RUTGERS
WINLAB | Wireless Information Network Laboratory

AR Mural

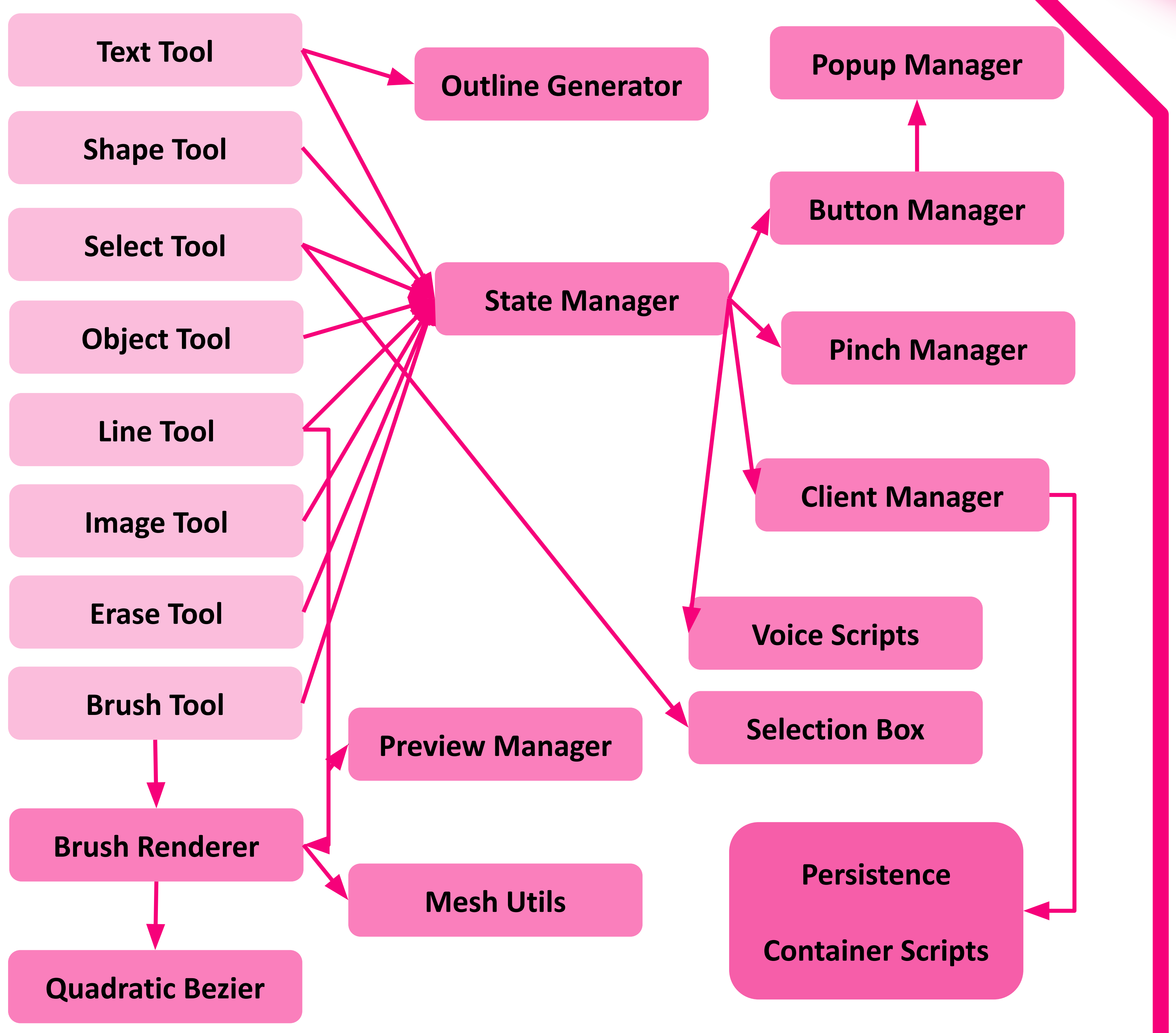
Jovana Lekic, Nima Fallah, Nishi Agrawal, Suhani Sengupta, Sachin Kandan
Advisors: Ivan Seskar & Jennifer Shane

