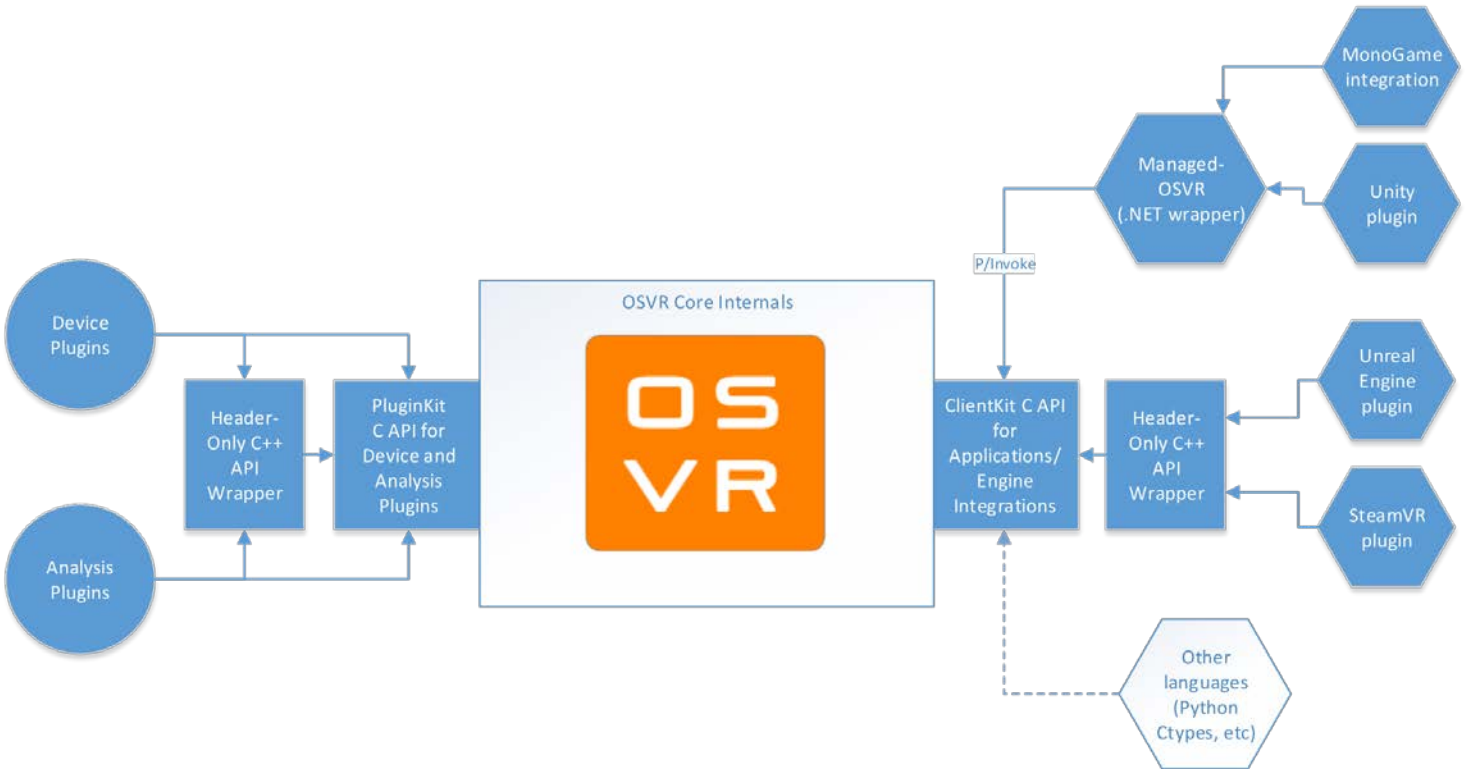
OSVR Software Framework

Overview and Semantic Path Tree

