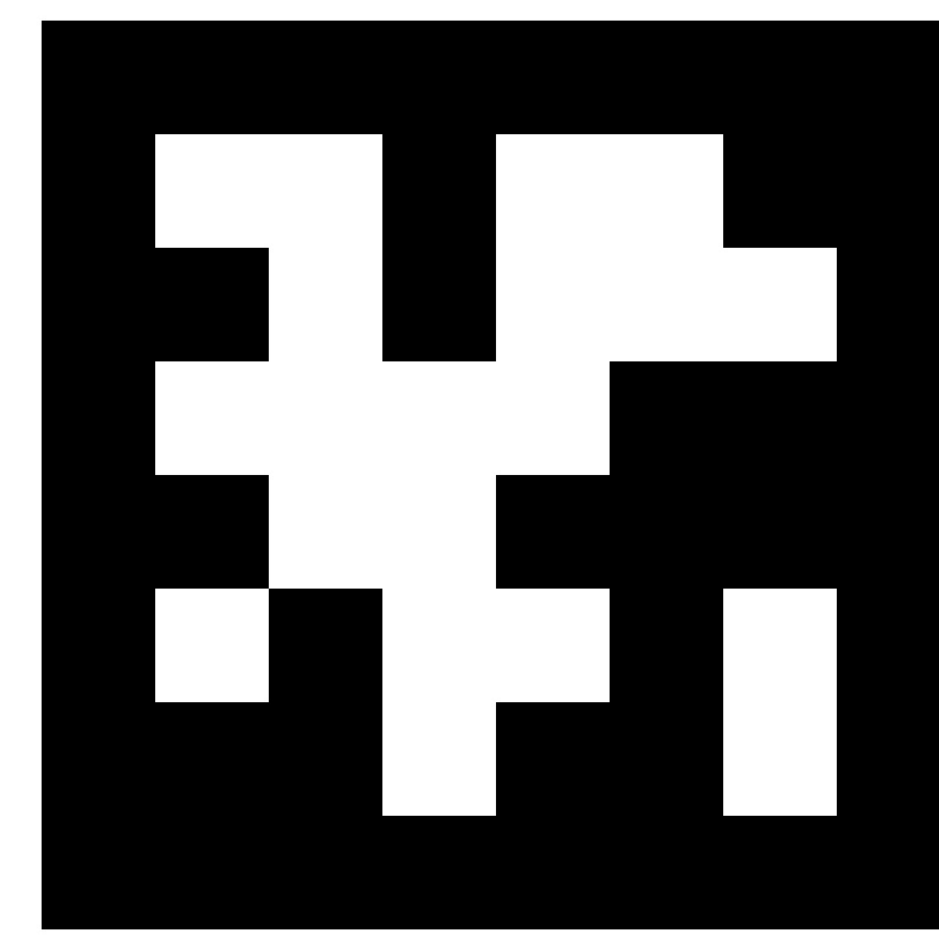


# XR ARENA

A multi-user and multi-application environment to simplify the development of mixed reality applications.

