Keysight 6G Antenna Testing

Senior Design Project

05/XX/2023

UCSC Team:

Shawn Armstrong
James Kohls
Jonah Labi
Luka Kolev
Sahil Gupta
Sebastian Kropp
Shayan Bathaee

Keysight Team:

Jeff Dralla (Director of business development)
Alan Copeland (Senior Software Engineer)
Ivan Diep (Software Solutions Engineer)
Brennen Direnzo (Product manager for automation)

KEYSIGHT TECHNOLOGIES

- Specializes in test and measurement space
- Assists companies with conducting their own internal test and measurement operations

The Customers

- Companies need their solutions
- <DRAFT>

Problem

Weak Testing Standards for 6G

Problems:

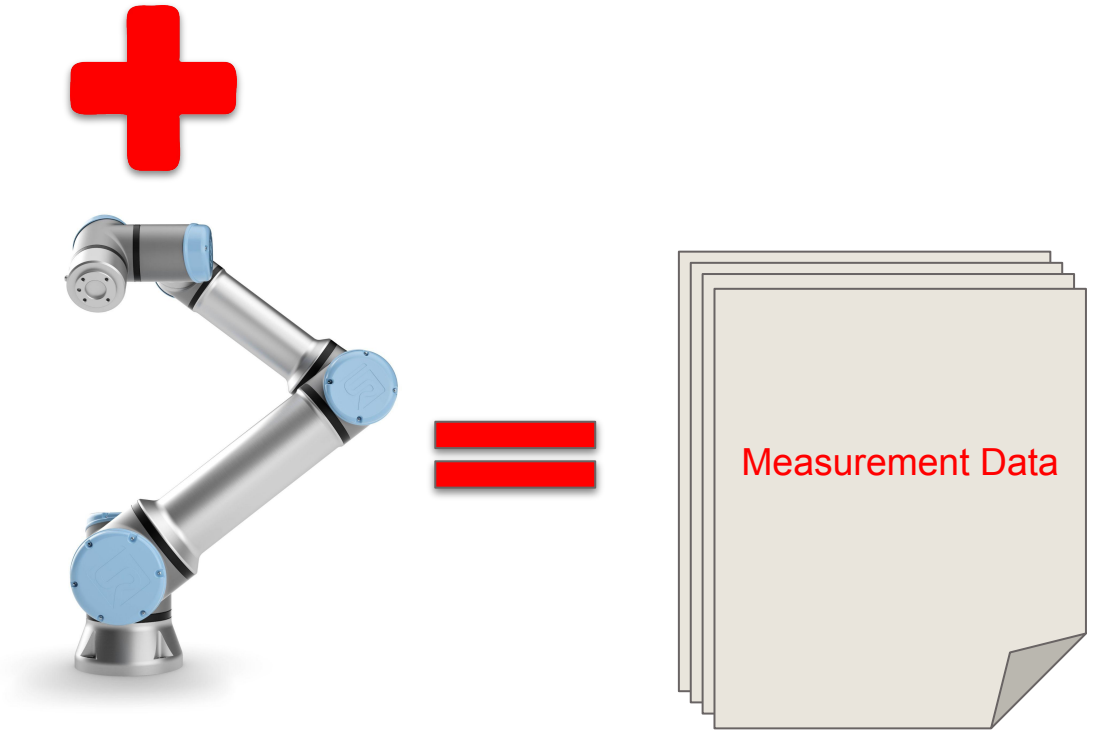
- Without a proper standard companies cannot address the bandwidth and performance demands of a 6G device

Our work:

- Created a user friendly solution that allow easy integration of 6G test plans
- Used OpenTAP software to communicate with cobots connected to a 6G antenna

What is opentap ?

