Real-Time Map Data Visualizer

Team 21: Ami Ikanovic, Benjamin Kelly, Isaac Littler, Zahydee Machado, Elizabeth Nelson, Parth Padmanabhan, Scott Fank
Client and Advisor: Dr. Joseph Zambreno

Introduction

Problem Statement:
Geographical data is often difficult to interpret and visualize. It can be challenging to portray data in a manner that conveys a sense of global scale to users.
Solution:
We developed a real-time map data AR visualizer of various publicly accessible data streams for the Microsoft HoloLens 2 using the Unity development platform

Intended Users and Uses

- Prospective students
- Highlight ISU’s commitment to research and innovation
- Showpiece for ECE department

Technical Details

Frontend:
- Mixed Reality Toolkit
- Unity
- Client in C#

Backend:
- Flask Server in Python
- Graphql data queries

Design Requirements

Functional Requirements:
- Ability to scale/rotate the three dimensional globe
- Maintains API layer that standardizes incoming data streams
- Ability to display 3 different types of geographical data streams as input

Non-Functional Requirements:
- Has “pick up and go” ease of use
- Maintain a stable 30 fps
- Runs smoothly, utilizing the limited HoloLens hardware performance

Engineering Constraints:
- Ability to add new data channels via new APIs
- Adaptable display for new categories of data

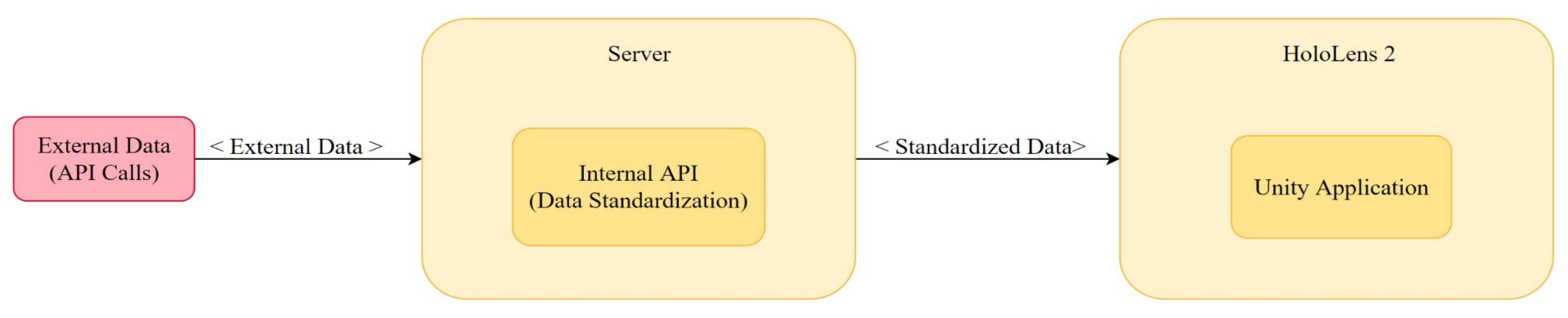
Operating Environments:
- Windows 10 on Microsoft HoloLens 2
- Flask Server on ETG Ubuntu VM

Relevant Standards
- Microsoft Hololens 2 Development Standards
- Unity Engine Development Standards
- IEEE 802.11 Standard for Information technology - Telecommunications and information exchange between systems

Design Approach

Server: Internal API
- Accepts the importation of external data and standardizes it into a single source and format

HoloLens 2: Unity Application
- Application takes geological data (coordinate-based data) and correlate that data to a 3d model of the globe