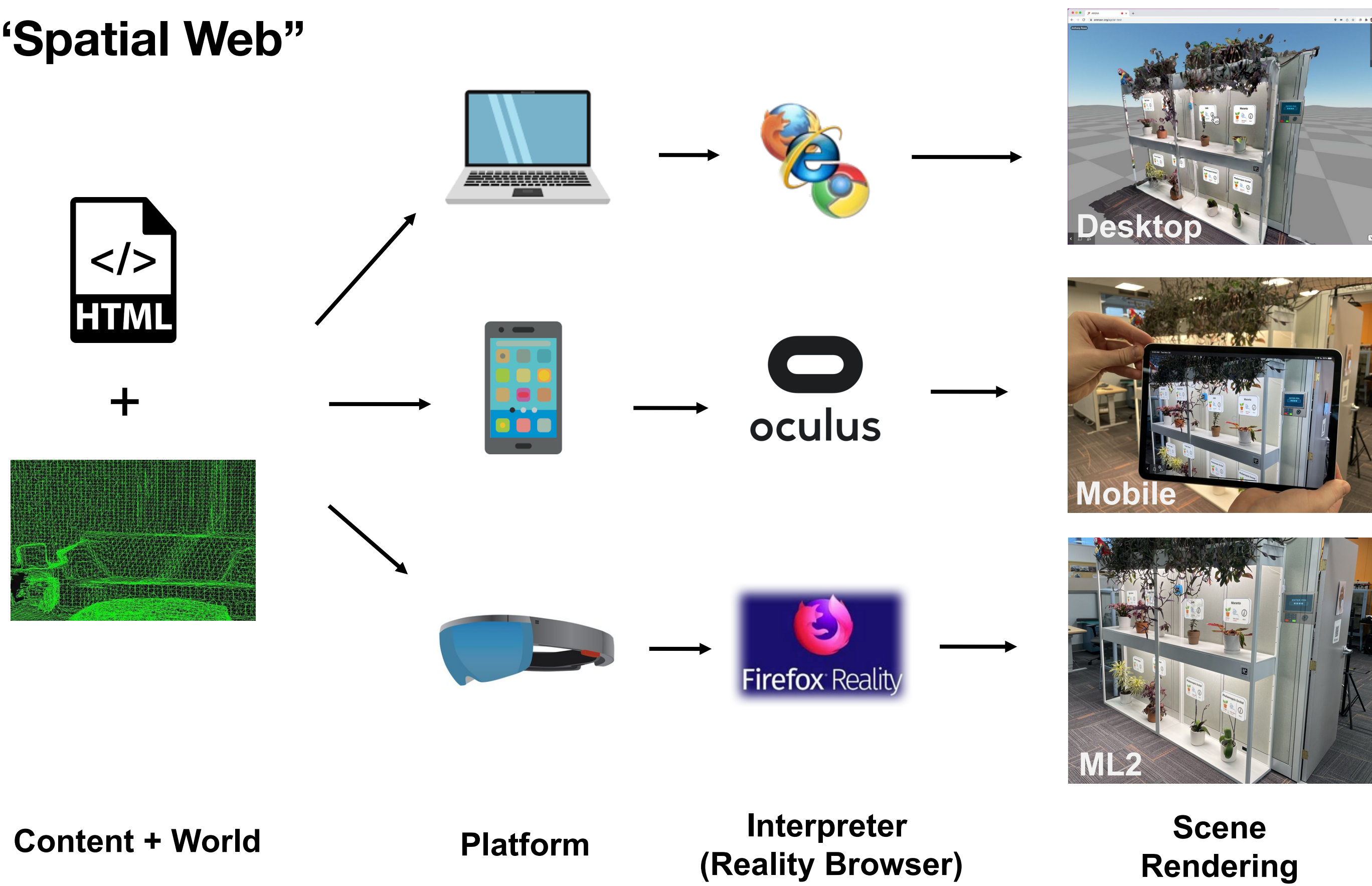


<https://arenaxr.org>

## XR Challenges

ARENA (**A**ugmented **R**eality **E**dge **N**etworking **A**rchitecture) is designed to address XR collaboration challenges by unifying access through **WebXR**, providing **distributed, programmable** scenes to experiment with **localization** techniques and ultimately **collaborative** volumetric rendering, which will be **scalable, secure, and efficient** across diverse **networks**.

## The "Spatial Web"



## Features

### Multi-platform AR, VR, XR

Seamless support for a spectrum of experiences through WebXR in:

- Tablets
- Phones
- Headsets
- Desktops

### Multuser

- Shared 3D environment
- Updated in real-time

### Multi-programming

Host user programs on any network connected device in:

- Python
- Unity

### Real-world Anchoring

- UWB
- OptiTrack
- AprilTags
- Easy integration

### Security

- Fine-grained access control
- Sandboxed user applications

### Spatial DNS

Geographic queries for:

- Content
- Assets
- Location markers
- Compute resources

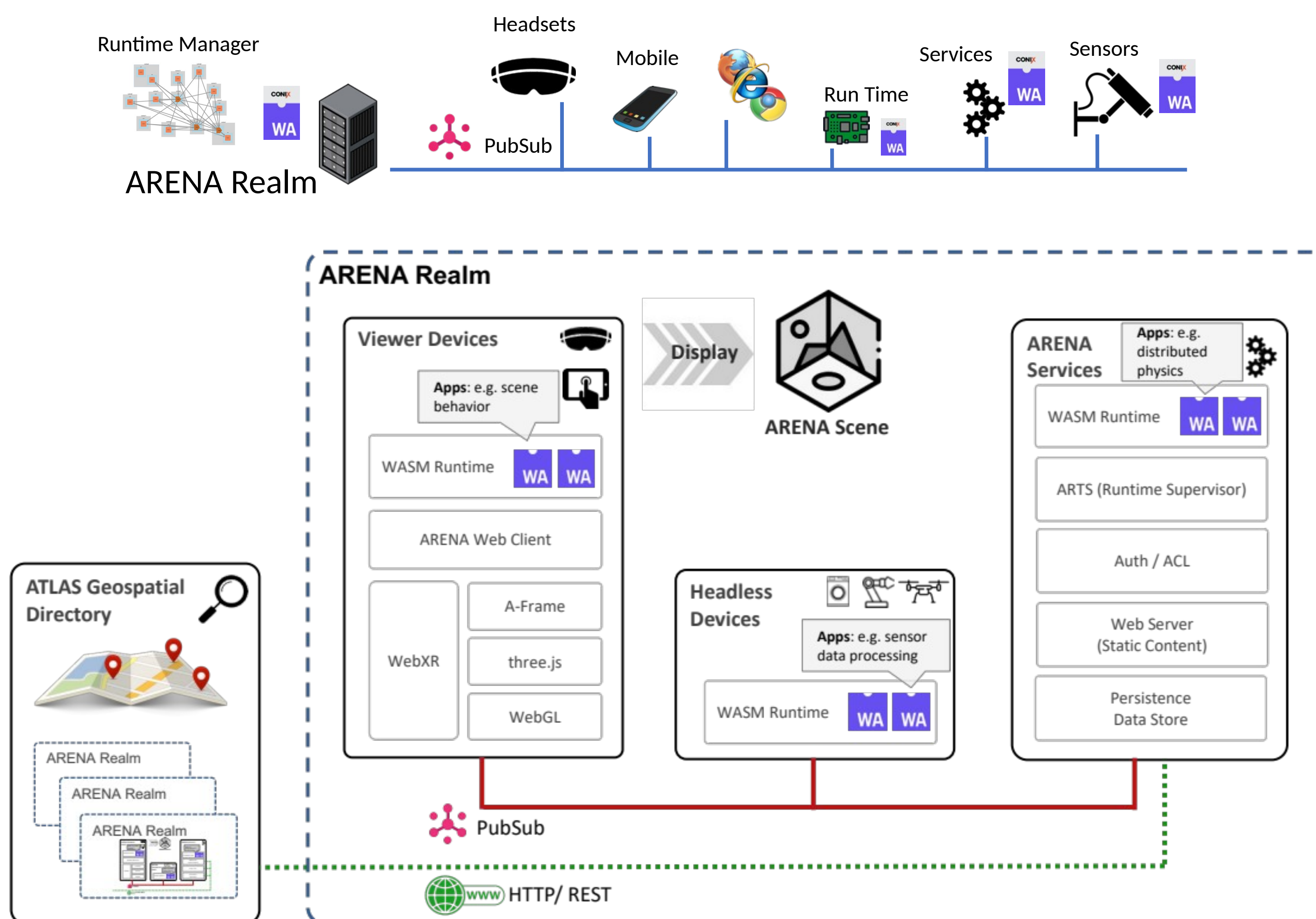
### Conferencing

- Video avatars