- Open source test and measurement framework
- Define testing operations for a piece of hardware
- Determines successful execution of testing operations
- Log the results and provide information regarding the tested hardware



Relevant OpenTAP Terminology

- Test step
  - Building block of OpenTAP test plans, performs a specific action or measurement.
- Test plan
  - Sequence of test steps.
- Plugin
  - Software component that extends the functionality of OpenTAP by providing additional features.
- Package
  - Collection of related test steps, plugins, and other resources.
- Instrument
  - A hardware device that is used for measuring or controlling a physical system.

Project Goals

High Level Goals:

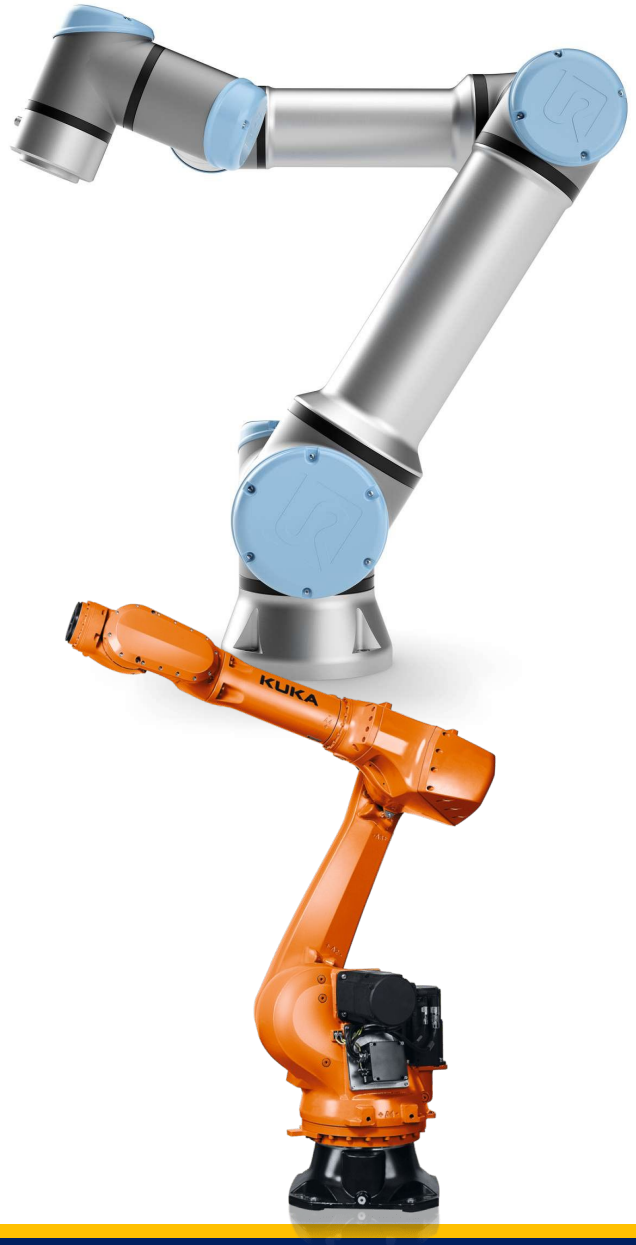
- Support Keysight's efforts in designing a robust system used to test 6G antennas in a controlled environment.

- Extend OpenTAP capabilities to easily enable customers to use collabo robots within their test plans.

Low Level Goals:

- Integrating OpenTAP software with collaborative robot software.

- Augmenting the design to include resources that can be scaled to an arbitrary collaborative robot.



Project Constraints

HARDWARE (IF APPLICABLE)

- Begin with a UR3e Bot.
- Expand functionality to all Cobots.

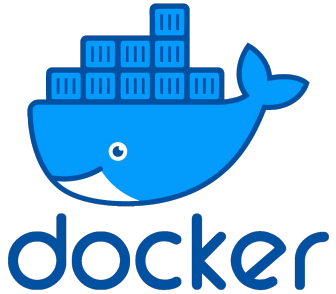
SOFTWARE (IF APPLICABLE)

- Create the plugin in Python.
- Integrate ROS2 into plugin.
- Run on Windows/Linux.