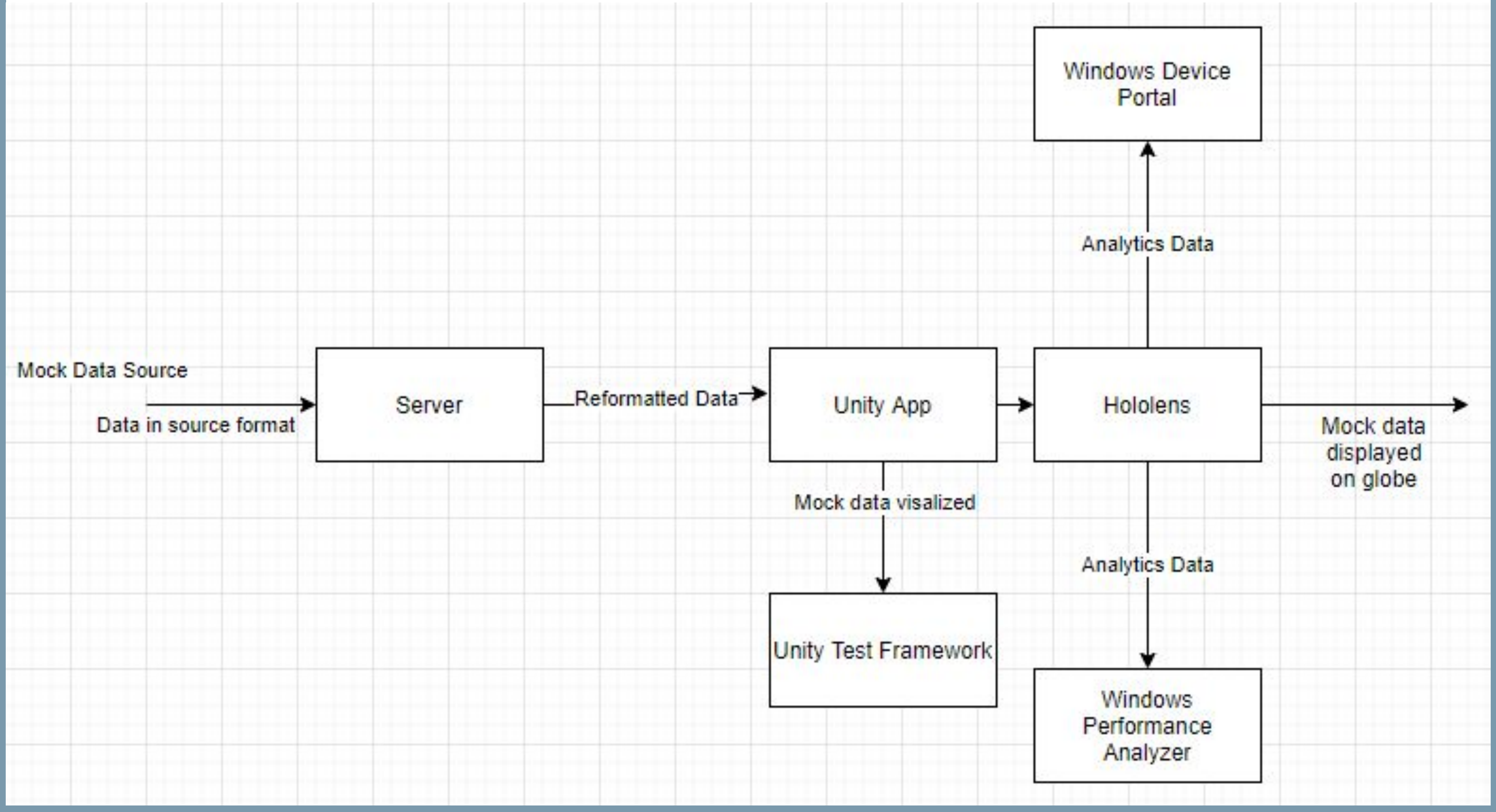


Testing

