sensics

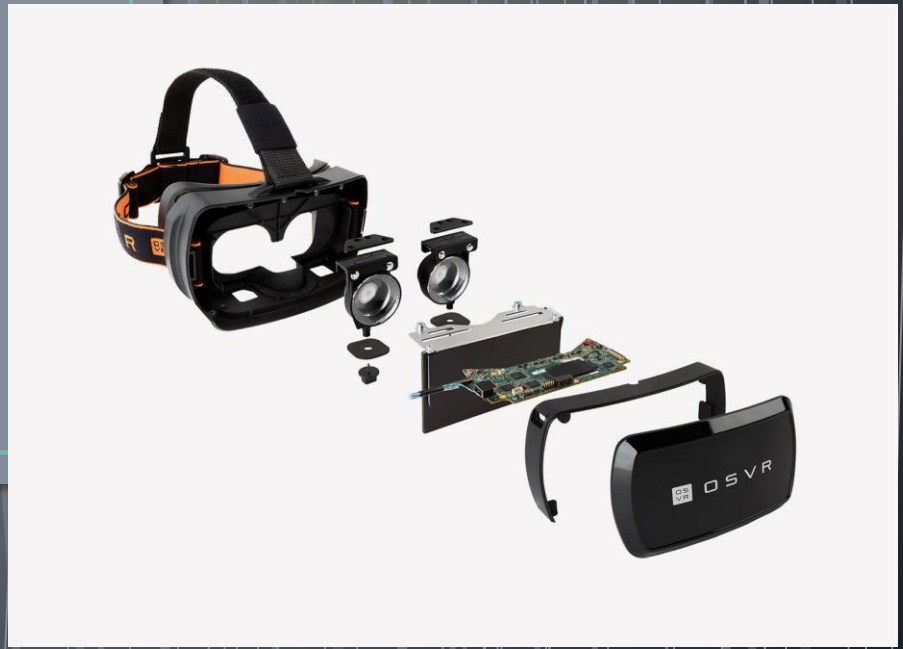
OSVR

Boston VR Meetup

Jul 2015

Yuval Boger, CEO/VRguy, Sensics

www.Sensics.com, www.vrguy.net



Corporate Fact Sheet

- Sensics designs, builds and delivers cutting-edge HMDs
- Substantial expertise, know-how and IP in VR
- In business since 2003; Over 200 customers world-wide

Markets Served

HMD Expertise

- Sensors
- Optics
- Electronics and firmware
- Industrial design
- Human factors
- Open Source

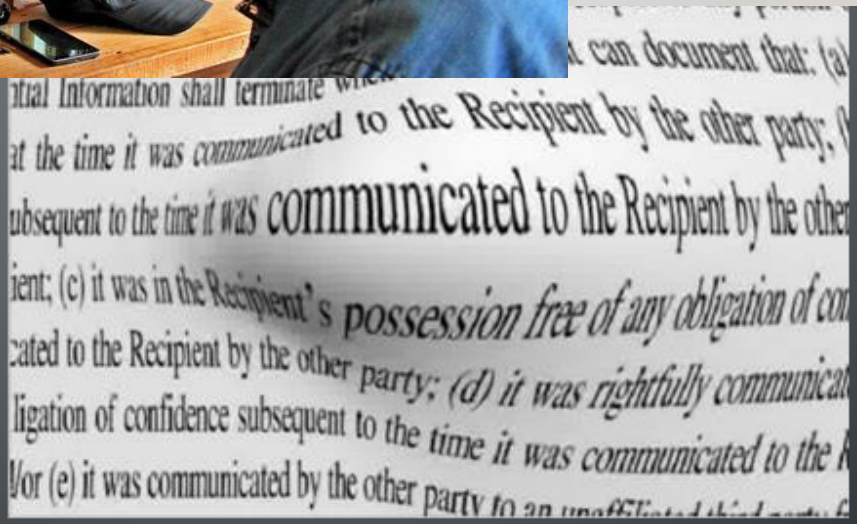
Gaming goggles (white label)

Low-vision devices (white label)

Professional (Sensics brand)