Overview

- Develop a interactive augmented reality application which will allow remote users to collaboratively create a virtual mural through the Unity Game Engine and deployment on AR platforms such as the Micosoft Hololens
- Ensure that the software can not only generate and render the mural, but also allow for very low latency throughout the experience

Functionality

- Have a variety of functionality from inputted text to creating shapes to adding in 3D objects
- Utilizes a server stack to ensure collaboration and real-time processing
  - Data is serialized into a centralized command class that contains the primitive instructions like adding, deleting, and moving existing drawings
  - Utilizes the command game structure in order to make future implementation of features such as undo and redo far more efficient in the future
  - Utilizes a chunk-based system to render only objects that are near you
- Key packages and platforms for this project include the Unity Game Engine, MRTK3 (Mixed Reality Tool Kit Version 3), MQTT (Messaging Queuing Telemetry Transport)



Text Tool
Shape Tool
Select Tool
Object Tool
Line Tool
Image Tool
Erase Tool
Brush Tool
Brush Renderer
Quadratic Bezier
Outline Generator
State Manager
Preview Manager
Mesh Utils
Popup Manager
Button Manager
Pinch Manager
Client Manager
Voice Scripts
Selection Box
Persistence Container Scripts

Diagram of the hierarchy of our code base. See the scripts enabling tools in light pink, their managers in a darker pink, and any external scripts contributing in the darkest shade.

