

**ECE445**

**SENIOR DESIGN LABORATORY**

**DESIGN DOCUMENT**

---

**Automated Microwave Scatterometer  
and its digital twin**

---

**Team #30**

Yurong Wang [yurongw2]

Keyi Jin [keyijin2]

Jianing Xiao [xiao36]

**Advisor: Shurun Tan**

**April 5, 2026**

## contents

1.1 Problem Statement .....	1
1.2 Solution Overview & Visual Aid.....	2
1.3 High-Level Requirements List.....	2
2. Design.....	3
2.1 Block Diagram .....	3
2.2 Subsystem Description .....	3
2.2.1 Scatterometer Hardware Subsystem .....	4
2.2.2 Interactive Visualization Subsystem.....	6
2.3 Remote control and wireless communication.....	8
2.3.1 receive/ send data.....	10
2.3.2 receive and send commands.....	11
2.4 built-in camera .....	11
2.7 Tolerance Analysis.....	12
3. Cost.....	13
4. Schedule .....	13
5. Ethics and Safety.....	14
5.1 Ethics .....	14
5.2 Safety .....	15
References.....	16

# 1. Introduction

## 1.1 Problem Statement

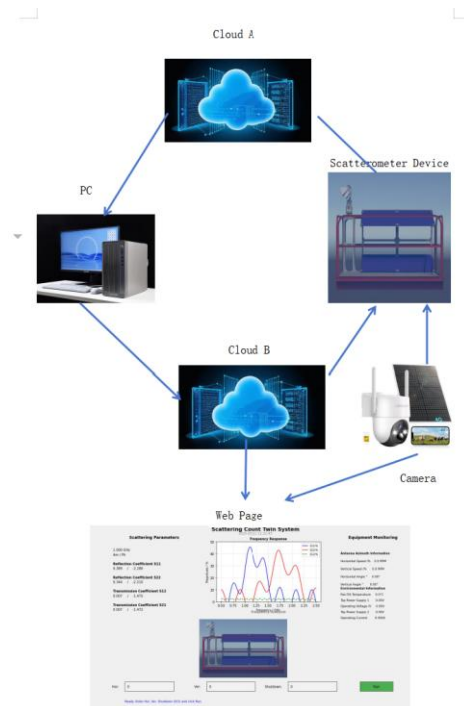


Figure 1: Visual Aid of the whole system

Traditional microwave scatterometer systems face significant challenges in remote operation, real-time monitoring, and experimental repeatability. Existing systems typically require on-site personnel to manually adjust antenna angles (both vertical and horizontal orientations), execute measurement sequences, and record data. This manual dependency introduces several limitations: (1) human errors in angle positioning affect measurement accuracy; (2) researchers cannot operate the instrument from remote locations; (3) real-time visualization of instrument status is unavailable; and (4) experimental repeatability suffers from inconsistent manual operations. Furthermore, current scatterometer systems lack integration with digital twin technology that could provide real-time 3D visualization, remote command execution, and simulation capabilities. The absence of automated angle control (VRS for vertical rotation, HRS for horizontal rotation) limits the efficiency of systematic measurement sweeps across different angles. There is also no existing solution that enables bidirectional file-based communication between a remote server and the scatterometer client for command delivery and data retrieval. The Functional Digital Twin for Automated Microwave Scatterometer addresses these issues by: (1) enabling remote VRS/HRS angle control via a web-based interface; (2) implementing automated command file download and execution on the client-side instrument; (3) providing real-time data upload and visualization; (4) establishing a dual-cloud architecture using the Render platform for reliable bidirectional communication; and (5) incorporating a camera for instrument status visualization. This system will significantly improve experimental efficiency, reduce operational costs, and enable remote scientific research.

## 1.2 Solution Overview & Visual Aid

The Functional Digital Twin system adopts a client-server architecture. The scatterometer (Client) is equipped with a built-in computer running Python scripts that automatically monitor a cloud-based Render platform for new command files. When a user inputs VRS (vertical rotation angle) and HRS (horizontal rotation angle) commands through the local web interface, these commands are packaged into a file and uploaded to Render Platform B. The client computer detects this file change, downloads it, parses the VRS/HRS values, and drives the scatterometer's motors to rotate to the specified angles. After executing the measurement, the client automatically uploads collected S-parameter data to Render Platform A, where the local server retrieves and displays it. A camera mounted on the scatterometer provides real-time visual feedback of the instrument's physical state on the web interface. The total system comprises three major subsystems: Scatterometer Hardware Subsystem (Client), Interactive Visualization Subsystem (Server), and Cloud Communication Subsystem (Render Platform). The following visual aid illustrates the complete architecture.

## 1.3 High-Level Requirements List

1.3 High-Level Requirements List The Functional Digital Twin system shall meet the following requirements:

### (1) Remote Control Requirement:

The server-side web interface shall allow users to input VRS (vertical rotation angle) and HRS (horizontal rotation angle) commands.

The client scatterometer shall automatically detect, download, and execute new commands within 5 seconds of file upload to Render Platform B.