Challenges

- Interfacing with any Cobot without its custom software.
  - Receiving data was the main challenge and we explored many avenues to solve this problem.
- Reverse engineering third party source code related to ROS2.
- Understanding the Python .NET flavor.
- Working with different machines.
  - This problem was solved by dockerizing the plugin development.

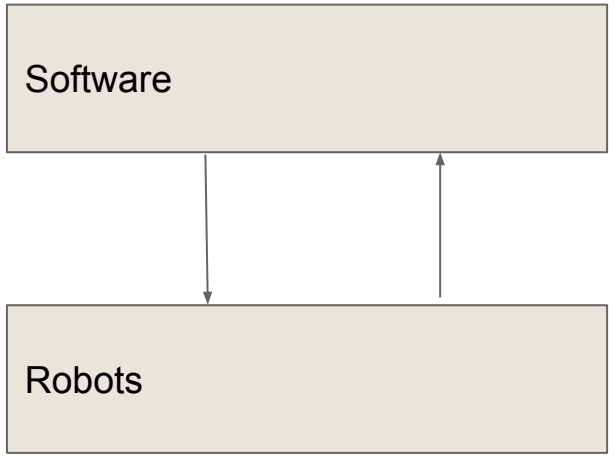
Technologies Used



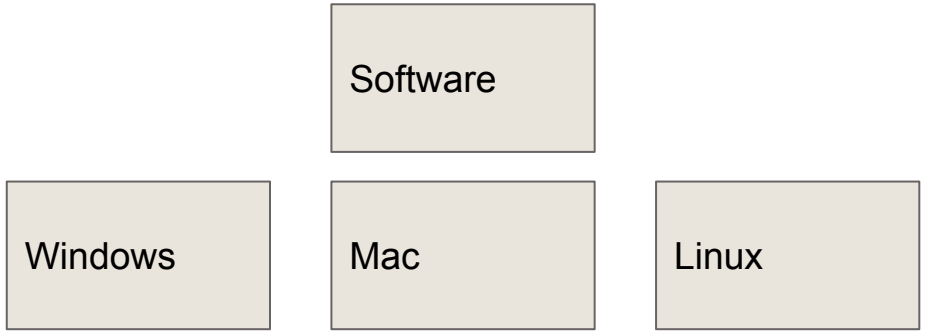
- VNC (Virtual Network Computing)
- Universal Robots Simulator, URcap (sim extension for third party communication)
- Rviz, MoveIt (Cobot visualization)
- Modbus
  - Our Keysight Test Automation plugin is built with a python flavored .NET framework. Therefore we used the Universal Robot's Python based ROS2 Driver.