- Spatial audio
- Multi-room 360 Video

### More...

- LoD models
- Hybrid remote rendering
- Volumetric streaming

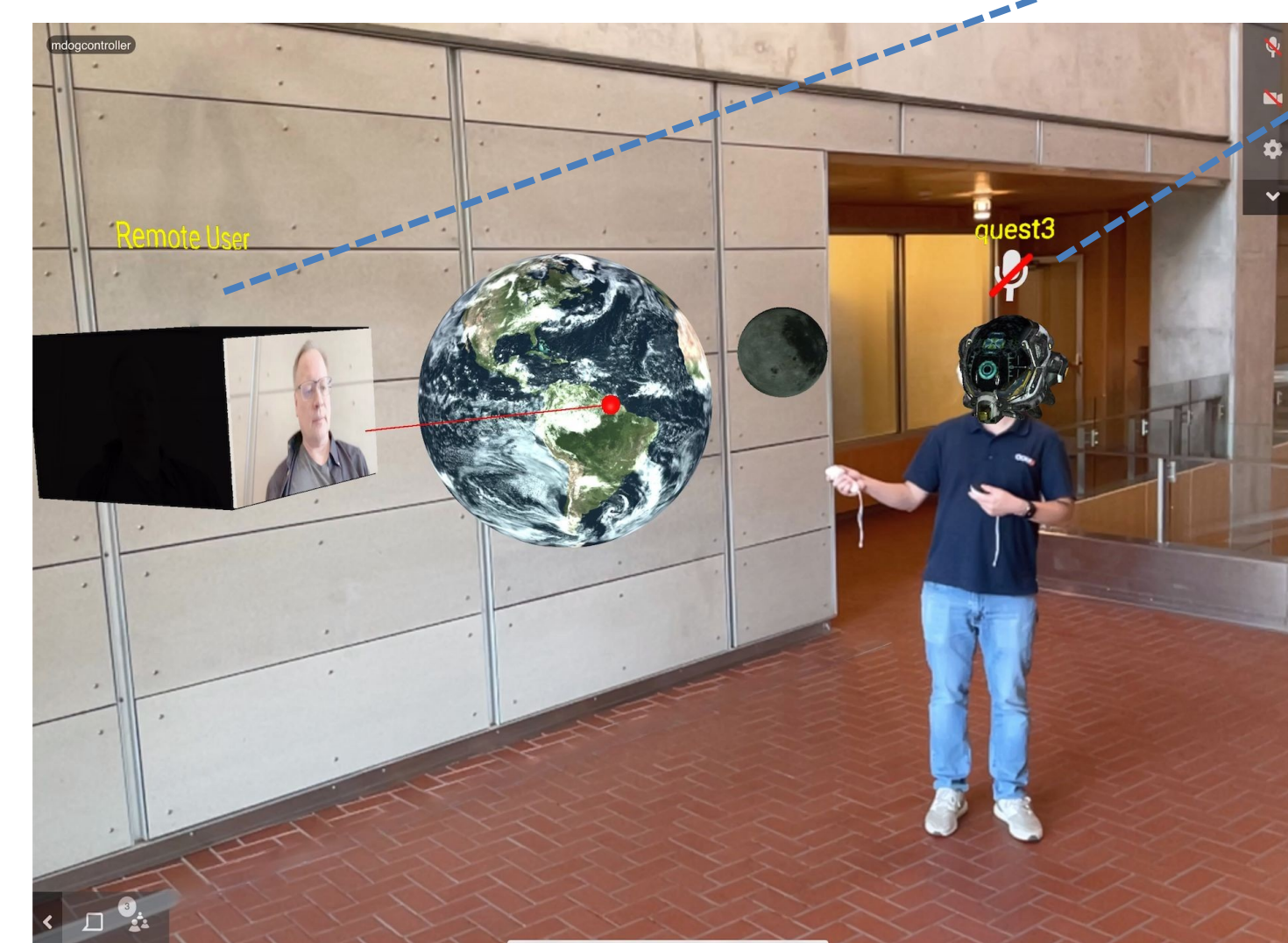
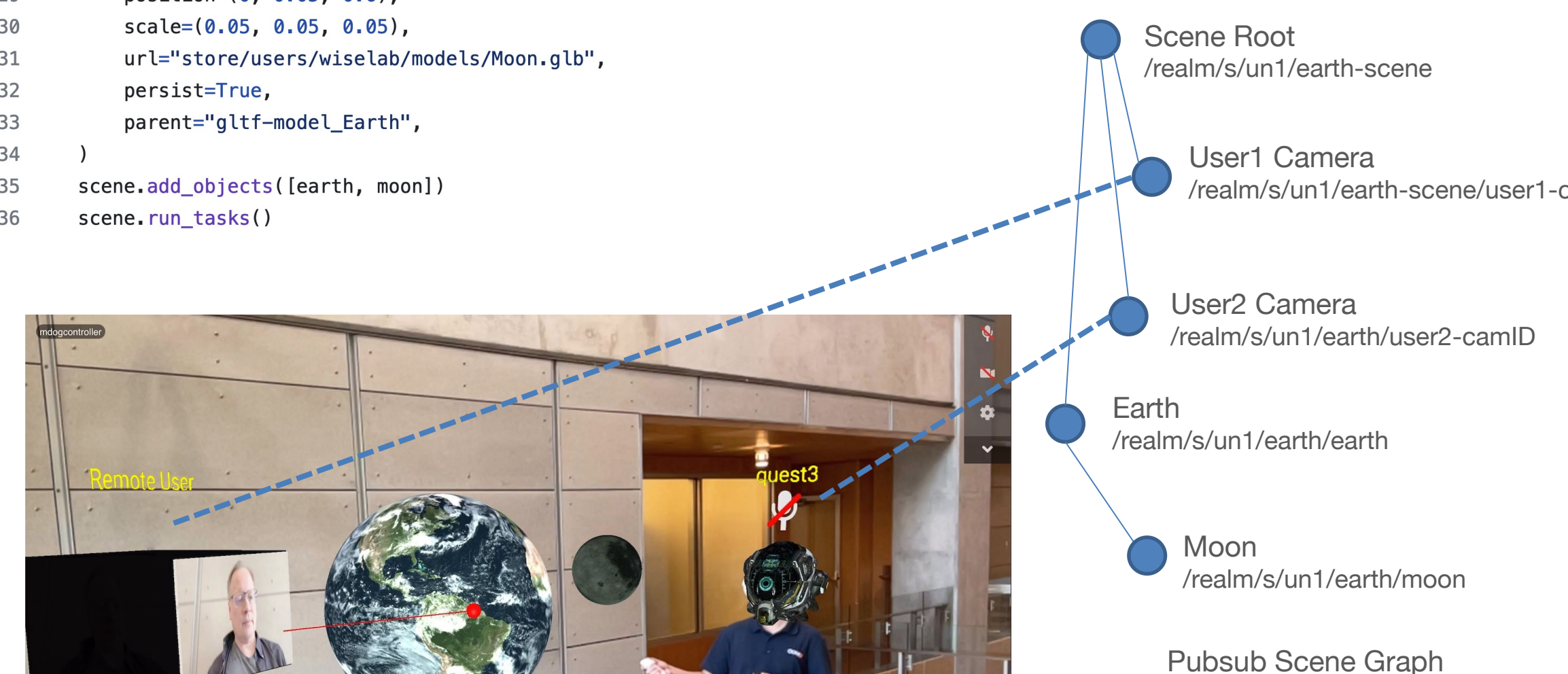
## Scalable Edge Architecture