The system shall achieve VRS/HRS angle positioning accuracy of  $\pm 1^\circ$  for vertical and  $\pm 1^\circ$  for horizontal rotations.

### (2) Communication Reliability:

The dual-cloud Render architecture shall maintain bidirectional file transfer with  $\geq 99\%$  success rate over 100 consecutive command-execution cycles.

Command file shall be present on Render Platform B within 1 second of user submission.

Client shall detect command file changes within 3 seconds of upload.

### (3) Data Acquisition & Visualization:

The client shall automatically upload measurement data (S-parameters) to Render Platform A within 10 seconds of measurement completion.

The server shall retrieve and display data within 2 seconds of user clicking the "Receive Data" button.

Data display shall show S11, S12, S21, S22 parameters with  $\pm 0.1$  dB display accuracy.

(4) Camera Visualization (Future Work):

The camera shall capture and transmit instrument status images at minimum 1 frame per second.

The web interface shall display live camera feed with latency  $\leq 2$  seconds.

## 2. Design

### 2.1 Block Diagram

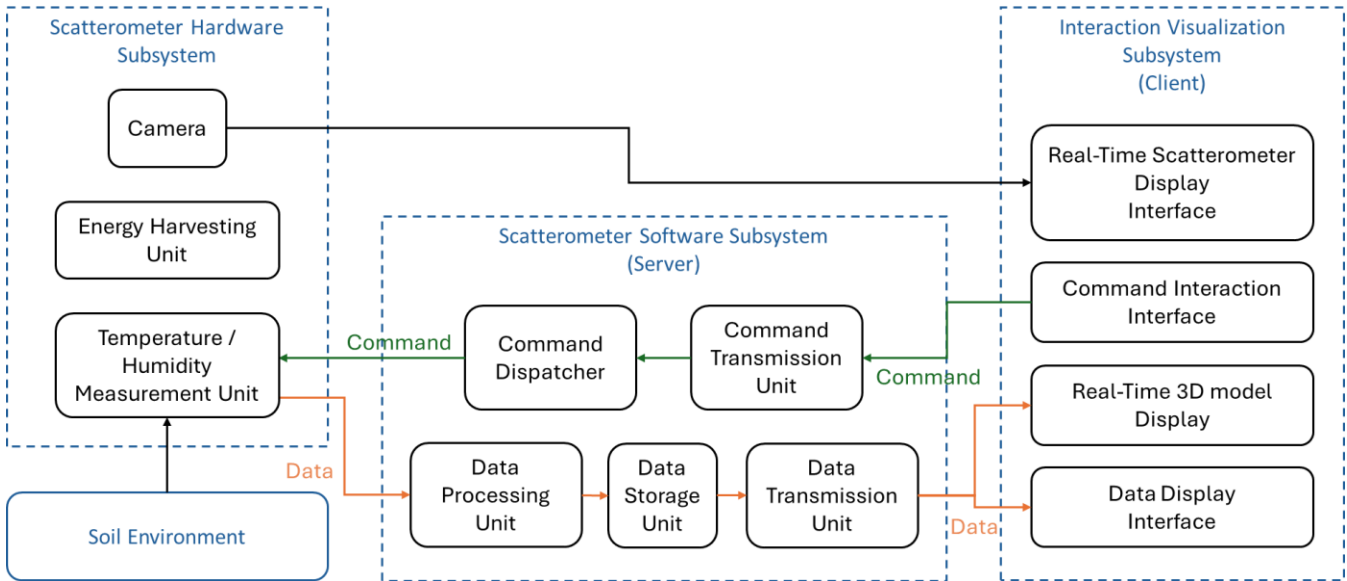


Figure 2: Block Diagram of the scatterometer and its digital twin

### 2.2 Subsystem Description

The Scatterometer Digital Twin System is responsible for real-time remote monitoring, control, and visualization of a physical scatterometer device used for soil and environmental sensing. It employs a coupled hardware-software architecture to achieve energy harvesting, environmental data acquisition (temperature, humidity, imaging), wireless data transmission, and an interactive 3D digital twin interface for user command and status display.