Approach

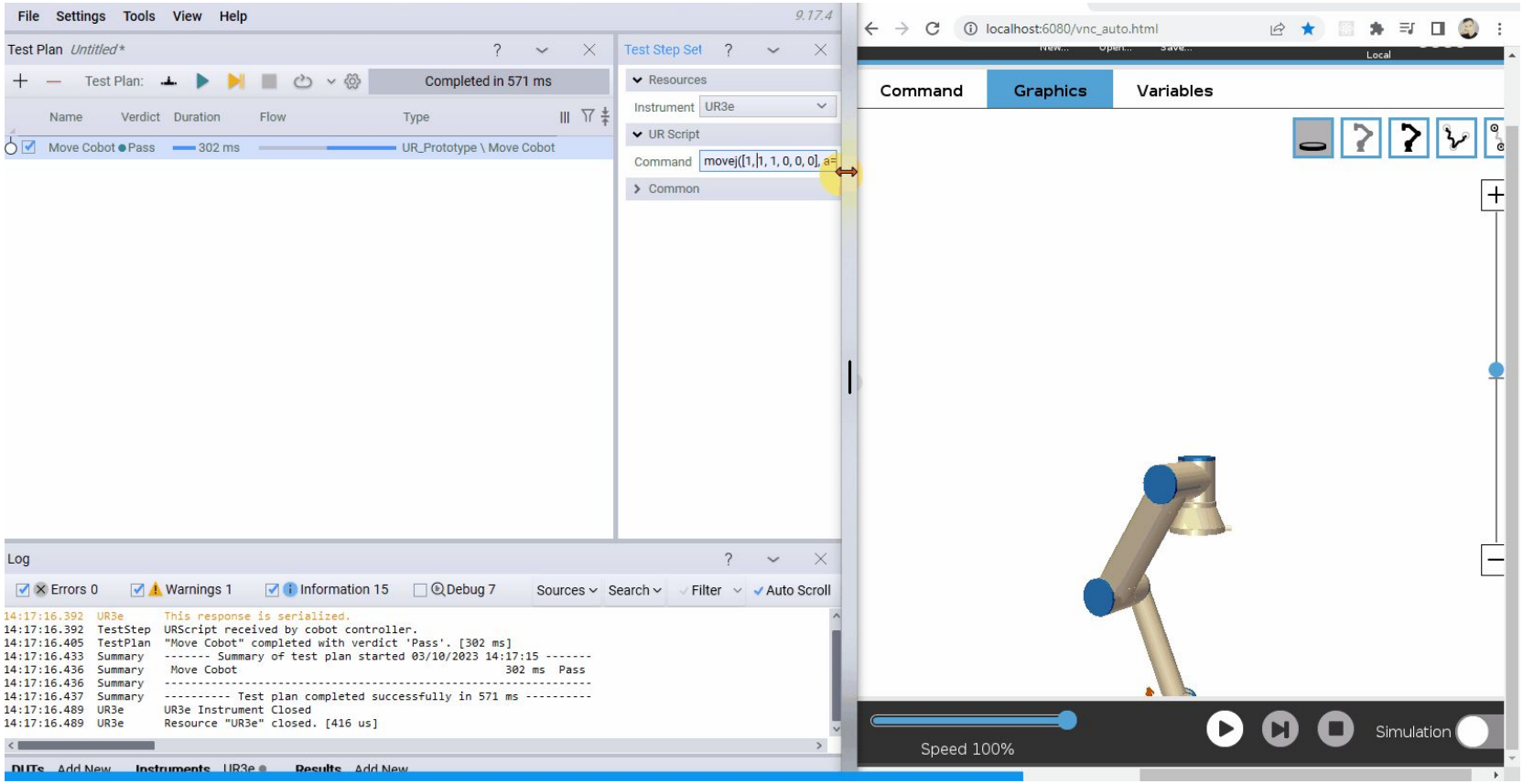
1. Robot Communication with the testing software



