environment.

6.6.1 Assumptions

The application will be loaded onto the VR and will be successfully receiving sensor data.

6.6.2 RESPONSIBILITIES

The User Input layer is responsible for receiving, processing, and forwarding input from the user to the controller layer to be executed in the simulation.

6.6.3 SUBSYSTEM INTERFACES

Table 11: User Input Interfaces

ID	Description	Inputs	Outputs
#01	VR Headset	User activity and movement	Executed user input in VR Environment

References