## Programmable Interaction

```

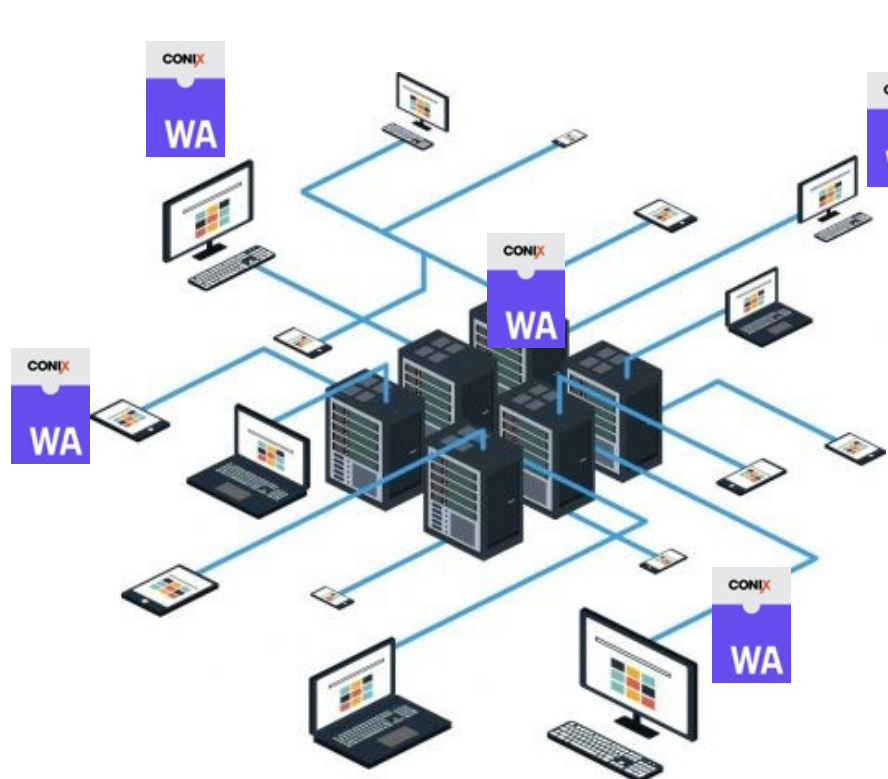
1 # earth-moon-laser.py
2 """ Simple program to animate the earth and moon with a laser pointer. """
3 from arena import *
4
5 def click_pointer(scene, event, msg):
6     """ Emit a 1-second laser line and target for each user click """
7     if event.type == "mousedown":
8         start = event.data.clickPos
9         end = event.data.position
10        start.y = start.y - 0.1 # emit below user frustum for visibility
11        line = ThickLine(path=(start, end), color=(255, 0, 0), lineWidth=5, ttl=1)
12        ball = Sphere(position=end, scale=(0.03, 0.03, 0.03), color=(255, 0, 0), ttl=1)
13        scene.add_objects([line, ball])
14
15 scene = Scene(host="arenaxr.org", scene="earth")
16 earth = GLTF(
17     object_id="gltf-model_Earth",
18     scale=(5, 5, 5),
19     url="store/users/wislab/models/Earth.glb",
20     clickable=True,
21     persist=True,
22     evt_handler=click_pointer,
23     animation=Animation(
24         property="rotation", end=(0, 360, 0), loop=True, dur=20000, easing="linear"
25     ),
26 )
27 moon = GLTF(
28     object_id="gltf-model_Moon",
29     position=(0, 0.05, 0.6),
30     scale=(0.05, 0.05, 0.05),
31     url="store/users/wislab/models/Moon.glb",
32     persist=True,
33     parent="gltf-model_Earth",
34 )
35 scene.add_objects([earth, moon])
36 scene.run_tasks()
    
```