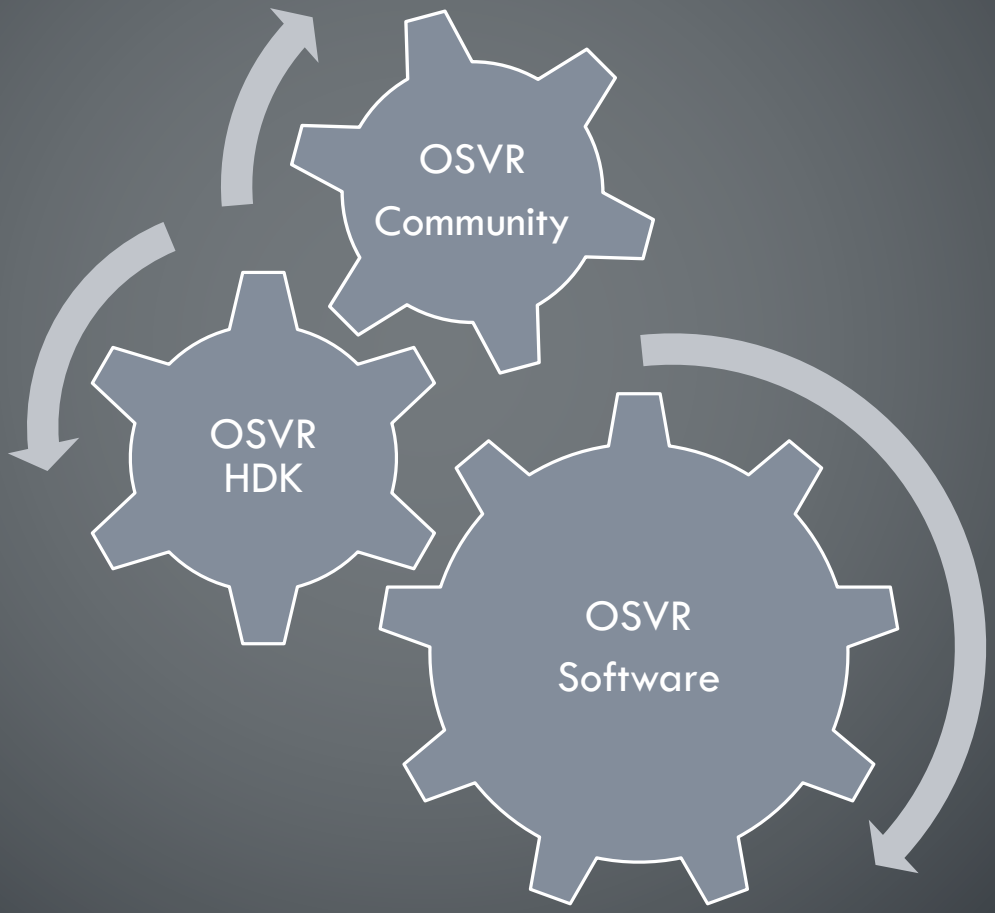
Military Products

Products for People with Vision Disabilities



Gaming Goggles

What is OSVR?

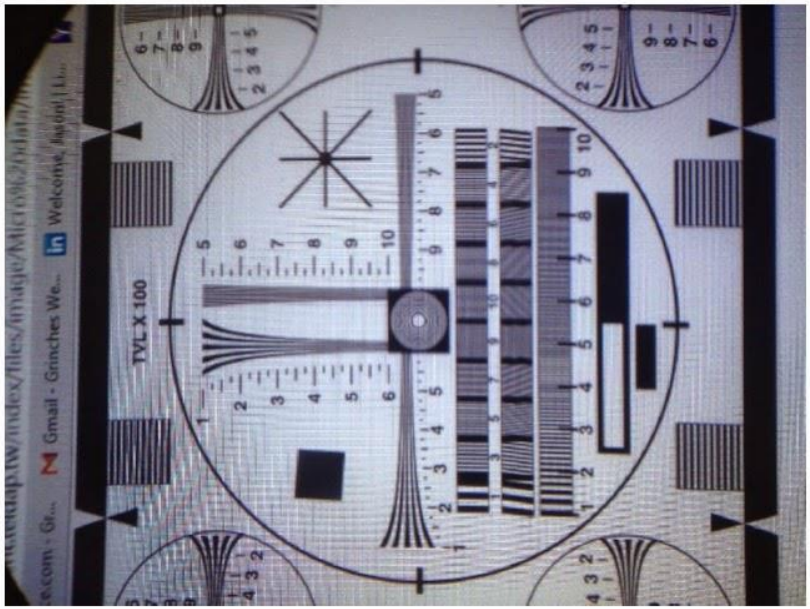



OSVR

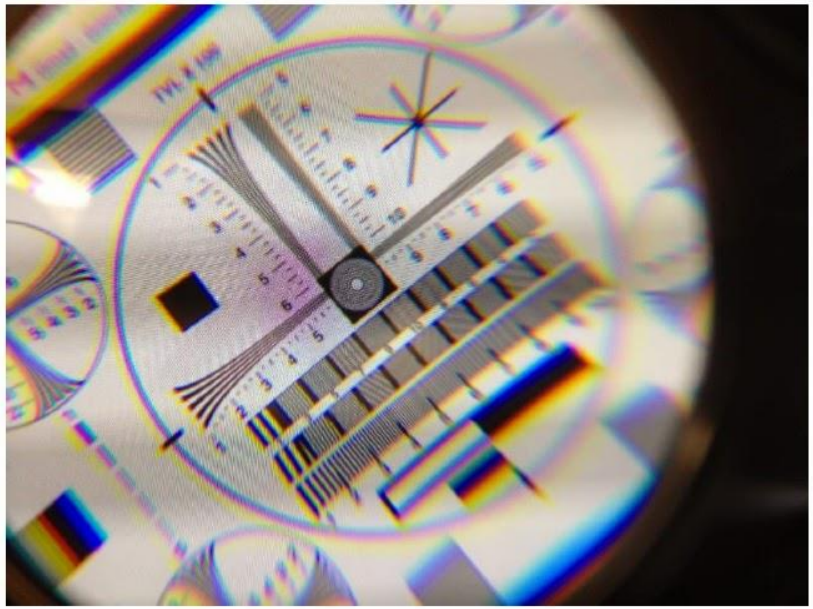
OSVR HDK Optics

- Very low geometric and color distortion
- Sharp image throughout

OSVR HDK: Dual-Element Optics



Typical Single-Element Design



What Is the OSVR™ API?

The OSVR™ API is a multi-platform, standardized interface to virtual reality devices and peripherals.

Designed to encourage adoption of and innovation in AR/VR.

With the OSVR API, developers can:

- Discover, configure and operate a wide range of devices through standardized, abstract interfaces
- Mix-and-match devices (e.g. hand trackers) with analysis plugins (e.g. gesture engines)
- Write once, run on many hardware/platform configurations

OSVR API License

- Software available under Apache Software License, version 2.0
  - http://osvr.github.io
  - Can change and redistribute
  - Can use proprietary plugins

- Choice of license intended to encourage participation of both academic institutions and commercial firms

OSVR Developer

Other OSVR Sites

Welcome to Open Source Virtual Reality.

OSVR is an open-source software platform for VR/AR applications.

Building with OSVR
and Developer Downloads

Using OSVR
and User Downloads

Contributing to OSVR
and Project Directory

Technical White Papers

- An Introduction to OSVR (Revised March 2015)
- Migrating Unity applications from Oculus to OSVR (revised March 11, 2015)
- Using External VRPN Devices in OSVR (April 27, 2015)

Slides and Speaker Notes from Presentations

Mailing Lists and Newsletters

Maintained at https://github.com/OSVR/OSVR.github.io

guidelines.

- If you're in doubt as to where your issue/request/contribution best fits, file a support ticket and we'll point you in the right direction.

Directory of Projects

Note that in case you can't find the project you're looking for below, you can access a full list of projects in the OSVR organization (note that this link includes a filter to exclude projects forked by OSVR).

OSVR-Core

The core libraries, applications, and plugins of the OSVR software platform.