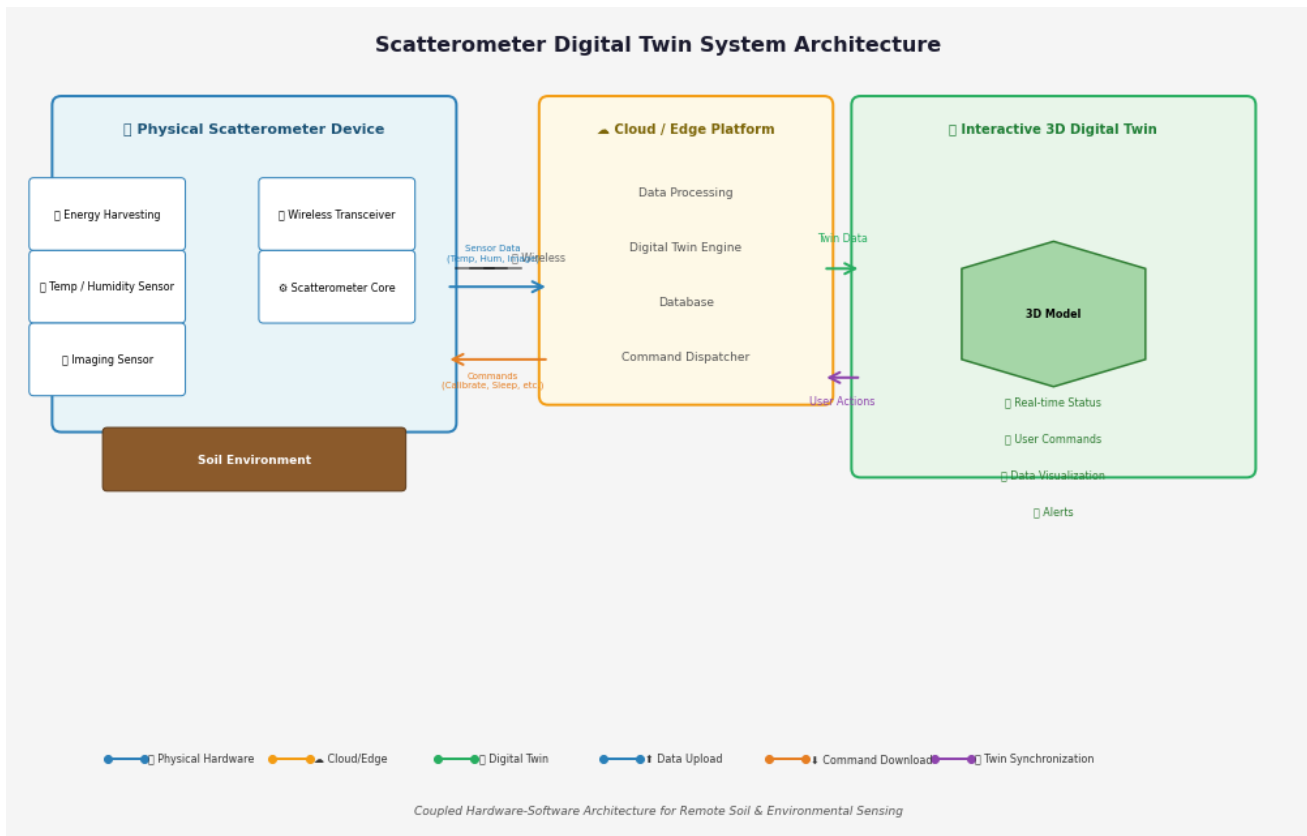


Figure 3: The total design for the scatterometer and its digital twin

### 2.2.1 Scatterometer Hardware Subsystem

The Scatterometer Hardware Subsystem is responsible for energy harvesting, power storage, environmental sensing, image capture, and wireless data transmission. The subsystem interacts with the external environment (sun, soil) and the Interactive Visualization Subsystem (digital twin) via wireless communication.

Components and Functions:

1. Sun and Solar Panel: Provides primary charging current to the storage battery under solar irradiance.
2. Li-Ion Charger (Emergency): Provides backup charging capability when solar input is unavailable or insufficient.
3. Storage Battery: Stores electrical energy from the solar panel and emergency charger; supplies regulated power to all sensors, camera, and wireless module.
4. Soil Temperature and Humidity Sensors: Detect real-time soil conditions at the deployment site.
5. Camera: Captures visual data of the scatterometer's environment or target area.
6. Coupler and Wireless Information Transfer Module: Transmits sensor data and images to the digital twin and receives user commands from the Interactive Visualization Subsystem.

Interactions with Other Subsystems:

- Input from: Sun (irradiance), soil (temperature/humidity), Interactive Visualization Subsystem (user commands via wireless link).
- Output to: Interactive Visualization Subsystem (sensor data, images, battery status via wireless link).