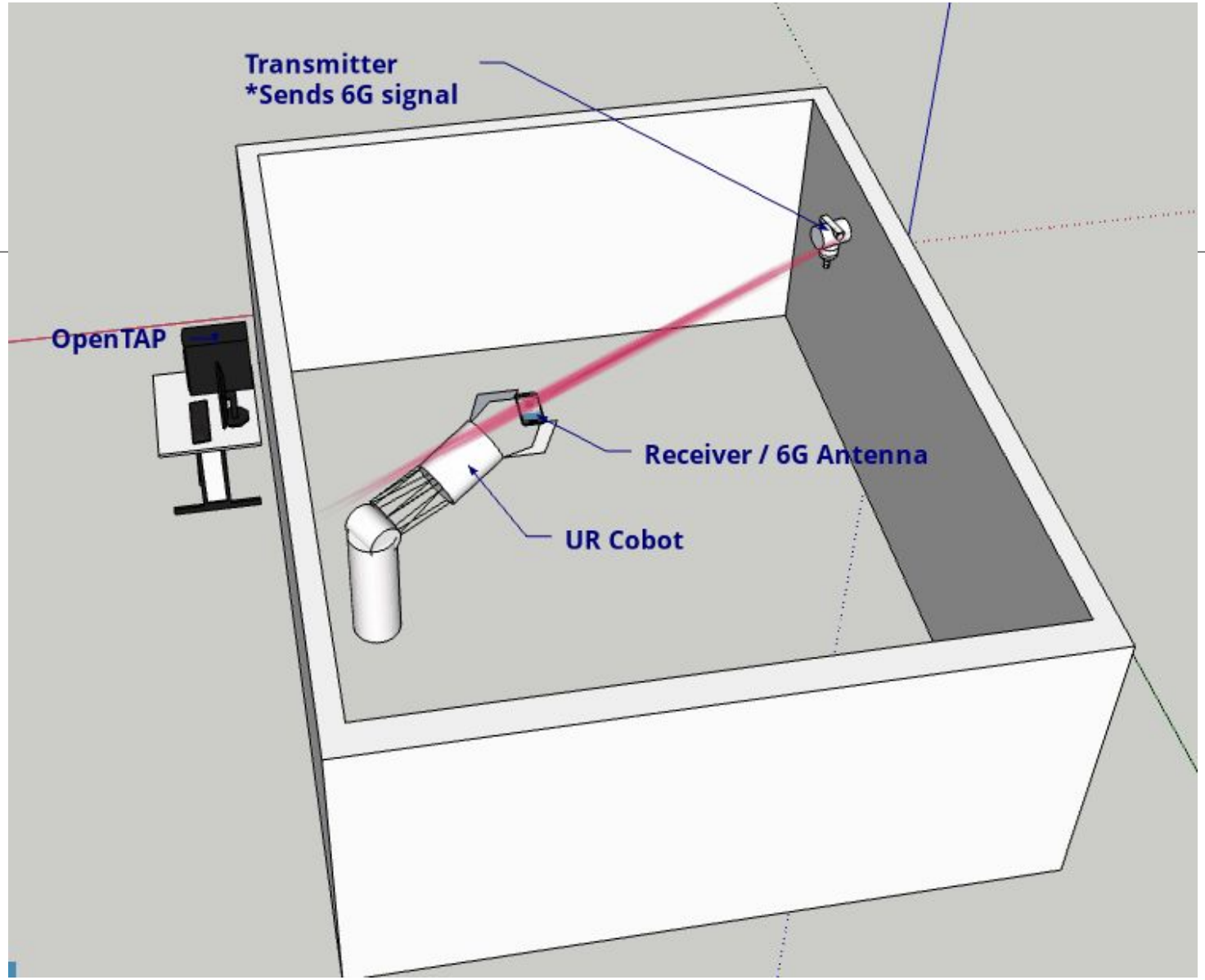
2. Platform independency of the testing software