- Contributing Guidelines
- Issue Tracker
- Windows binary downloads
- Doxygen docs for library users
- Internal Doxygen docs for contributors
- Getting started with core development

OSVR-HDK

Production file for OSVR HDK.

- Issue Tracker

Managed-OSVR

Managed code (.NET) wrapper for OSVR ClientKit.

- Contributing Guidelines
- Issue Tracker

OSVR-Unreal

Integration of OSVR with the Unreal Engine.

- Contributing Guidelines
- Issue Tracker

OSVR-Unity

Package for authoring OSVR experiences with Unity.

- Contributing Guidelines
- Issue Tracker
- Windows binary downloads
- Video showing how to integrate OSVR into a Unity project
- White paper: migrating Unity apps to OSVR

OSVR-Unity-Palace-Demo

A simple example OSVR Unity 4 project.

- Issue Tracker

JeroMiya/OSVR-MonoGame

Integration of OSVR with the MonoGame Engine.

- Issue Tracker

Distortionizer

Tool for determining distortion parameters of arbitrary HMDs, and a corresponding set of shaders to correct that distortion.

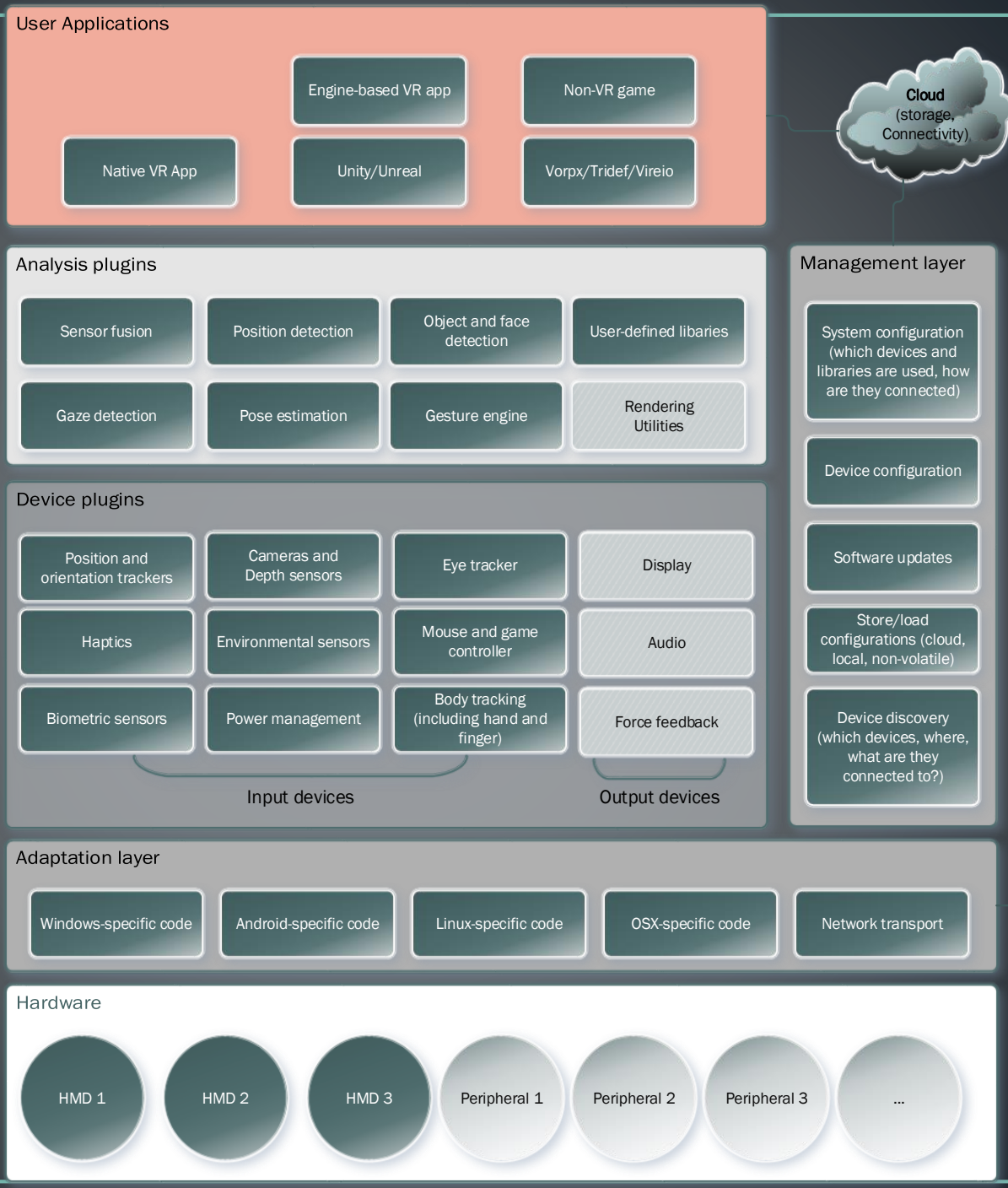
- Contributing Guidelines
- Issue Tracker
- Releases (including standalone Windows binaries of the Vizard-based shader tester)