requirement	verification
<ol style="list-style-type: none"> <li>1. The solar panel shall provide a minimum output power of 5 W <math>\pm</math>0.5 W under standard test conditions (1000 W/m<sup>2</sup> irradiance, 25°C ambient).</li> <li>2. The Li-Ion emergency charger shall fully charge the storage battery from 0% to 100% state of charge (SOC) within 4 hours <math>\pm</math>10 minutes when solar input is disconnected.</li> <li>3. The storage battery shall maintain system operation for at least 8 hours <math>\pm</math>15 minutes under nominal load (2 W average power consumption) with no charging input.</li> <li>4. The soil temperature sensor shall measure temperature from -10°C to 60°C with an accuracy of <math>\pm</math>0.5°C and resolution of 0.1°C.</li> <li>5. The soil humidity sensor shall measure volumetric water content from 0% to 100% with an accuracy of <math>\pm</math>3% and resolution of 1%.</li> <li>6. The camera shall capture color images at a minimum resolution of 640×480 pixels (<math>\pm</math>5%) and a frame rate of at least 1 frame per minute when commanded by the user.</li> <li>7. The wireless information transfer module shall achieve a minimum data rate of 250 kbps <math>\pm</math>10% at a line-of-sight distance of 100 m <math>\pm</math>5 m.</li> <li>8. The coupler shall interface the sensors and camera to the wireless module with a signal loss less than 1 dB at the operating frequency.</li> </ol>	<ol style="list-style-type: none"> <li>1. Measured power output is between 4.5 W and 5.5 W.</li> <li>2. Charge the equipment from 0% SOC to 100% SOC and the charging time between 3 hours 50 minutes and 4 hours 10 minutes.</li> <li>3. Runtime between 7 hours 45 minutes and 8 hours 15 minutes when fully charged.</li> <li>4. Place soil temperature sensor in a calibrated temperature bath at -10°C, 0°C, 25°C, 40°C, and 60°C. Record sensor output at each setpoint after stabilization (5 minutes). Each reading within <math>\pm</math>0.5°C of bath temperature. Resolution confirmed as 0.1°C on display.</li> <li>5. Prepare soil samples with known volumetric moisture content: 0% (oven-dry), 25%, 50%, 75%, and 100% (saturated by weight). Insert humidity sensor into each sample. Record readings after 2 minutes. Each reading within <math>\pm</math>3% of prepared reference value.</li> <li>6. All images have resolution between 608×456 and 672×504 pixels. All images captured successfully.</li> <li>7. Average data rate between 225 kbps and 275 kbps for all 10 trials.</li> <li>8. Use a network analyzer or signal generator to inject a known signal level at the sensor interface. Measure signal level at the wireless module input. Compute loss = 20 <math>\log_{10}(V_{in}/V_{out})</math> and the measured loss &lt; 1 dB.</li> </ol>

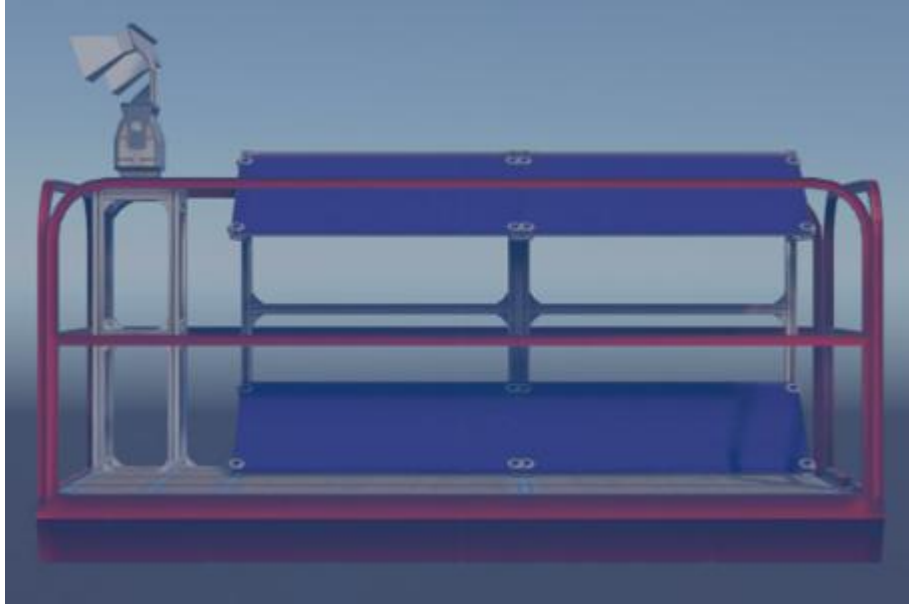


Figure 4: The effect demonstration diagram for scatterometer hardware

### 2.2.2 Interactive Visualization Subsystem

The Interactive Visualization Subsystem is responsible for displaying the scatterometer's operational status in real-time 3D, providing an interface for user command entry, and presenting incoming sensor data from the physical scatterometer device. This subsystem interacts with the user (input/output) and the Scatterometer Hardware Subsystem (receives data, sends commands via wireless link).

Components and Functions:

1. Real-time 3D Scatterometer Status Display: Renders a three-dimensional model of the scatterometer device, dynamically updating to reflect current battery level, charging status, sensor activity, and camera orientation.
2. Interactive Interface for User Commands: Provides graphical controls (buttons, sliders, text input) allowing the user to send commands such as: request sensor data, capture image and turn to given angle to measure the data.
3. Data Display Interface: Presents numerical and graphical representations of incoming scatterometer data, including soil temperature, soil humidity, camera feed, and S11, S12, S21, S22.

Interactions with Other Subsystems:

- Input from: Scatterometer Hardware Subsystem (sensor data, images), User (keyboard/mouse commands).
- Output to: Scatterometer Hardware Subsystem (control commands via wireless link), User (visual display, 3D model, numerical readouts).