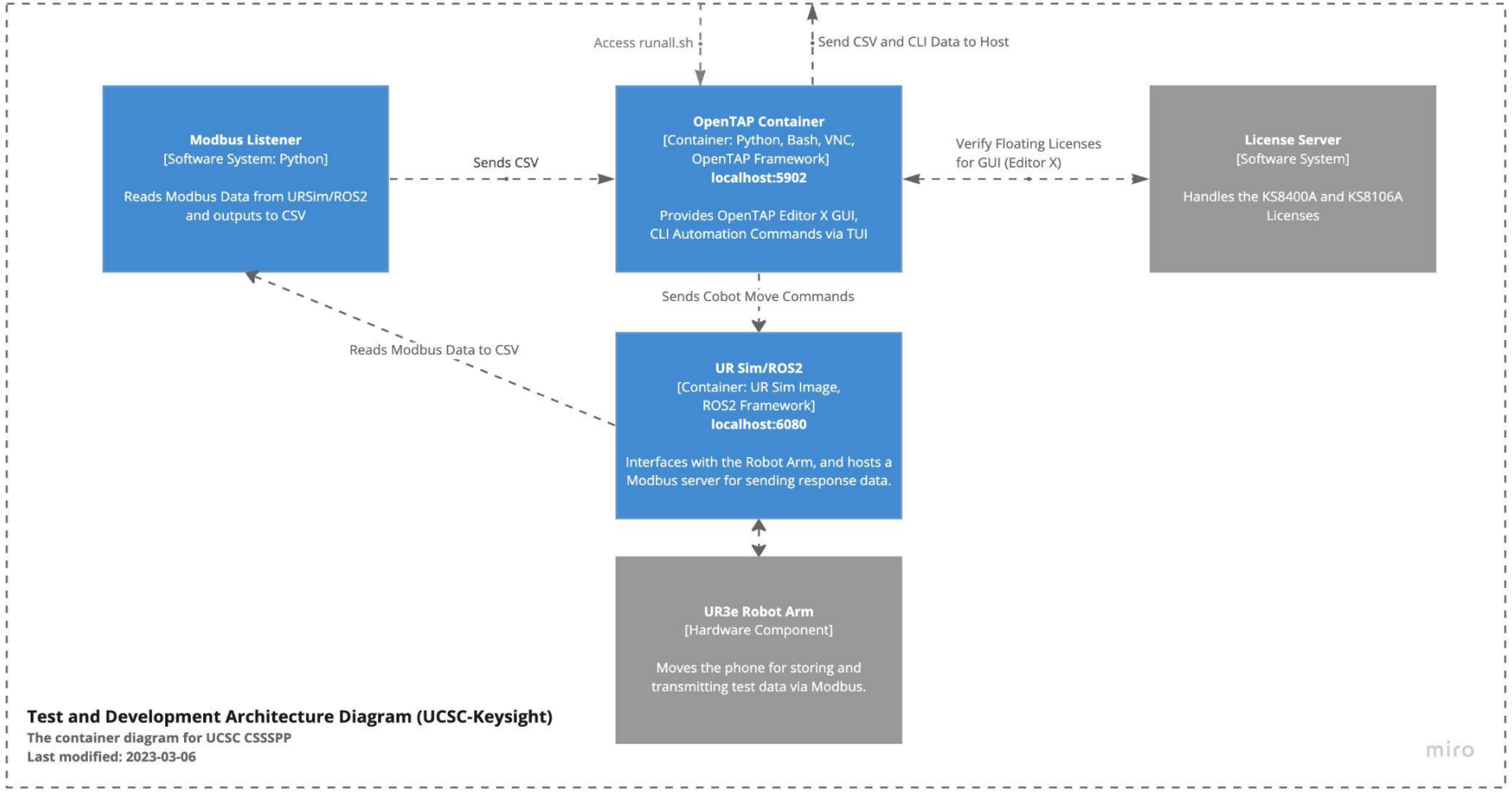
System Illustration

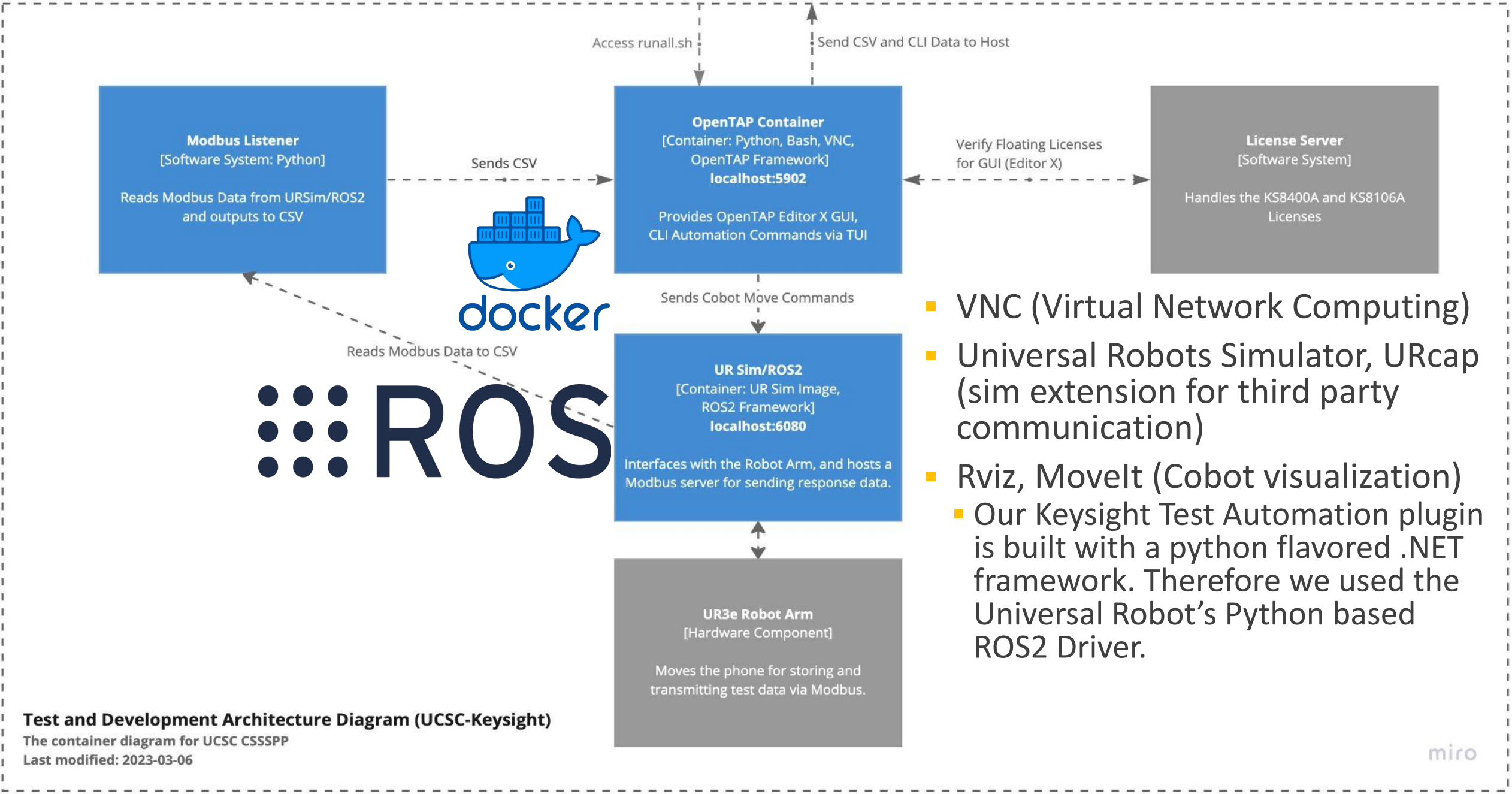


Full Screen Link


Transmitter
*Sends 6G signal
OpenTAP
Receiver / 6G Antenna
UR Cobot

Development Architectural Diagram





- VNC (Virtual Network Computing)
- Universal Robots Simulator, URcap (sim extension for third party communication)
- Rviz, MoveIt (Cobot visualization)
  - Our Keysight Test Automation plugin is built with a python flavored .NET framework. Therefore we used the Universal Robot's Python based ROS2 Driver.

CI/CD Pipeline

To ensure that plugin build remains successful for every merge, a CI pipeline has been created that installs all the necessary dependencies and checks that the plugin builds successfully.

Current pipeline:

- Currently contains build scripts.
- Runs on ubuntu environments.
- Uses .NET CLI to build the project and install plugin dependencies

Results

- UR3e OpenTAP instrument.
- Test step to send commands to cobot via TCP.
- Automated the process of sending move commands.
- Dockerized the project.
- Exploring ROS2 (more than one cobot, scalability)