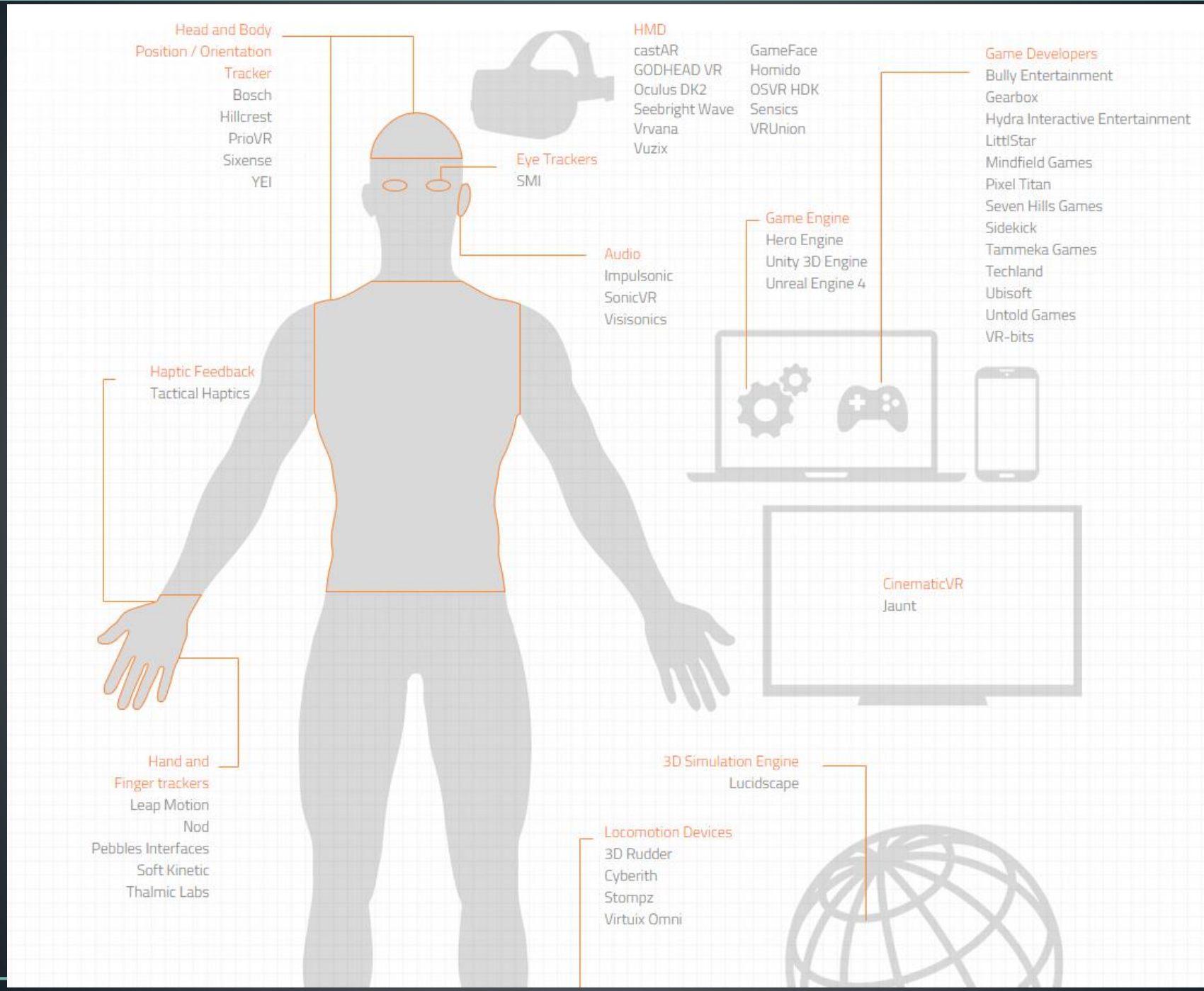
OSVR-Tracker-Viewer

OSVR-JSON-Editor

OSVR-JSON-Schemas

Open architecture supports additional plugins




Head and Body
Position / Orientation
Tracker
Bosch
Hillcrest
PrioVR
Sixense
YEI
HMD
castAR
GODHEAD VR
Oculus DK2
Seebright Wave
Vrvana
Vuzix
GameFace
Homido
OSVR HDK
Sensics
VRUnion
Eye Trackers
SMI
Audio
Impulsonic
SonicVR
Visisonics
Game Developers
Bully Entertainment
Gearbox
Hydra Interactive Entertainment
LittlStar
Mindfield Games
Pixel Titan
Seven Hills Games
Sidekick
Tammeka Games
Techland
Ubisoft
Untold Games
VR-bits
Game Engine
Hero Engine
Unity 3D Engine
Unreal Engine 4
Haptic Feedback
Tactical Haptics
CinematicVR
Jaunt
Hand and
Finger trackers
Leap Motion
Nod
Pebbles Interfaces
Soft Kinetic
Thalmic Labs
3D Simulation Engine
Lucidscape
Locomotion Devices
3D Rudder
Cyberith
Stompz
Virtuix Omni

OSVR Concepts

- Device
- Interface
- Plugin

Devices

- A **device** is a physical entity such as:
  - *Head orientation sensor*
  - *Razer Hydra*

Interfaces

- **Interfaces** are the primitive "pipes of data“
- A *Device* exposes one or more *interfaces*

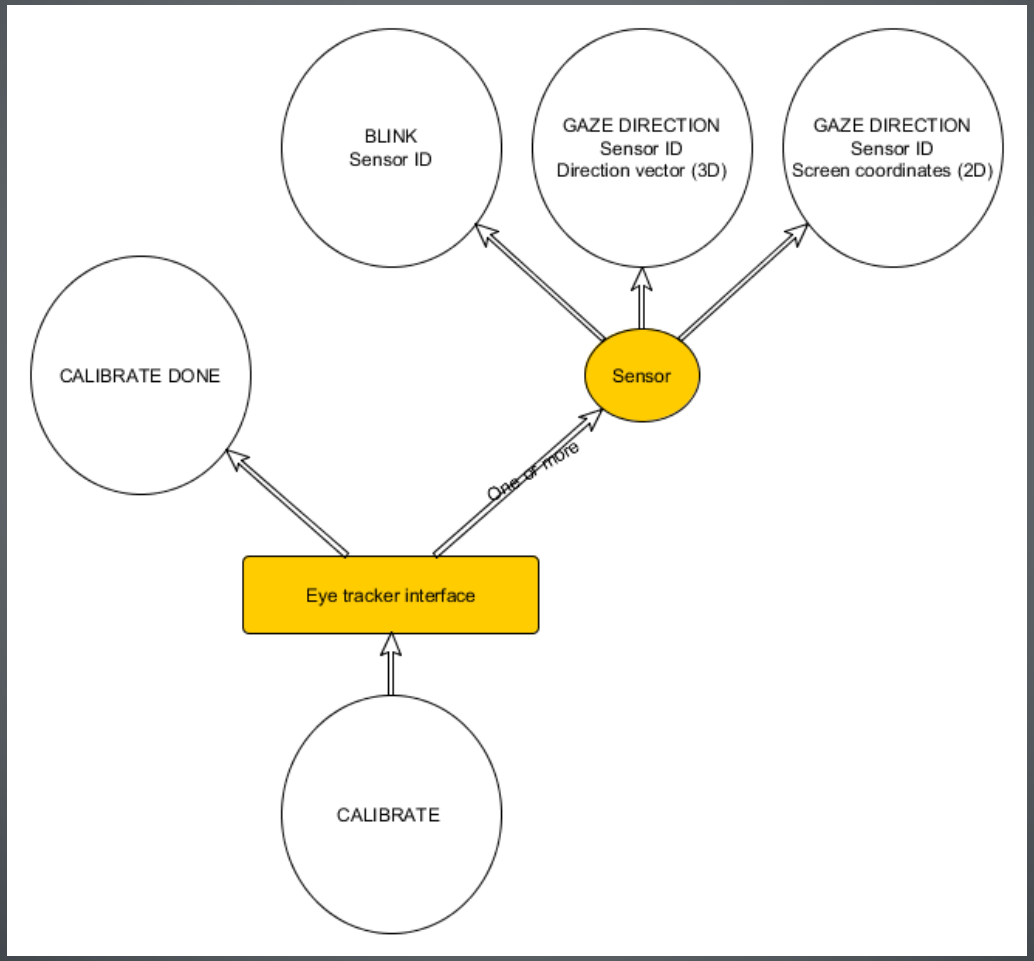
An *orientation sensor* exposes:

