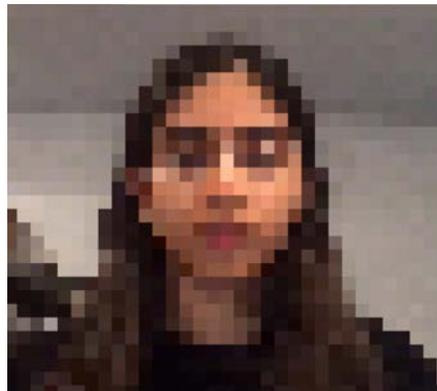
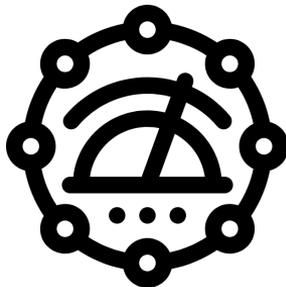


# Low Bandwidth Video Calling

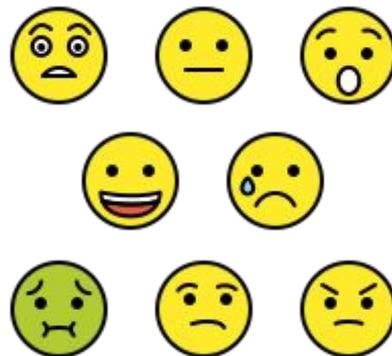
Ananya Joshi, Mentor: Roger Iyengar



Your video call quality is low and frustrating.



The problem is bandwidth constraints.

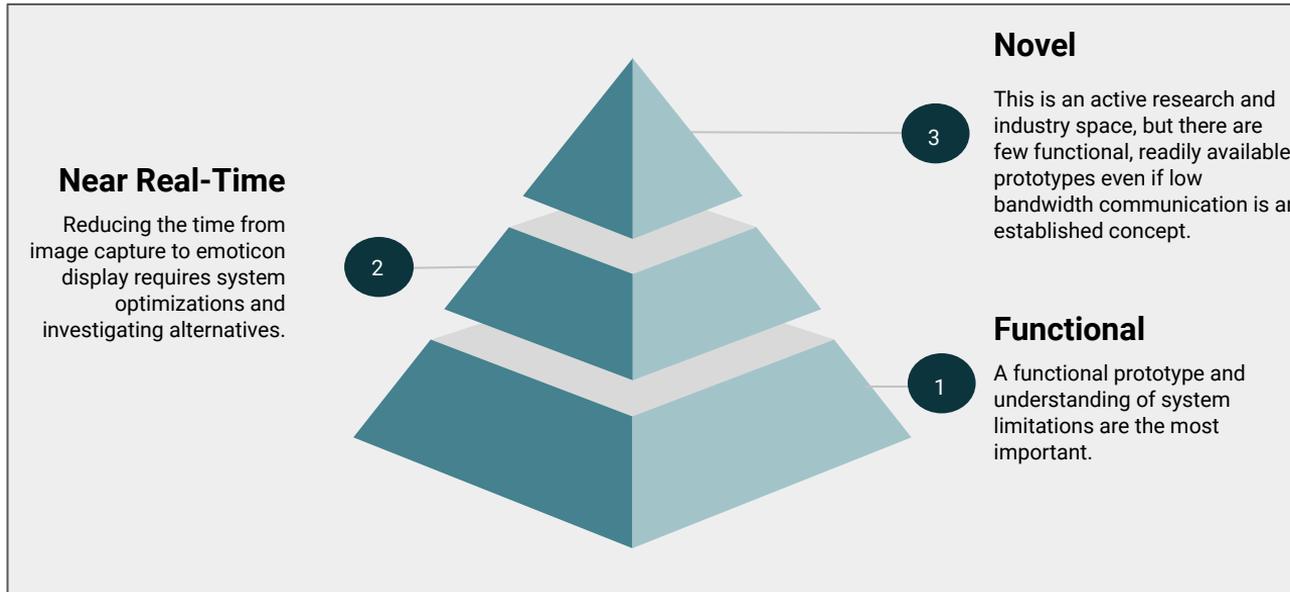


Images have more emotional info than audio.