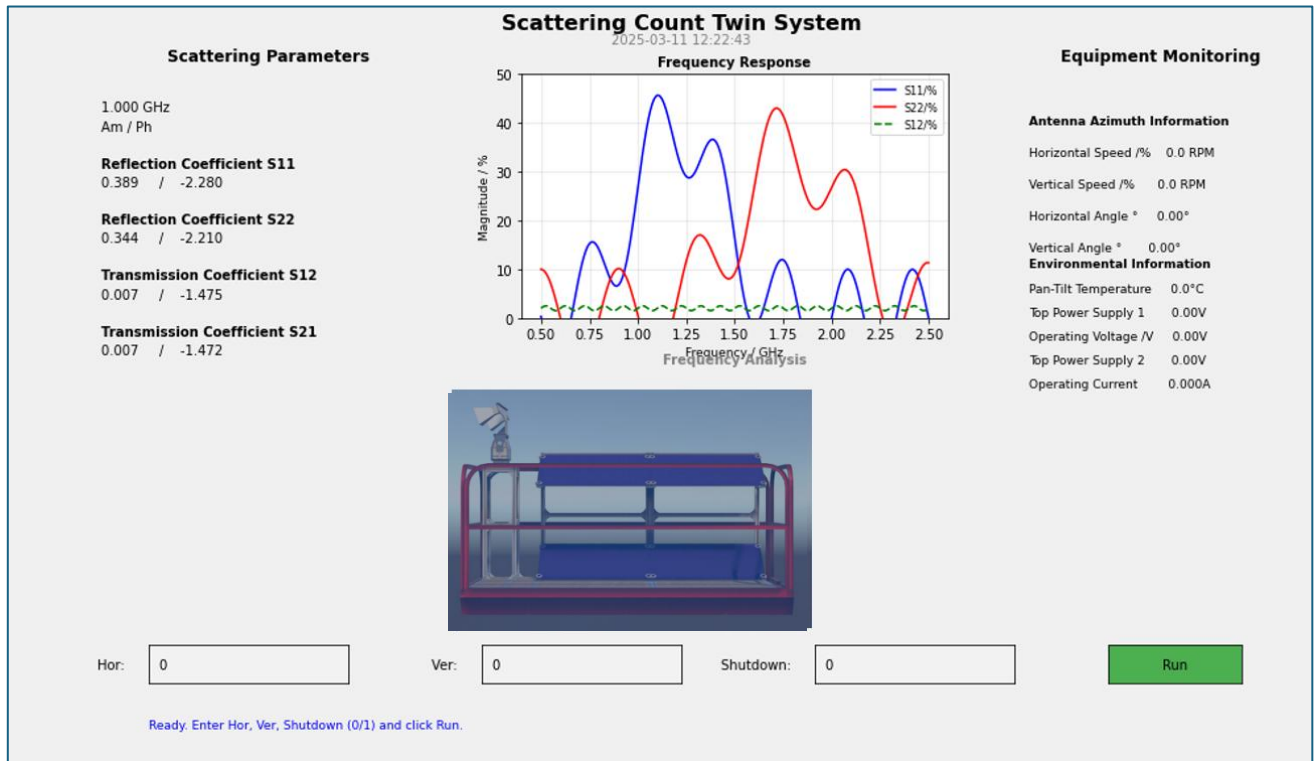


Figure 5: The effect demonstration diagram for interactive interface

Requirement	verification
<ol style="list-style-type: none"> <li>The 3D scatterometer display shall render at a minimum of 30 frames per second (FPS) <math>\pm 5</math> FPS under nominal operating conditions.</li> <li>The interactive command interface shall register and transmit a user command to the hardware subsystem within 100 ms <math>\pm 20</math> ms of button click or key press.</li> <li>The data display interface shall show soil temperature readings with an accuracy equal to the sensor's transmitted value <math>\pm 0.1^\circ\text{C}</math> (display rounding tolerance).</li> <li>The data display interface shall refresh numerical sensor readouts at a minimum rate of 1 Hz <math>\pm 0.1</math> Hz (once per second).</li> <li>The camera feed display window shall show incoming video at a minimum of 10 FPS <math>\pm 2</math> FPS when the camera is active.</li> <li>The subsystem shall receive the command user input (the horizontal angle, vertical angle and</li> </ol>	<ol style="list-style-type: none"> <li>Average frame rate between 25 FPS and 35 FPS. Minimum instantaneous frame rate <math>\geq 20</math> FPS.</li> <li>95% of commands transmitted within 80 ms to 120 ms.</li> <li>All displayed values within <math>\pm 0.1^\circ\text{C}</math> of transmitted values.</li> <li>All refresh intervals between 0.9 s and 1.1 s. Average interval = 1.0 s <math>\pm 0.05</math> s.</li> <li>Average displayed frame rate between 8 FPS and 12 FPS.</li> <li>Every user's input corresponds to an input file and can be transferred to the equipment and we can find the file in equipment. And the equipment can automatically download and read the file.</li> <li>Each view update occurs within 2s of command input. But it can't display every state of the scatterometer, but just display the status when the equipment moves to certain angle and keeps still.</li> </ol>

whether to reset) and then generate a data file and automatically send to the equipment.

