X-Reality-Lab & mozXR Open-Source Framework

The X-Reality-Lab is the entirety of infrastructure – the room itself and the technology available in it. mozXR is the open-source framework (OSF), allowing you to create content for the X-Reality-Lab.

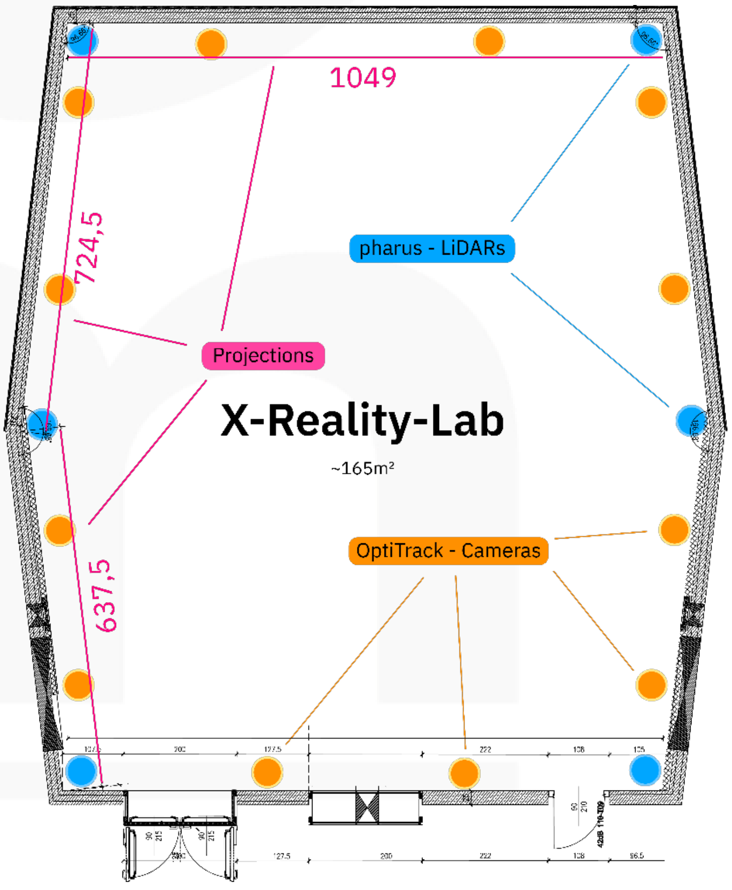
Possibilities and possible Scenarios

With mozXR in the X-Reality-Lab a broad variety of interactive audio-visual projects can be realized. Two tracking systems allow for the localization of people and objects within this large projection room. Four powerful PCs provide high processing and rendering performance and a sophisticated sound system provides high-quality sonification.

Interactive dance performances, co-located games and experiences and various kinds of generative audio-visual pieces are some examples of what's possible with mozXR in the X-Reality-Lab.

X-Reality-Lab Infrastructure

The X-Reality-Lab is an approximately 165 square meters hexagonal projection room.



Rendering and Projection

Five of the walls and the floor are covered with stereoscopic projection. Seven projectors are showing images, rendered by four PCs.

Sound System

The sophisticated sound system consists of an array of 60+ speakers and subwoofers.

Tracking

Two tracking systems cover the whole room and are available in the Unreal Engine OSF project.

pharus – 2D floor Tracking

pharus is a LiDAR based 2D tracking system which is integrated in the mozXR OSF.

- 6 LiDARs located on the floor in the corners and in front of the central wall projection
- It can track up to 20 people robustly and more. At a certain point occlusion between tracked people limit the performance of pharus.

OptiTrack – optical marker tracking

- The whole room is tracked with 12x OptiTrack Prime^x 120W cameras – mounted on ceiling.
- Occlusions near the wall are a natural limitation.

mozXR Open-Source Framework

With the mozXR Open-Source Framework comes a synchronized stereoscopic 3D cluster rendering setup withs plugins and integrations which ease the use of the technologies available in the X-Reality-Lab. It is developed for Unreal Engine, Unity Engine and TouchDesigner.

In all three cases the general setup is similar. One primary node is synchronized do secondary nodes. The exact nature of the synchronization varies between the three OSFs.

Mid 2025 the mozXR OSF Unreal Engine integration will be available. The integration in Unity Engine and TouchDesigner will follow.

The complete rollout of the Open-Source Framework including documentation and training materials is planned for the end of March 2026.

Rendering and Projection

In the X-Reality-Lab, on five projection surfaces (seven projections witch edge blending on wall and floor) images are rendered frame synchronous in sequential stereoscopic 3D. Room and infrastructure configuration files are used so the created content is rendered as desired. The projection setup (the geometry of projections screens in the room) and the cluster setup (settings for the render PCs, such as IP address and which image to render) are defined within these configs. Different configs allow for different projection setups in a given room. Configs can be edited and new configs for further rooms – such as the Deep Space 8K in the Ars Electronic Center in Linz – can be added.

World/Object Synchronization

While the rendering is synchronized automatically. The synchronization of virtual objects, data, events and so on must be managed to a certain extent. Synchronization mechanisms are present in all three engines and examples are present in the template projects.

Simulation

A simulation environment allows the validation of both, the visual output and technical correctness. It helps content creators to have a preview of their projects on their desktop and simulate the distributed cluster setup to a certain extent. This allows the detection of potential issues in an early stage of a project. Also inputs and interactions can be simulated. A pharus simulator allows interaction tests and OptiTrack data can be played back in Motive (proprietary Software).

Sound System

There are two distinct ways of how to get audio output in the X-Reality-Lab using mozXR.