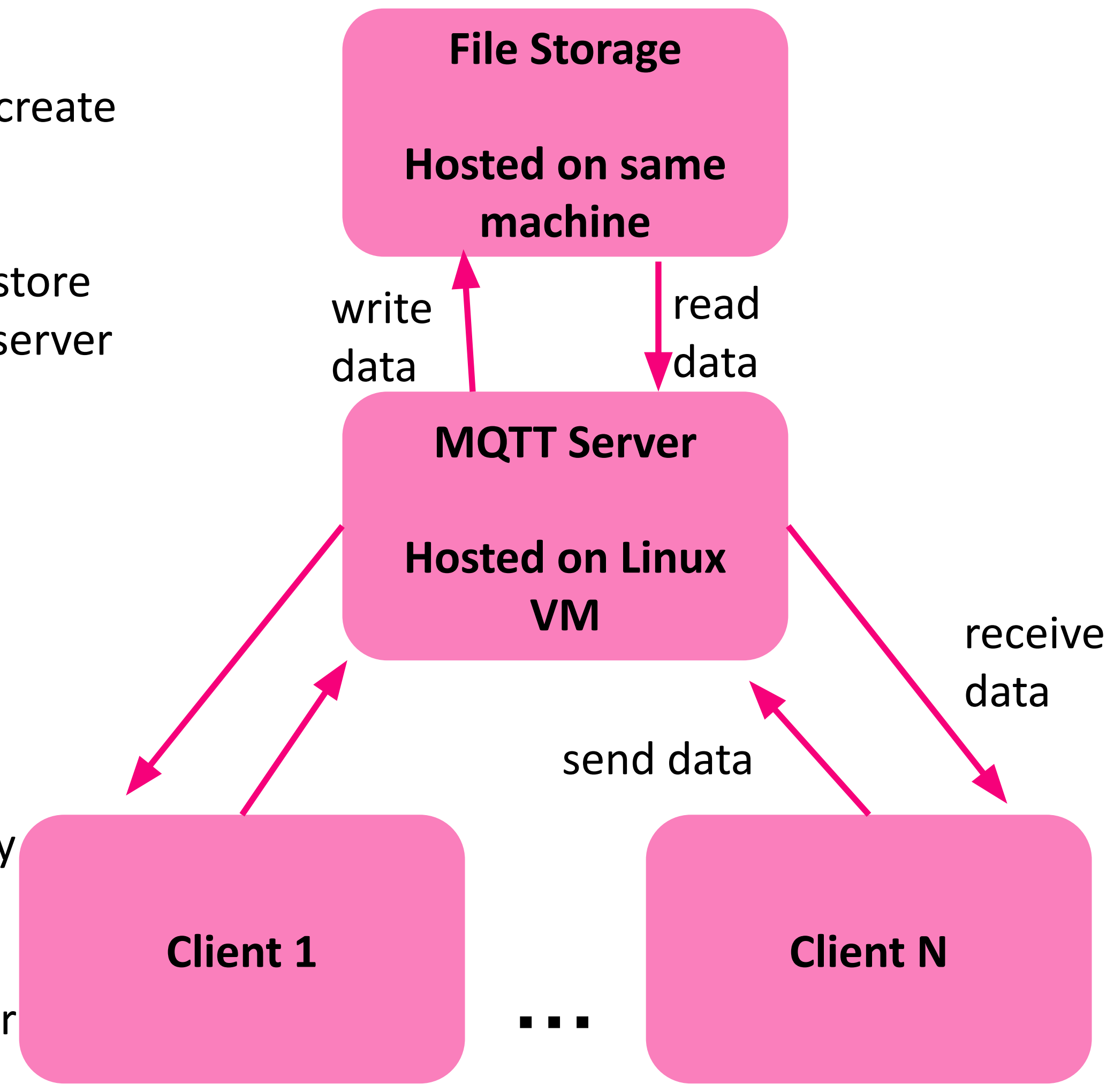
Server

Server Stack

- **MQTTnet** - C# library used to create customized server broker
- **LevelDB** - C++ library used to store data in key-value pairs to the server hard disk