7. The 3D display shall support rotation (pan: 0° to 360°, tilt: -90° to +90°) with a response time of ≤200 ms per user input.

### 2.3 Remote control and wireless communication

To realize the mutual communication between the digital twin and the scatterometer hardware, we utilize cloud storage as the bridge between the digital twin and the hardware equipment. And we use *render* as the cloud storage platform.

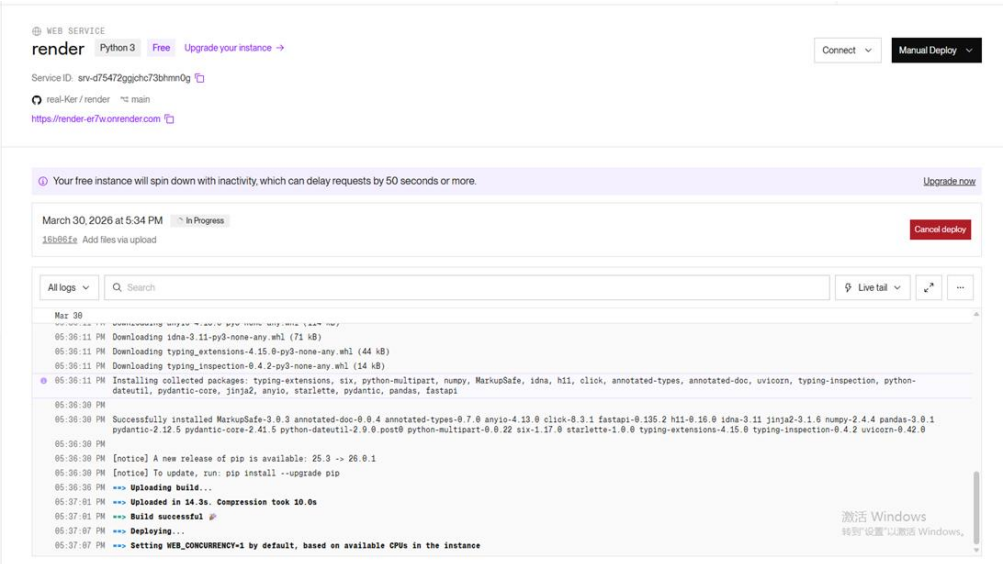


Figure 6: The built render platform on GitHub

<p><b>Free</b></p> <p>\$0 / month</p> <p>512 MB (RAM)</p> <p>0.1 CPU</p>	<p><b>Upgrade to enable more features</b></p> <p>Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.</p>
<p><b>Starter</b></p> <p>\$7 / month</p> <p>512 MB (RAM)</p> <p>0.5 CPU</p>	<p><b>Standard</b></p> <p>\$25 / month</p> <p>2 GB (RAM)</p> <p>1 CPU</p>
<p><b>Pro</b></p> <p>4 GB (RAM)</p>	<p><b>Pro Plus</b></p> <p>8 GB (RAM)</p>