**Send only the most important information.**

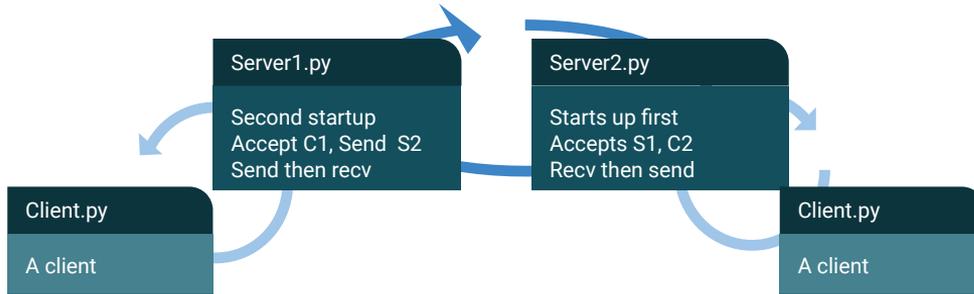
This system is a prototype for VR communication with combined sensor information.

# Project Goals and Background



1. No existing, open-source, and easily accessible implementations
2. Do not take advantage of pervasive computing concepts like cloudlets

# Implementation Details



1. Threaded implementation
2. Processes implementation
3. Multiconn and Selectors
4. Async and websockets

Azure

Sockets

OpenCV

Cloudlet

Python

Docker

Different user needs lead naturally to other implementations (starvation).

# Testing Details

## 98.1%

**Vector size savings**

Average image size: **60991 bytes**

Average vector size: **1104 bytes**

## 0.40 s

**Avg photo to display delay**

Based on the NJ Client, NJ Cloudlet,  
to the PA Cloudlet, to NJ Client 2

## 4

**Max recommended people**

Emoticons may be too small  
Long delays

### Avg Time from Image Capture (1 person, NJ Experiment)

|     |                                 |        |
|-----|---------------------------------|--------|
| 13% | <b>Image Recv at Cloudlet1</b>  | 0.05 s |
| 50% | <b>Vector Send (Azure)</b>      | 0.20 s |
| 25% | <b>Vector Recv at Cloudlet2</b> | 0.10 s |
| 3%  | <b>Img Display at Client 2</b>  | 0.01 s |

# Testing Details

01

## One Device Tests

- Verify basic correctness
- Ignore connection delays



02

## Separate Clients

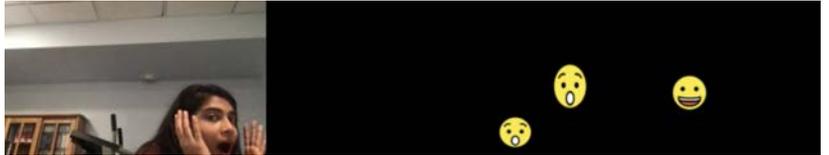
- Identify local conn. errs.
- Understand starvation
- Multi-people testing



03

## Elijah Cloudlet

- Impact of conn. errs.
- Impact of starvation
- Port forwarding



03

## PA Client

- OpenCV versions
- Non-blocking support
- UI insight



# Lessons Learned

## Usability is Costly

Designing with the user in mind means making sure they know that the system is functioning as intended.

## Lack of Info

There is far less documentation for some newer Python packages—few working implementations for sockets.

## Trust your logic

After mapping out the connection logic, it was easy to build a threaded implementation from scratch than relying on existing packages.

## Generalization

Premature optimization might not lead to speedups. Working on the base prototype first might have been less painful than the generalized, multiconn version.

## Chasing novelty

I wanted to build something novel, so I spent time focusing on the project's non-essential components (better model, face representation, multiconn).

Much of the struggle with a multi-connection approach was the problem with blocking sockets, a known challenge in the community.

