NAU NORTHERN ARIZONA UNIVERSITY

College of Engineering, Informatics, and Applied Sciences

# Site Weather And Power Recorder

Team: Dylan Woolley, Jabril Gray, Randy Duerinck, Vidal Martinez

Client: General Dynamics Mission Systems

Department: Computer Science, Northern Arizona University, Flagstaff, AZ 86011

## Motivation

The United States Coast Guard (USCG) is responsible for 296,000 nautical miles of sea, and this is a significant job that isn't done alone. Our client, General Dynamics Mission Systems (GDMS), is responsible for working with the USCG on the Rescue21 system for assistance with search and rescue (SAR) missions. The Rescue21 system is a system of antenna towers, each tower called a Remote Fixed Facility (RFF). The RFFs are placed strategically off the coast of the Continental US, major rivers, lakes, and islands (Hawaii, Puerto Rico, and Guam). Each RFF site is equipped with directional finding very high-frequency radios that are used to pick up distress signals, the direction of the signal, the strength of the signal, and whether or not it is a hoax call.

GDMS is content with the Rescue21 system; however, they want some additional features to help them maintain their system. There are two main features that they need: the ability to record the power levels of the antennas at an RFF and the ability to record the weather information at the RFF sites. Knowing this information will help GDMS with determining the cause of Radio Frequency (RF) interference. RF interference causes static while communicating over the VHF signal. A 2020 Electrical Engineering Capstone team created the Site Weather And Power Recorder (SWAPR) device prototype. The project successfully connects to a rain sensor, VHF transceiver, VHF radio, temperature and humidity sensors, and a wind sensor. It uses these tools to record weather and power data to the SWAPR device, however GDMS contacted Northern Arizona University to build upon this prototype with a web based system which solves the following problems.

**Problems**

- RFF site power and weather data is inaccessible remotely
  - Recorded weather and power data is only viewable in a raw text format
- No remote capability for monitoring the operational status of an RFF in the event an RFF is damaged or malfunctioning
  - Rain can cause the VHF signal to fade, interrupting received distress signals at RFFs
  - Severe storms can cause damage to radio equipment at RFFs, disabling data recording capabilities

## Technologies

- Built and used in a Windows Environment
- Developed in the C# programming language
- Used built-in .NET 5 libraries
- Developed using Microsoft Visual Studio Integrated Development Environment (IDE)
- Website application built using the Blazor Framework
- MySQL used to store SWAPR data
- Amazon Web Services (AWS) hosts the relational database on a remote server

**Frontend:**

- Blazor Framework
- JavaScript
- C#
- HTML5

**Backend:**

- MySQL
- C#
- AWS
- Amazon Elastic Compute Cloud (EC2)

## Key Features

- Create data imitating the SWAPR device's output
- Take data from a SWAPR device and send it securely off-site
- Store and serve SWAPR data in a secure manner
- Establish a secure website environment with authentication
- Create a list, map, and historical views of the SWAPR devices in the network
- Create a way to notify operator when there is a problem with an RFF site
- Export data from the database as a .csv file
- Emulate the entire SWAPR device network for stress testing