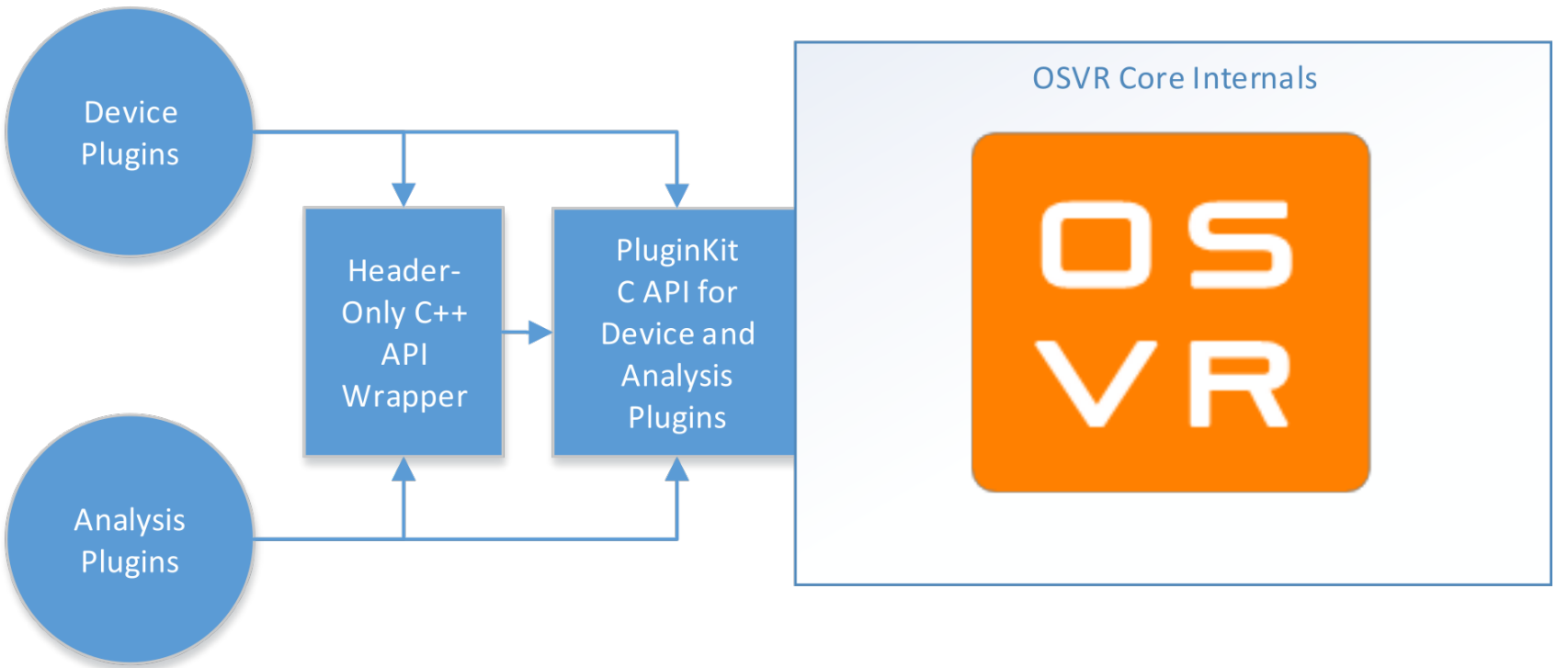
Ryan A. Pavlik, PhD
Sensics, Inc.