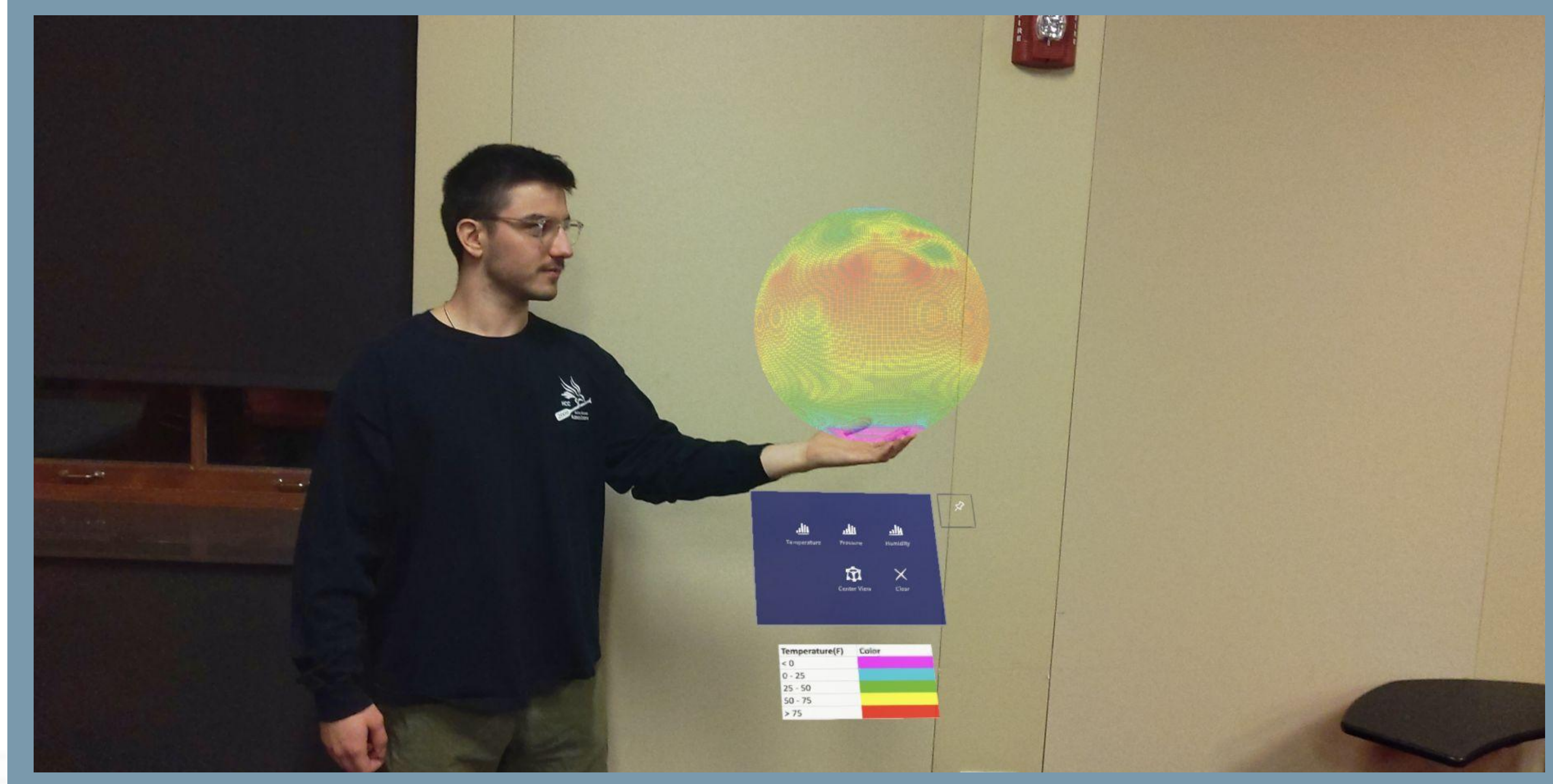
Testing Tools:
- Graphql Playground
- Unity