- Orientation interface
- Linear acceleration interface

A *Razer Hydra* exposes:

- XYZ position for left hand
- XYZ position for right hand
- Orientation for left hand
- Orientation for right hand
- Button set for left hand
- Button set for right hand

Eye Tracker Interface Class



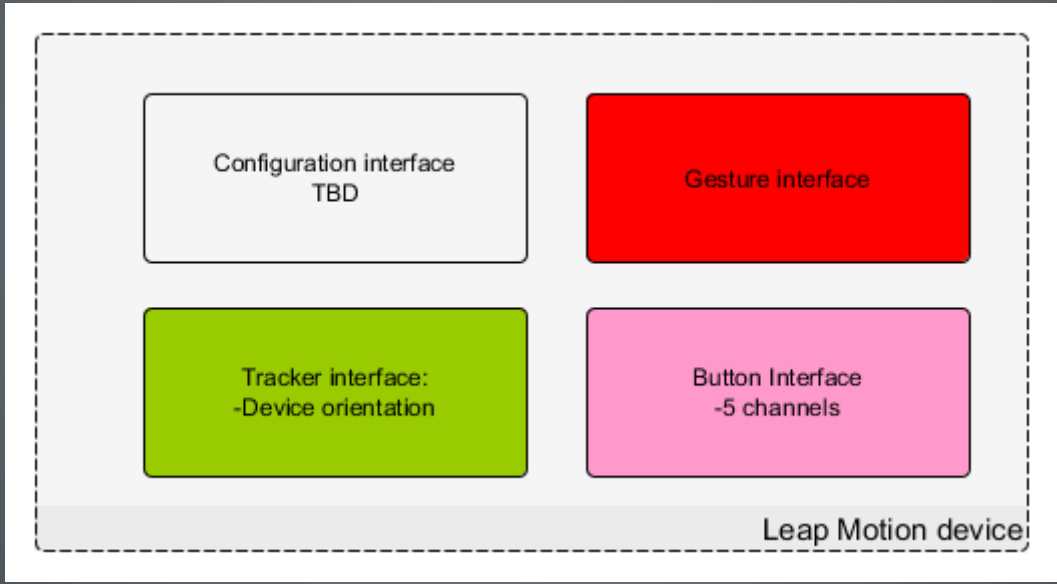
Working with an Interface

- Can use one or both of the following methods:

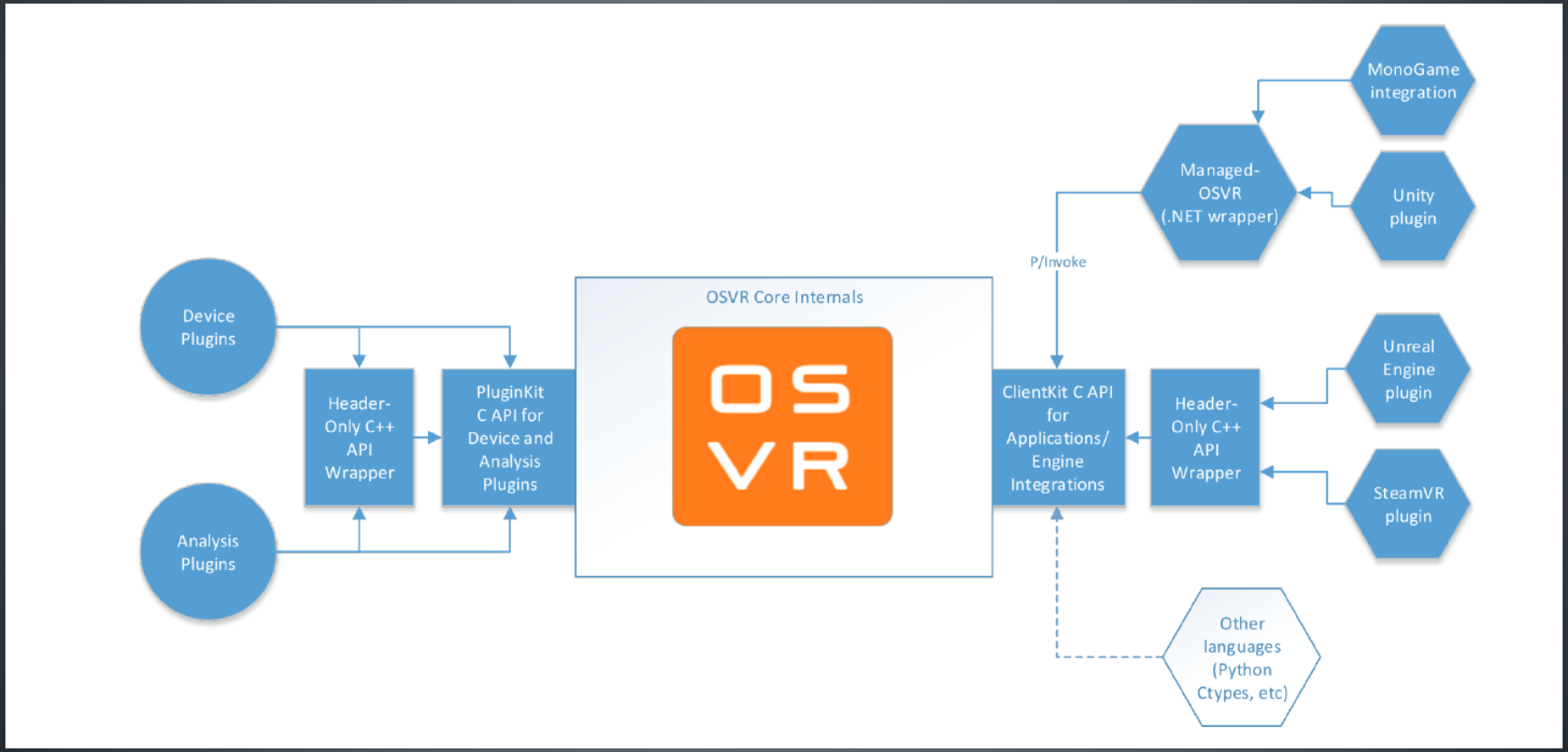
1. Synchronous: blocking read or write
2. Asynchronous: register a callback for a particular event