## Challenges

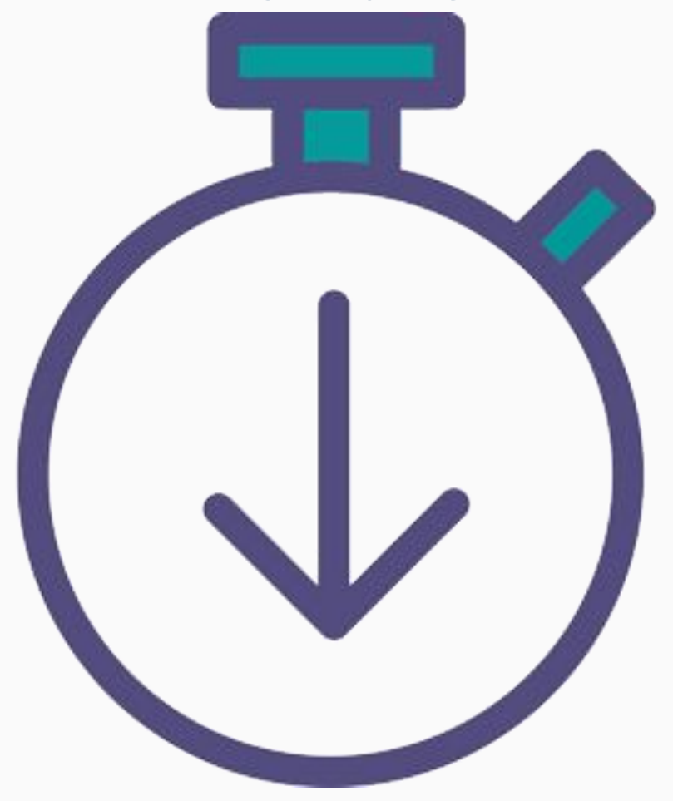
- Issues with Amazon Web Services (AWS) accessibility
  - **Solution:** Our team temporarily paid for AWS until issues were resolved
- Website Summary map view difficulties in implementation
  - It has been challenging to use standard library tools in a Blazor environment
  - **Solution:** Found alternative to standard Blazor library tools
- Website summary list view difficulties in implementation
  - Issues with updating the draw area for all device panels to fit
  - **Solution:** We split up devices and used pagination

## Solution Overview

Our team has created a secure web application capable of collecting, storing, analyzing, and displaying data collected by the SWAPR devices in a clear and efficient manner. Our project is divided into five different subsystems which work together to create our solution. That is: the Simulator, Reader, Database, Website, and Orchestra subsystems. We have chosen these five subsystems to solve the known problems for the original SWAPR device project. The simulator is used to generate data identical to the data output from a SWAPR device. Our team included this to simplify implementing and testing the functionality of a physical SWAPR device. The reader is used to transport the generated data from the simulator to the database. This solves the problem of SWAPR data only being accessible at the site. The Website will use the database to retrieve weather and power data regularly in real-time and create graphical views. These views present the data in a more useful and informative fashion compared to plain text. The website also informs users of critical events at RFFs. In the occurrence of a critical event, users will know the location and the latest recorded data. The orchestra is used to simulate multiple SWAPR device connections, reproducing a real-world implementation of the system.