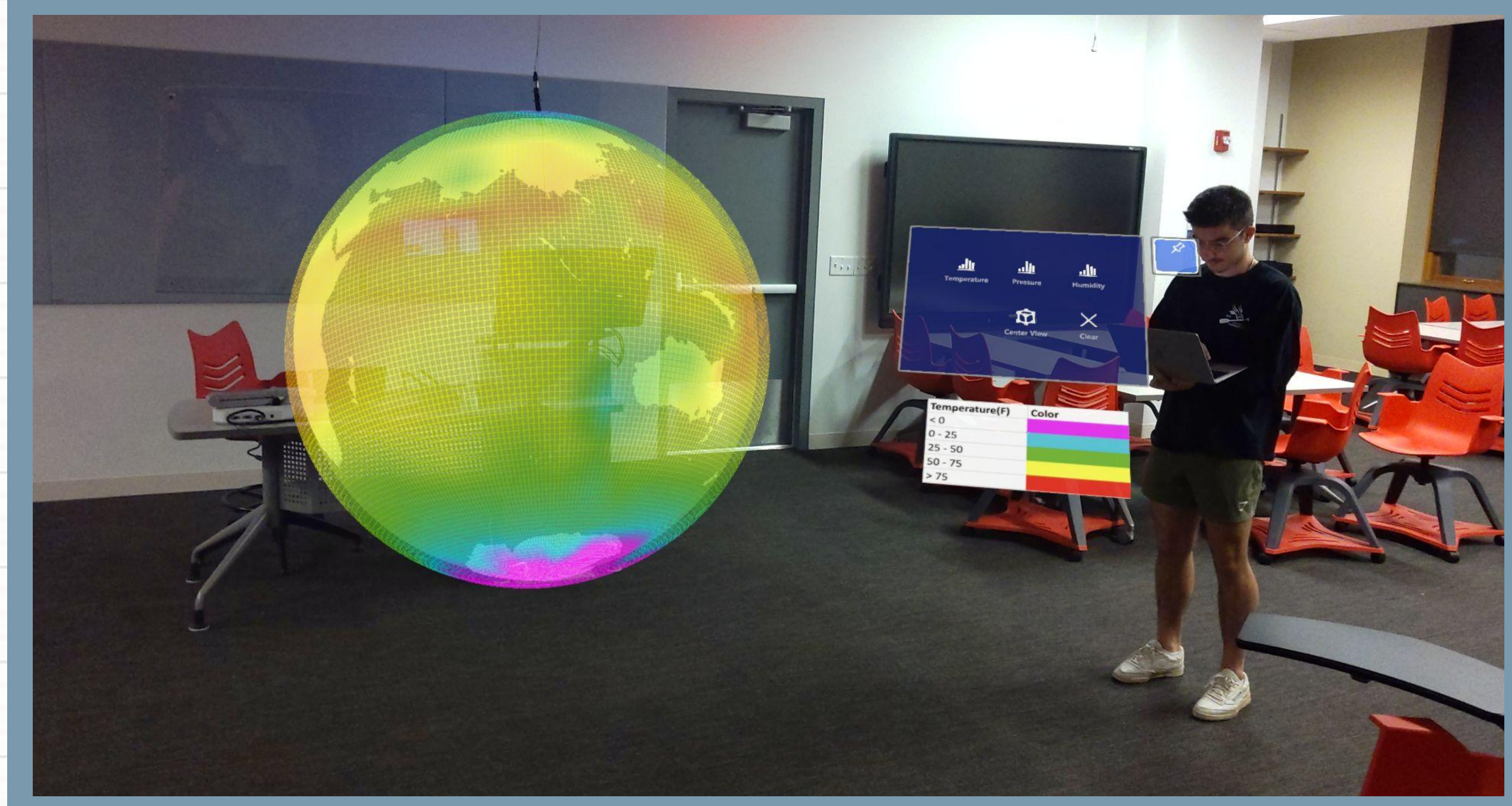
Unit Testing:
- Frontend was tested with a trial set of data to display on a simple sphere in Unity
- Backend was tested to be able to set channels and standardize data from calls through Graphql Playground

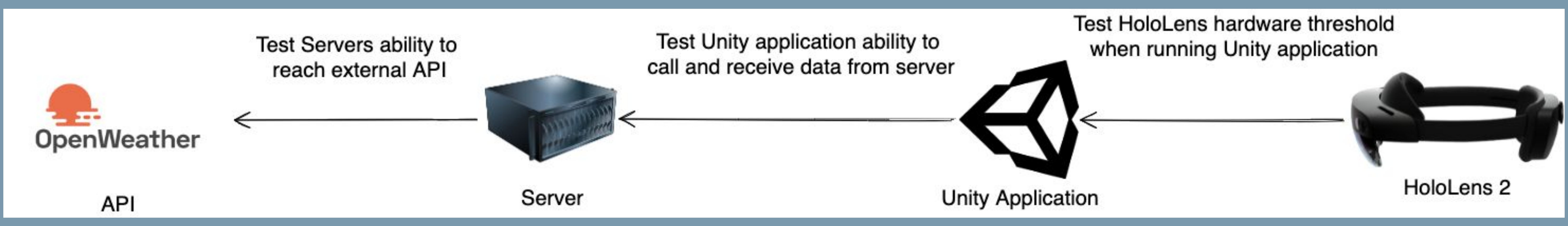
Integration Testing:
- Testing individual data streams through mutations and requests on Unity application
- Building project onto HoloLens 2 for UI and controller testing



Visuals from HoloLens 2






Test Servers ability to reach external API
Test Unity application ability to call and receive data from server
Test HoloLens hardware threshold when running Unity application
OpenWeather
API
Server
Unity Application
HoloLens 2