- For instance, a game might decide to:
  - Sample head tracker orientation in main loop using synchronous call
  - Register a callback for event-based call on “fire” button

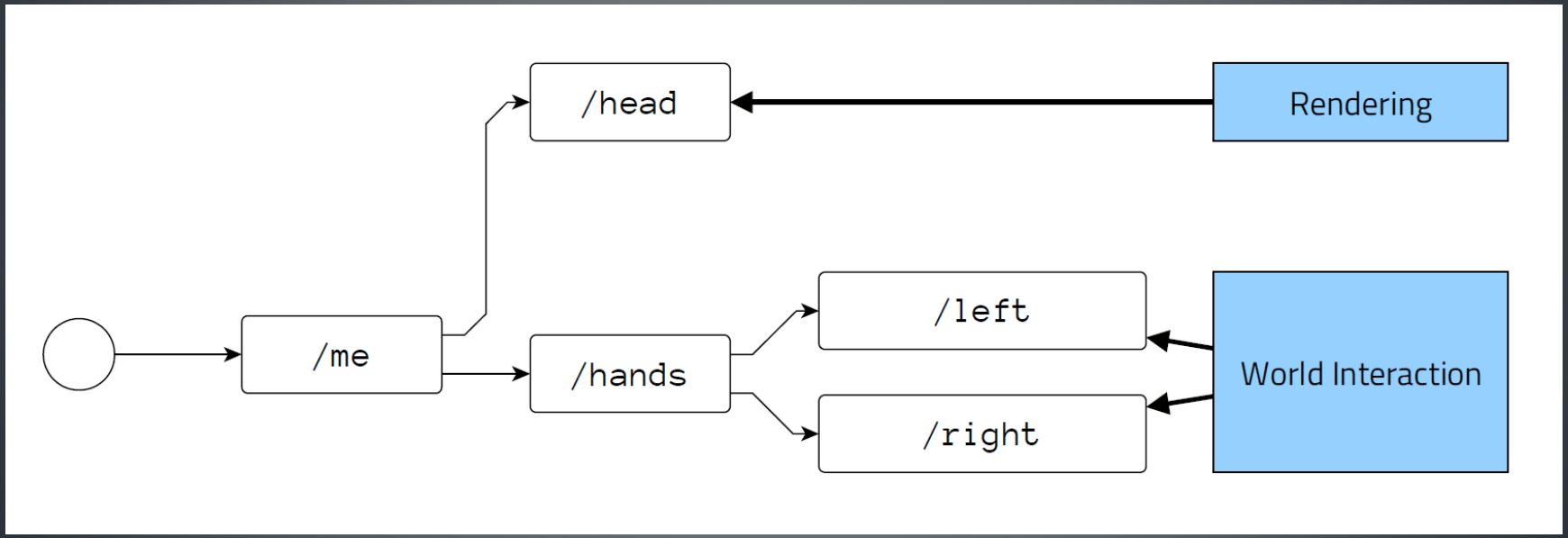
Device Interface – Nod Ring



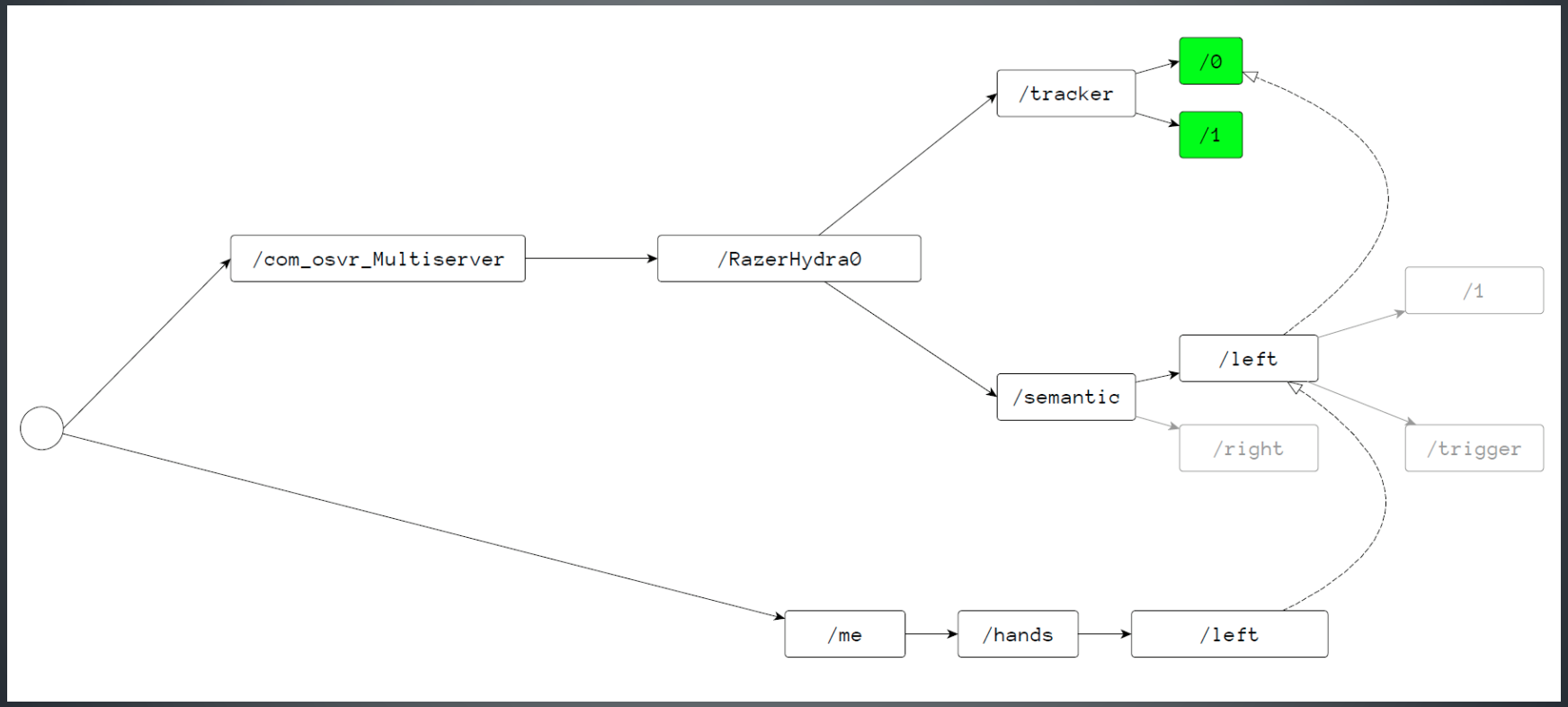
Technical View of the System



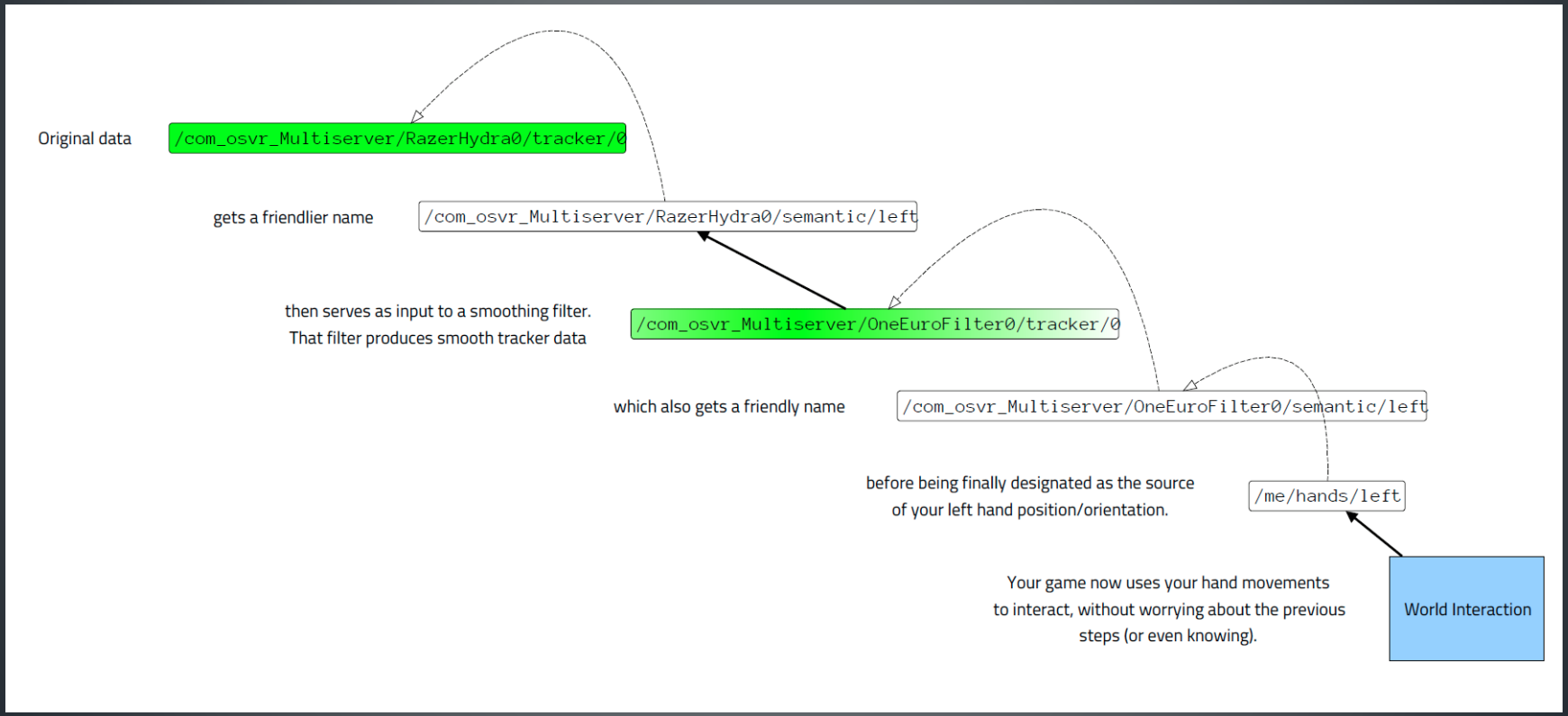
Semantic Paths



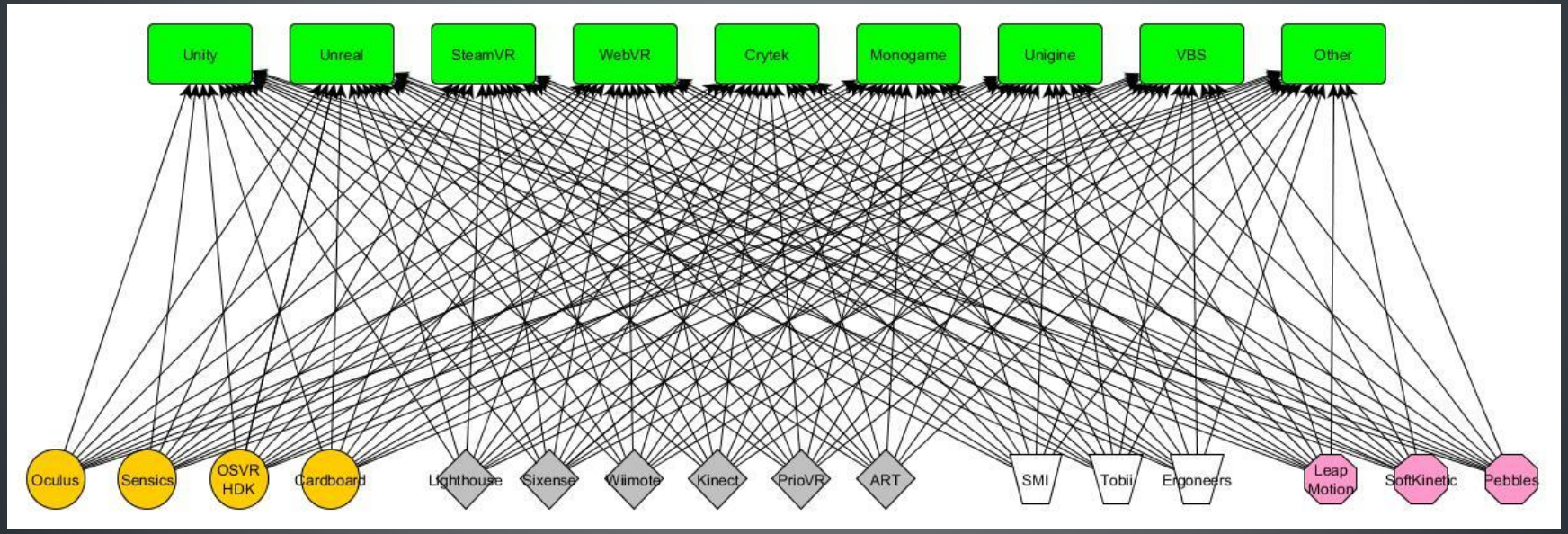