Figure 7: The render price and its storage size



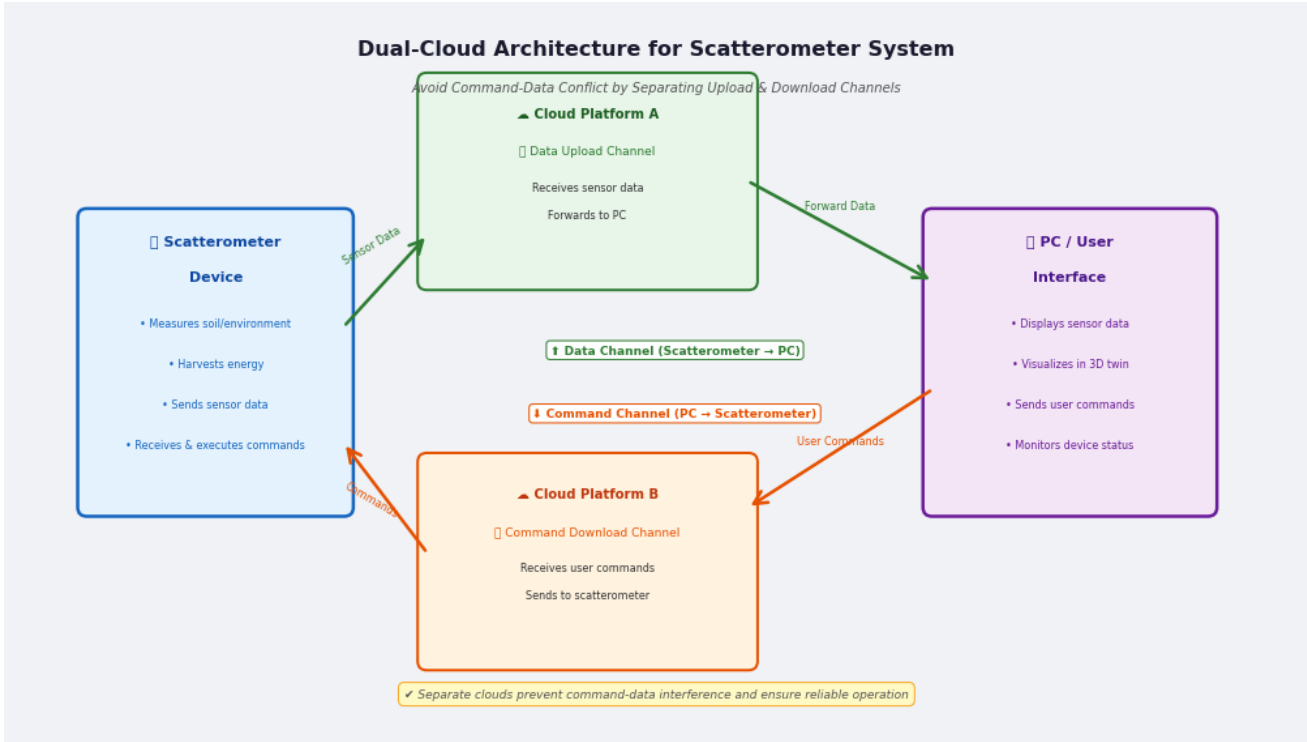


Figure 9: The dual-cloud architecture for scatterometer system

We choose render free for the set up of the cloud storage. Because the data size from the digital twin and scatterometer is small and the free version can totally handle it. However, as the data size grows or the frequent request of measurement, we may consider NAS for the cloud storage platform.



Figure 10: The alternative NAS and its description

2.3.1 receive/ send data

To receive data from scatterometer, we will build a specific *render* platform for scatterometer to send and for user's PC to receive. To realize this function, the scatterometer will automatically upload the data

measured in file. And then the client PC will automatically read the data and display the data collected with the button “receive data” from the interactive interface.

requirement	verification
<ol style="list-style-type: none"> <li>1. The scatterometer can upload the data measured to cloud storage automatically.</li> <li>2. The client PC will receive the data from the storage and display it when the receive button is triggered.</li> </ol>	<ol style="list-style-type: none"> <li>1. The measured data will be presented in cloud storage with 3s after the completion of the measurement.</li> <li>2. After the present of the data in cloud storage, press the button from the interactive interface, the data will be displayed within 1s.</li> </ol>

### 2.3.2 receive and send commands

To ensure commands communication, we will set up another cloud storage platform.

Clients can input the commands for horizontal angle and the vertical angle of the scattermeter and whether to shut down after measurement from the interactive interface. Then the interface will generate a file and upload it to the cloud automatically.

Scatterometer can download the file from the cloud and execute it automatically.

requirement	verification
<ol style="list-style-type: none"> <li>1. The client interface can generate the file automatically.</li> <li>2. The equipment should download the file automatically when detecting a new command file.</li> <li>3. The equipment should execute commands automatically.</li> </ol>	<ol style="list-style-type: none"> <li>1. After the input from clients, the command file should be present at the cloud storage within 1s.</li> <li>2. The file should be found on the equipment after the presence of command file at the cloud storage.</li> <li>3. The instrument should turn to the angle given by clients correctly, measure and generate data file.</li> </ol>

### 2.4 built-in camera

The Built-in Camera Subsystem is responsible for capturing visual data of the scatterometer's surrounding environment and transmitting image frames to the onboard processing unit. This subsystem enables remote visual monitoring of the deployment site and provides supplementary data for situational awareness. The camera interfaces directly with the coupler and wireless information transfer module for transmission to the Interactive Visualization Subsystem.

requirement	verification
The built-in camera shall capture color images at a minimum resolution of 1280×720 pixels (720p) ±5%.	All images have width between 1216 and 1344 pixels and height between 684 and 756 pixels. And can be displayed on the interactive interface.

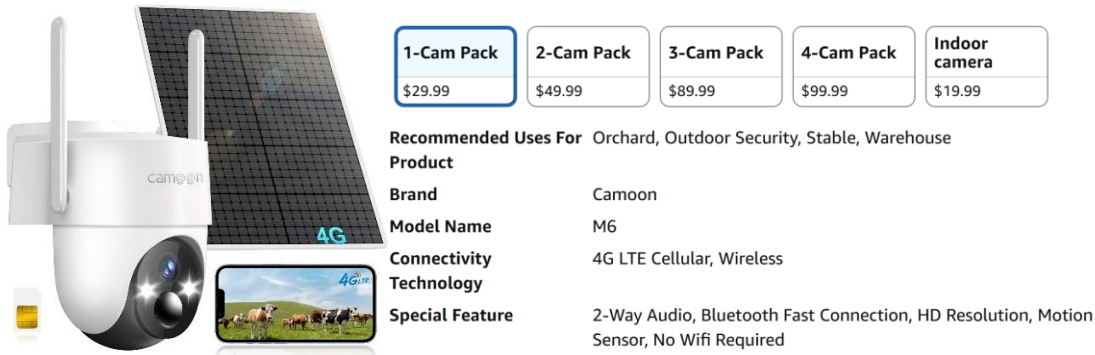


Figure 11: The information of the monitor

## 2.7 Tolerance Analysis

The uncertainties of the Scatterometer Digital Twin System arise from angle positioning, communication delay, and data accuracy, all of which may affect the overall performance of the system.