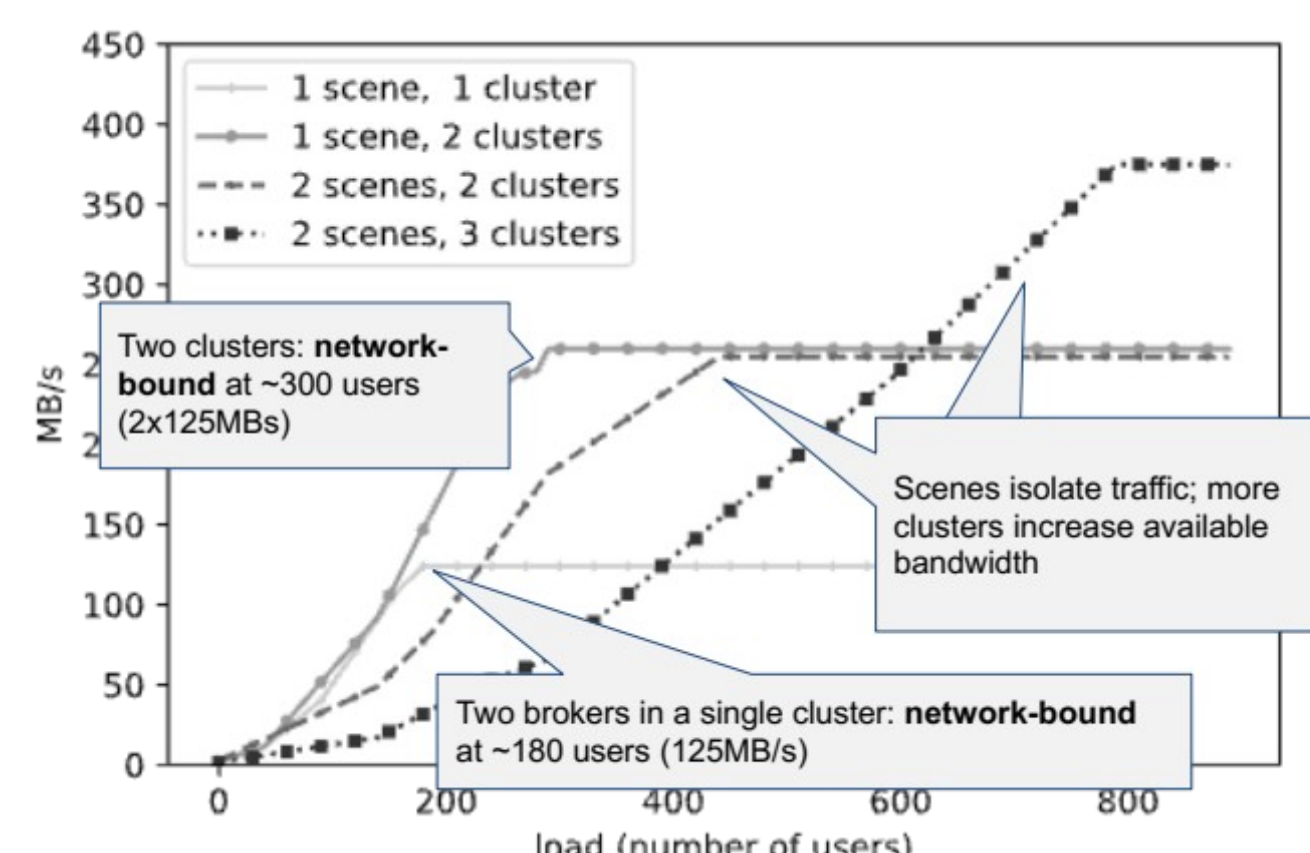
AR Scene

A geographical directory services, **Atlas**, allows users to find nearby content based on coarse location and then supports managing the data needed to link **Scene** content with the physical world. As users find local content, they are handed off to a **Realm**, a geographically distinct set of resources. Each realm has its own set of ARENA services.

ARENA includes a Wasm-based **Edge runtime environment** that provides a basis for agile programs that operate in the dynamic, distributed computing contexts we imagine for future XR applications.



Edge Runtime



## References

Edward Lu, Sagar B. K. Seetharam, Mallesh Dasari, Connor Smith, Srinivasan Seshan, Anthony Rowe, "RenderFusion: Balancing Local and Remote Rendering for Interactive 3D Scenes", 22nd IEEE International Symposium on Mixed and Augmented Reality (ISMAR), October, 2023, Sydney, Australia.

Mallesh Dasari, Edward Lu, Michael W. Farb, Nuno Pereira, Ivan Liang, Anthony Rowe, "Scaling VR Video Conferencing", IEEE VR, 2023

N. Pereira, A. Rowe, M. W. Farb, I. Liang, E. Lu and E. Riebling, "ARENA: The Augmented Reality Edge Networking Architecture", 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Bari, Italy, 2021, pp. 479-488.