## Outcomes

Less RFF Downtime



Increased Technician Safety

More Money Saved

## Architecture

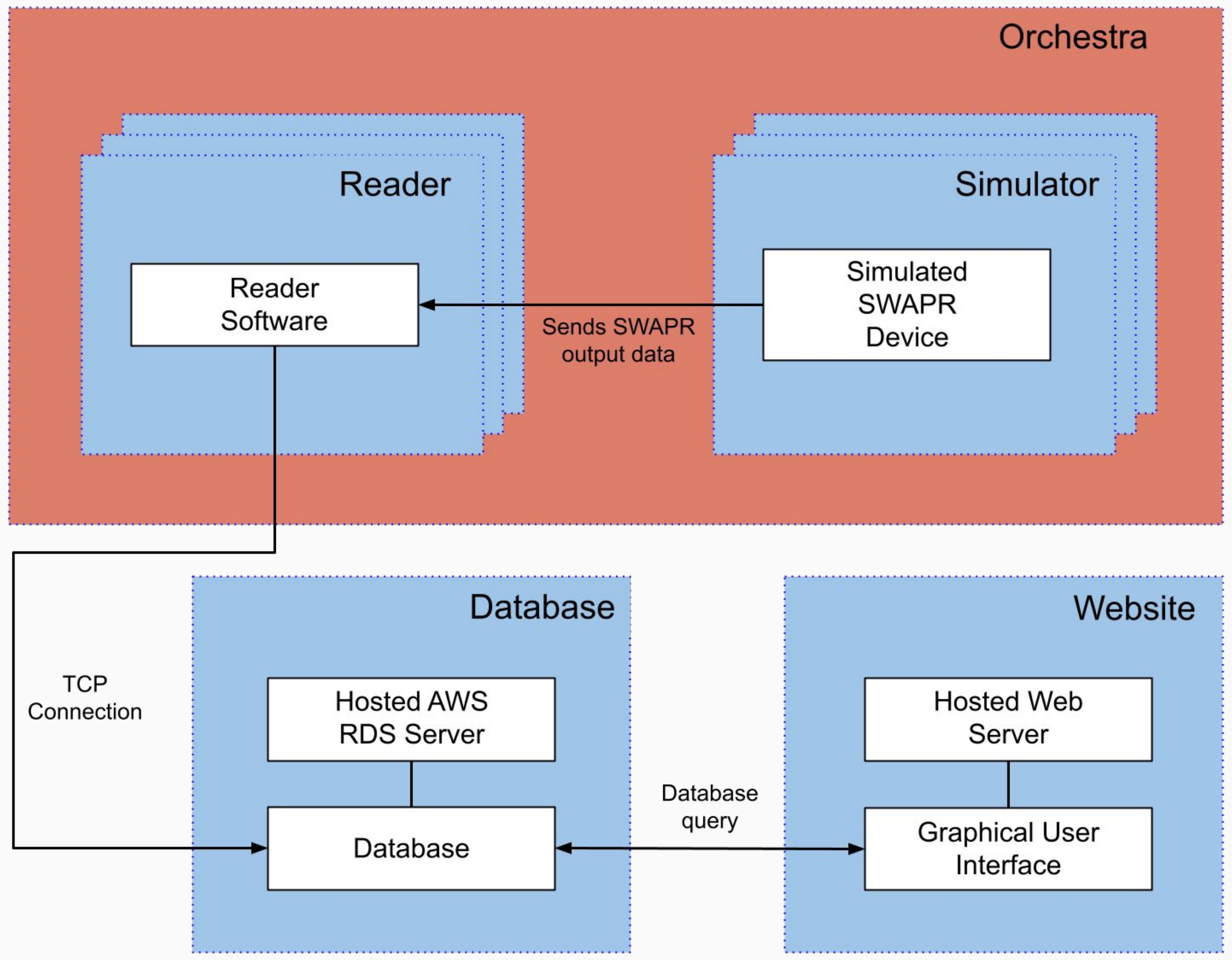


Diagram of system components and communication workflow

## Testing

**Unit Testing**

Unit testing is when small pieces of a program are tested individually to see if they are functioning as planned. Unit testing will be extensive for the Website Subsystem as it is essential the data is produced properly and in the intended format. The Simulator and Reader Subsystems are unnecessary to provide unit tests for.

**Hybrid Integration Testing**

Hybrid integration testing is an approach used to test a system of modules as a whole by testing the communication between each connected module. The test is structured in three layers: the main layer, top layer, and bottom layer. The approach utilizes two other approaches, the top-down and bottom-up approach. The top layer represents the top-down approach, testing from the highest level and down in the system, and the bottom layer represents the bottom-up approach, testing from the lowest level and up in the system. The main layer is the central component for communication in the system.

**Testing Simulator**

- Generating Data
- Connection and sending data via virtualized COM port

**Testing AWS/ Database**

- Check for Data Accuracy
- AWS Lambda Function Access SQS
- Population of the Notifications Table and SWAPR tables

**Testing Reader**

- Connecting and receiving data via virtualized COM port
- Converting data to JSON
- Send formatted data to AWS Message Queue Service

**Testing Websites**

- Checking Notifications
- Accurate location of all RFF sites in the Map View
- Select any SWAPR and data ranges for Historical View
- View of all the RFF sites with pagination for List View