1. Sending an audio signal to the sound system. This can be Stereo, 5.1, 7.1. This can differ between the different engines.
2. Sending OSC commands to an external DAW which can leverage the full potential of the sound field system installed in the X-Reality-Lab. The OSC data can include the position of and an id of a given sound.

Tracking

Integrations are available for both tracking systems and included in the template projects. The tracking technologies are utilized in the three artistic exploration projects (templates) to a different extent.

2D floor Tracking

The 2D floor tracking system pharus provides the position of people and objects within the X-Reality-Lab. pharus data is made available in the network via multicast and therefore can be received by any connected device.

OptiTrack – optical marker tracking

OptiTrack is a marker-based tracking system. It provides very precise low latency 3d tracking. An integration is available in the three engines.

- Rigid Body Tracking
- Passive reflective or active Marker Tracking

mozXR Unreal Engine Open-Source Framework

Frame synchronous stereoscopic rendering and world synchronization is achieved using Unreal Engines nDisplay.

Rendering and Projection

mozXR for Unreal Engine comes with nDisplay configs for the X-Reality-lab. Different configs exist for different use cases. There will be one which allows simulated testing on the desktop (see Simulation).

World/Object State Synchronization

The synchronization of the virtual world and it's entities is also handled with nDisplay. Different solutions are present in the template project.

- Transformation synchronization using *Cluster Scene Components:* Easiest way which lets users decide which objects need to be synchronized between nodes/projections.

- Custom synchronization using *Cluster Events*: Provides more control and allows the synchronization of arbitrary data between nodes/projections.
- Particle system synchronization: Deterministic Niagara systems can be synchronized between nodes/projections.

Simulation

A visual preview of how the project will look like in the target room is equally important as technical verifications.

- Visual preview: as the geometry of the room is known due to the nDisplay config it is possible to get a preview of the rendered result within the Unreal Engine Editor.
- By using Switchboard locally, the whole cluster setup (4 render PCs in the X-Reality-Lab) can be started virtually. The rendered output and data synchronization and rendering can be tested on the desktop close to the real-life setup.

Sound System

mozXR Unreal OSF provides:

1. Audio output in Stereo, 5.1, 7.1.
2. Sending OSC commands to an external DAW.

Tracking

pharus – 2D floor Tracking

the pharus plugin receives data via multicast and is also included in the project.

OptiTrack – optical marker tracking

OptiTrack data is received via a Live Link Plugin.

How to work with mozXR Unreal Engine OSF

- Unreal Engine 5.5.x (TBD) installed.
- Clone the repository from GitHub (not available yet).
- Follow instructions in readmes and learn how to use certain parts of the system using the project template. The complete documentation for this project follows later this year and will be officially published beginning of 2026.

mozXR Unity Engine Open-Source Framework

The Unity Engine integration of mozXR leverages the features of nDisplay by communication with an Unreal Engine instance. Thereby it achieves synchronization and stereoscopic 3D rendering. Inter process communication (IPC) between these engines enable the synchronized flow of input and output data between the different render PCs in the cluster.

Rendering and Projection

Unity renders in stereo 3D, Unreal Engine displays the rendered frames in synced stereo.

World/Object State Synchronization

The Unity primary node processes all inputs and performs all simulations. A synchronization mechanism allows content creators to decide which objects or variables need to be synchronized. Via IPC the data is communicated to Unreal Engine and synchronized via nDisplay.

Simulation

Both, the visual results, as well as technical validation to a certain extent can be simulated.

Sound System

The mozXR Unity OSF provides:

1. Audio output in Stereo, 5.1, 7.1.
2. Sending OSC commands to an external DAW.

Tracking

pharus – 2D floor Tracking

the pharus integration receives data via multicast and is also included in the project.

OptiTrack – optical marker tracking

OptiTrack tracking data is available in mozXR Unity applications.

How to work with mozXR Unity Engine OSF

- Unity Engine 6000.xxx (TBD) installed.
- Download the mozXR Unreal Engine for Unity render and sync application
- Clone the mozXR Unity Engine OSF repository from GitHub (not available yet).
- Follow instructions in readmes and learn how to use certain parts of the system using the project template. The complete documentation for this project follows later this year and will be officially published beginning of 2026.

mozXR TouchDesigner Open-Source Framework

The Unity Engine integration of mozXR leverages the features of nDisplay by communication with an Unreal Engine instance. Thereby it achieves synchronization and stereoscopic 3D rendering. Inter process communication (IPC) between these engines enable the synchronized flow of input and output data between the different render PCs in the cluster.

Rendering and Projection

TouchDesigner .tox files are loaded in an Unreal Engine application using TouchEngine. This enables synchronous stereoscopic 3D rendering.

World/Object State Synchronization

The mozXR TouchDesigner implementation uses OSC synchronization and SyncOut for data synchronizations.

Simulation

Both, the visual results, as well as technical validation to a certain extent can be simulated.

Sound System

The mozXR TouchDesigner OSF provides:

3. Audio output in Stereo, 5.1, 7.1.
4. Sending OSC commands to an external DAW.

Tracking

pharus – 2D floor Tracking

the pharus integration receives data via multicast and is also included in the project.

OptiTrack – optical marker tracking

OptiTrack tracking data is available in mozXR Unity applications.

How to work with mozXR TouchDesigner OSF

- Install TouchDesigner v202x.xxx (TBD)
- Download the mozXR TouchDesigner OSF packages consisting of:
  - Touch Designer base project and/or template project
  - mozXR Unreal Engine – TouchEngine application