### 2.7.1 Angle Error

The scatterometer measures electromagnetic responses at specific angles. Therefore, angular deviation directly affects measurement accuracy. A small angular deviation may introduce shifts in beam direction shifts, reflection characteristics change, resulting in variations in the measured scattering parameters. Consequently, such deviations can reduce both measurement accuracy and repeatability.

As specified in the system requirements, the angle positioning error (VRS/HRS) should be within  $\pm 1^\circ$ .

### 2.7.2 Communication Delay

The system relies on a cloud-based dual-platform architecture, where delays arise from file upload/download latency and data detection delay. The total delay can be up to 5~6 seconds.

Although the communication delay does not affect static measurement accuracy, it impacts system responsiveness. This may result in non-real-time data visualization or delayed command execution; in extreme cases, it may cause command misalignment.

### 2.7.3 Sensor and Data Accuracy

Sensor tolerances affect environmental data:

1. Temperature:  $\pm 0.5^{\circ}\text{C}$ .
2. Humidity:  $\pm 3\%$ .

These uncertainties are relatively small and are not expected to significantly affect overall system-level performance.

### 2.7.1 Conclusion

The system maintains acceptable performance within specified tolerances, as the primary function (remote measurement and visualization) is not critically affected by communication delay or minor data inaccuracies, although improvements in system responsiveness could further enhance performance.

## 3. Cost

Part	Cost(rmb)
Cables & Connectors	50
1-Cam Pack Camera	440
Render Cloud Platform	200
Mounting Hardware & Enclosure	100
Network Module	150
Total Cost	940

## 4. Schedule

Week	Yurong Wang	Keyi Jin	Jianing Xiao
2/24	Project scope finalization	System architecture design	Hardware inventory
3/3	Web interface design	Render dual-cloud platform setup and testing	Frontend framework selection
3/10	Command input UI implementation		Command file parser and

			validation logic
3/17	Integration: Render → command parser	Python script development for Render	Data display panel(S- parameters table)
3/24			
3/31	End-to-end: Server → Render B → Client	Data upload to Render implementation	Encoder feedback and position verification
4/7			Camera feed display module
4/14	Real-time data plotting	Camera hardware mounting and testing	Camera software integration
4/21			Accuracy testing
4/28	System integration and debugging		
5/5	Performance testing (latency, reliability)		System- level validation and error handling
5/12	Final presentation preparation		

## 5. Ethics and Safety

### 5.1 Ethics

The Functional Digital Twin for Automated Microwave Scatterometer is designed to enable remote scientific research and experimentation. As such, it must adhere to ethical guidelines.

**Data Privacy:** The system transmits measurement data (S-parameters) and camera images through cloud platforms. All data transmitted is non-personal and related only to instrument status and experimental results. No personally identifiable information is collected or transmitted.

**Intellectual Property:** The software developed for this project (Python scripts, web interface, control algorithms) will be documented and made available for academic use. Third-party libraries (Flask, Requests, etc.) will be properly attributed.

**Responsible Use:** The system is intended for legitimate scientific research and educational purposes. Remote control features should only be used by authorized personnel. Access controls will be implemented to prevent unauthorized command execution.

**Environmental Impact:** The scatterometer operates at low power. No hazardous materials are used. Electronic components will be recycled or disposed of properly at end of life.

## 5.2 Safety

**Electrical Safety:** The scatterometer and built-in computer operate at low voltage (5V-24V). All power supplies shall be properly grounded. Electrical connections shall be insulated to prevent short circuits.

**Mechanical Safety:** The VRS and HRS motors can produce torque sufficient to cause injury if fingers or objects are caught in moving parts. Protective covers shall be installed over all moving components. Emergency stop buttons shall be accessible at the client site.

**Remote Operation Safety:** Remote users shall not operate the system without visual confirmation that the test area is clear. A "heartbeat" monitoring system shall ensure the remote connection is active before allowing motor movement.

**Software Safety:** All command files shall be validated before execution. Out-of-range VRS/HRS values shall be rejected. The client script shall include timeout mechanisms to prevent infinite loops or hung operations.

## References

- [1] C. Tison et al., "Processing of the CFOSAT-SWIM data: Algorithm prototyping and simulations," in 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 2015, pp. 5328-5331. doi: 10.1109/IGARSS.2015.7327038.
- [2] A. Mironov, "GUST: A General Utility Scatterometry Tool for Multi-Platform Ocean Wind Retrieval," presented at the EGU General Assembly 2026, Vienna, Austria, May 3–8, 2026, Paper EGU26-5630. doi: 10.5194/egusphere-egu26-5630.
- [3] Render.com, "Render Cloud Platform — Pricing and Storage Specifications," Render, San Francisco, CA, USA. [Online]. Available: <https://render.com/pricing>.