July-2015

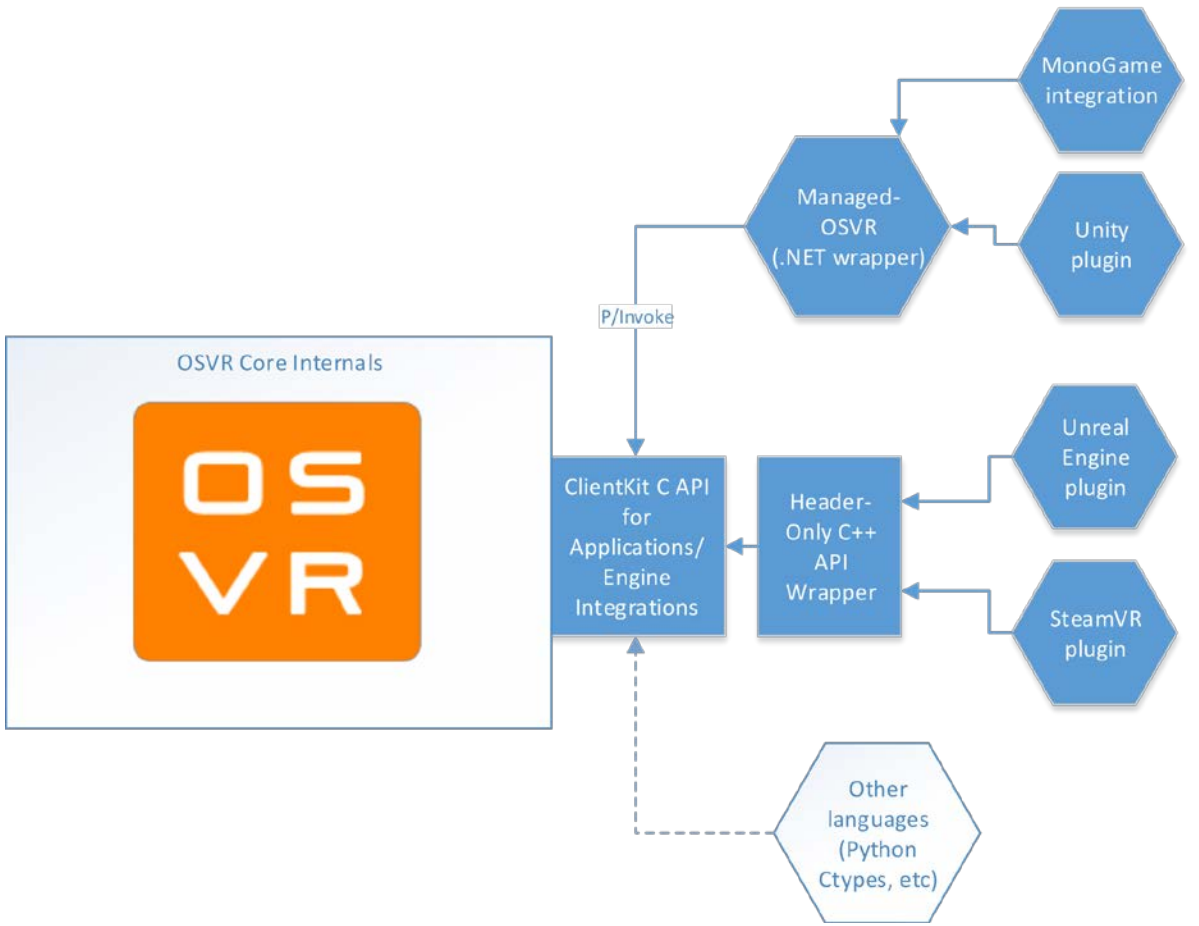
A View of the System



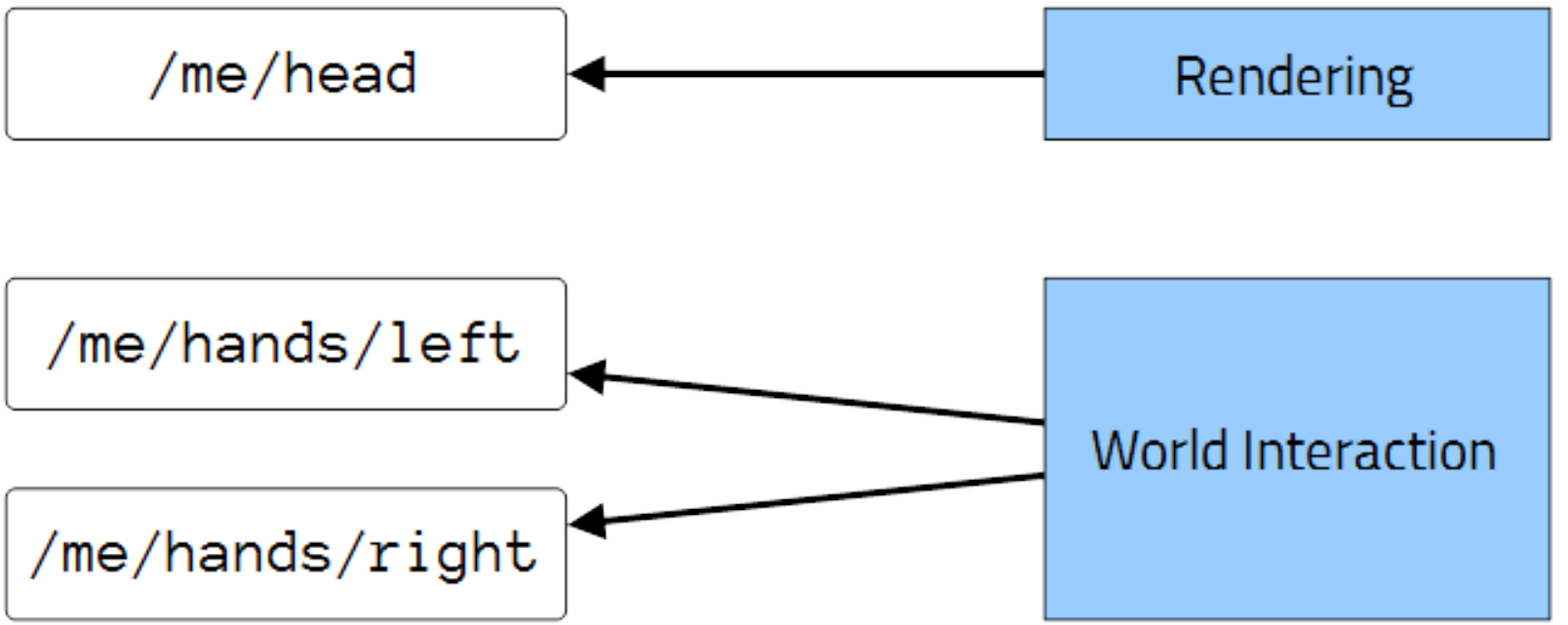