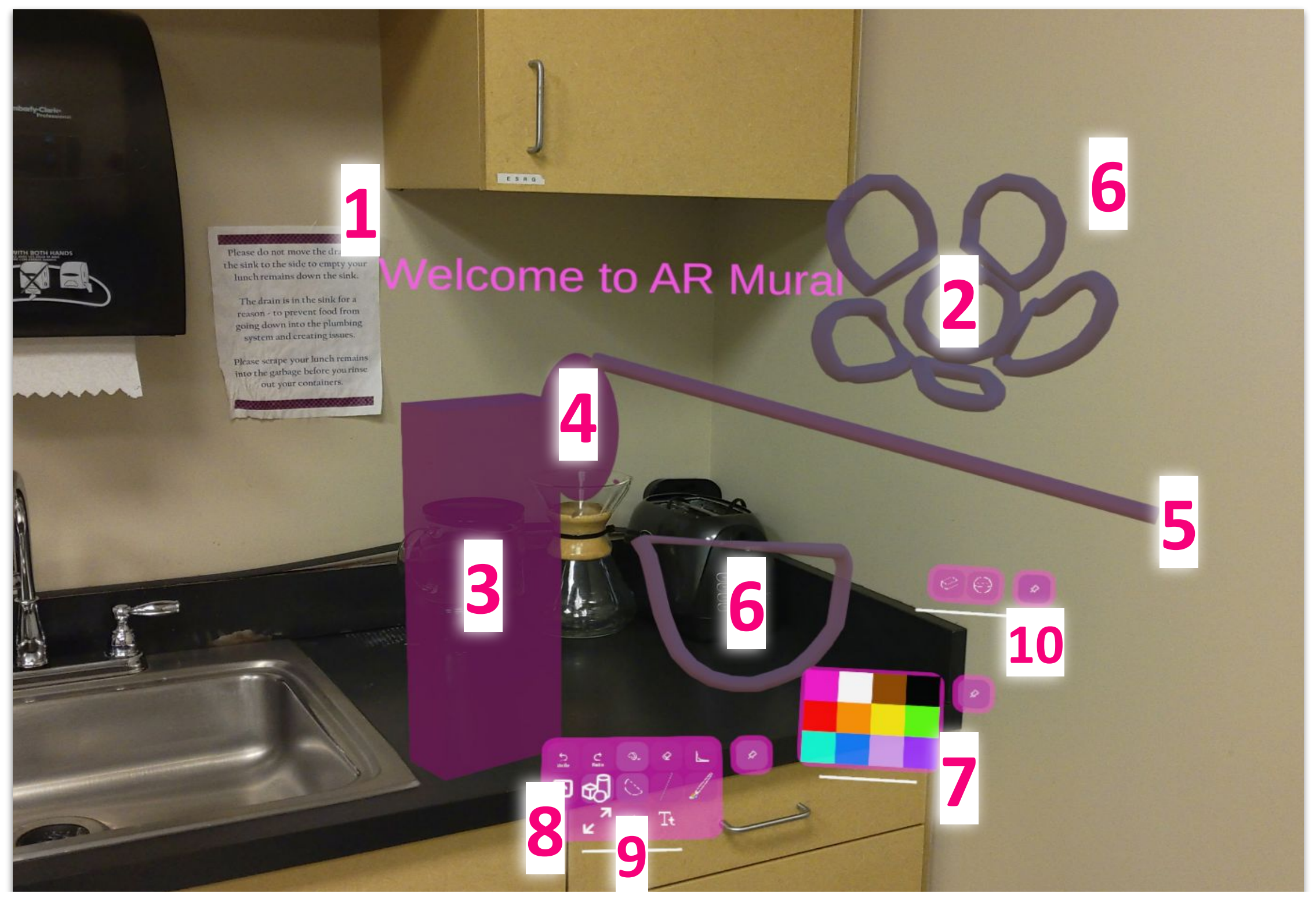
Clientside Stack

- **Unity** - 3D game engine used as main client application
- **C#** - 3D scripting integrated with the Unity standard library
- **MQTTnet** - also used for the user to create customized user broker



File Storage
Hosted on same machine
write data
read data
MQTT Server
Hosted on Linux VM
receive data
send data
Client 1
...
Client N

Simplified diagram of how the server works.



Screenshot of our platform showcasing a variety of our functionality!

1. **Text Tool**: Create custom text in any color. Example: "Welcome to AR Mural".
2. **Brush Tool**: Draw in any color. Example: a white cloud around "Welcome to AR Mural".
3. **Shape Tool (Rectangle)**: Create rectangles in any size and color. Example: a pink rectangular prism.
4. **Shape Tool (Ellipse)**: Create ellipses in any size and color. Example: a pink ellipse.
5. **Line Tool**: Draw lines between two points in any color. Example: a baby pink line.
6. **Loop Tool**: Connect curved line endpoints with a straight line. Example: connecting a semicircle in a custom color.
7. **Color Palette**: Select base colors or create custom colors by combining RGB values.
8. **Submenu**: Opens by clicking the three dots in the first menu. Adopts the current color.
9. **Main Menu**: The initial menu that appears when opening the application. Adopts the current color. Can go back to main menu by clicking the "back" button in the submenu.
10. **Shape Menu:** Select a rectangular prism of an ellipsoid to draw and change its dimensions!

Challenges

- Limited documentation on the Mixed Reality Toolkit Version 3, a critical package for allowing us to develop an AR application within the bounds of the Unity Game Engine
- Balancing efficiency with user experience, especially with regard to deciding whether to have actions by other user render before they finished adding a specific component
- Ensuring easy user navigation, especially as our features grew and our toolbars needed to expand to accommodate
  - Resolved through the usage of many nested submenus

Conclusions

This project allows users to collaboratively create their own murals and explore not only the breadth of their own creativity, but also AR technologies! This work will help to enhance the world's knowledge of what boundaries exist around current packages such as MRTK3 and also introduce those unfamiliar with VR headset technologies to it with a format they are familiar with drawing!

It also provides a great way to jumpstart a world-wide mural sharing and creating application as well as 3D whiteboard brainstorming platforms, should such a project be taken beyond the breadth of our work this summer.

Future Work

- Create the ability for geotagging, or tying murals to physical locations which can be then viewed when the user is a certain distance away from it
- Implementing additional features such as undo and redo, shaped line tools, adding more shapes to the shape tool, etc.
- Consider configuring the code to be deployable on various AR environments including the Oculus Quest

Acknowledgements

We would like to extend a huge thank you to Ivan Seskar and Jennifer Shane for all of their guidance and support throughout this summer for our advisors and mentors.

WINLAB