Map to the Hardware



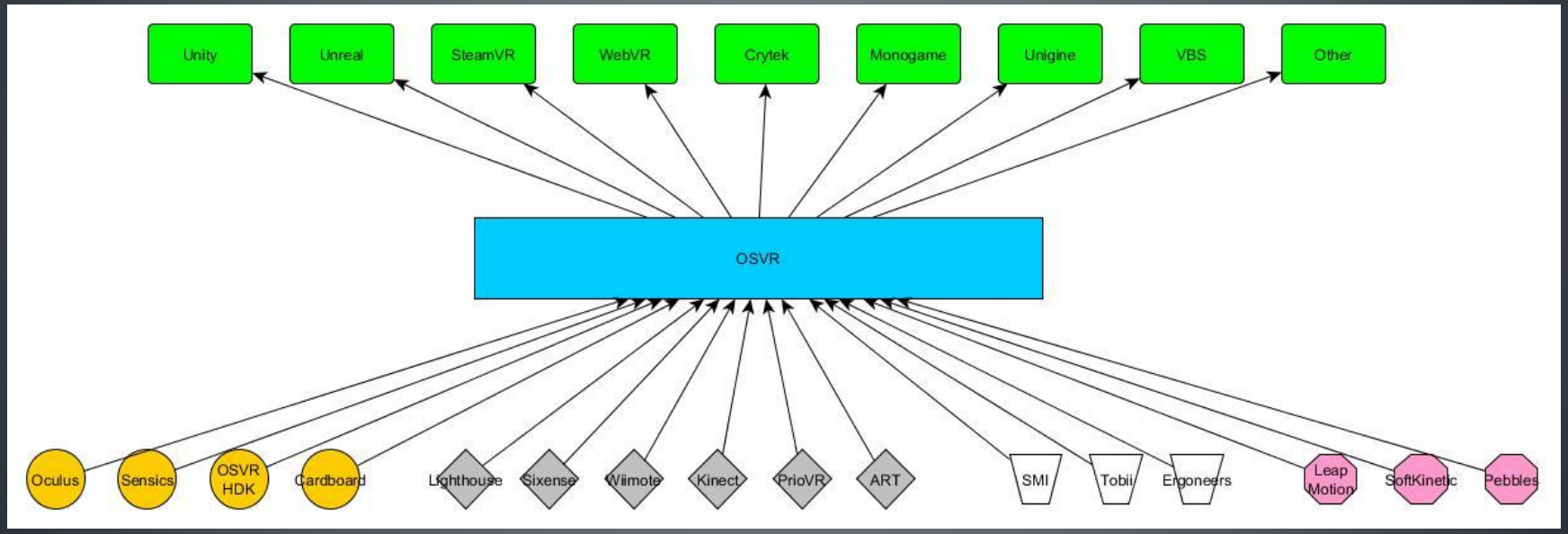
Could also map to plugins



Before OSVR


Unity
Unreal
SteamVR
WebVR
Crytek
Monogame
Unigine
VBS
Other
Oculus
Sensics
OSVR HDK
Cardboard
Lighthouse
Sixense
Wiimote
Kinect
PrioVR
ART
SMI
Tobii
Ergoneers
Leap Motion
SoftKinetic
Pebbles

With OSVR


Unity
Unreal
SteamVR
WebVR
Crytek
Monogame
Unigine
VBS
Other
OSVR
Oculus
Sensics
OSVR HDK
Cardboard
Lighthouse
Sixense
Wiimote
Kinect
PrioVR
ART
SMI
Tobii
Ergoneers
Leap Motion
SoftKinetic
Pebbles

Summary of OSVR Advantages

- Supports multiple devices, operating systems, game engines
- Unified, device-independent programming model
- Optimized game engine interfaces
- Increasingly complete set of capabilities
- Does not force a particular app distribution method
- Free and open source