Close-Up of Plugin Side



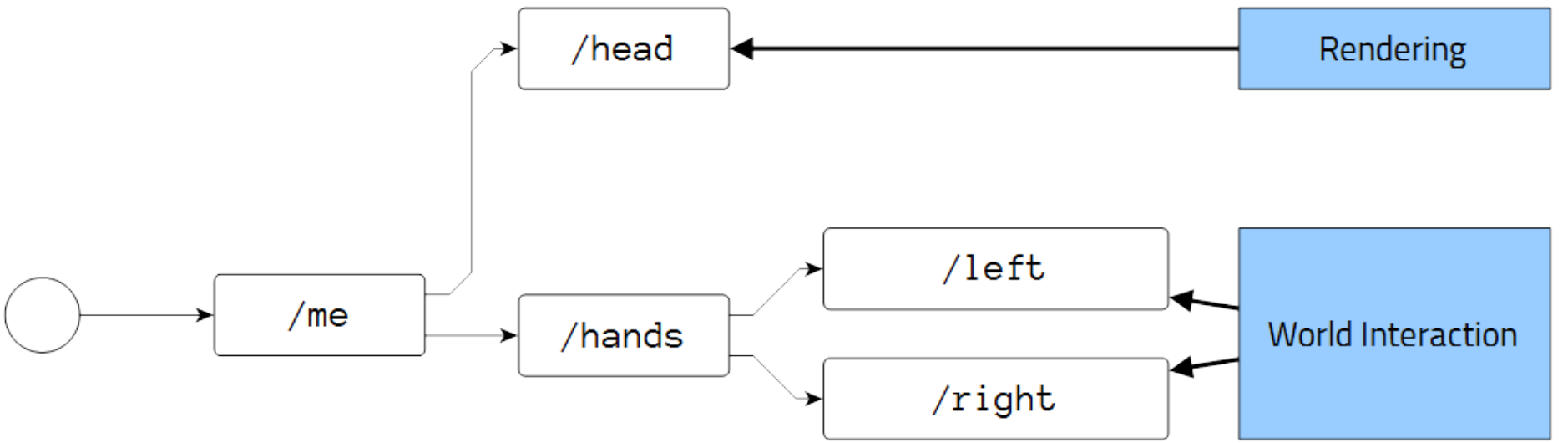
Close-up of App Side



An app asks for resources by “semantic name”



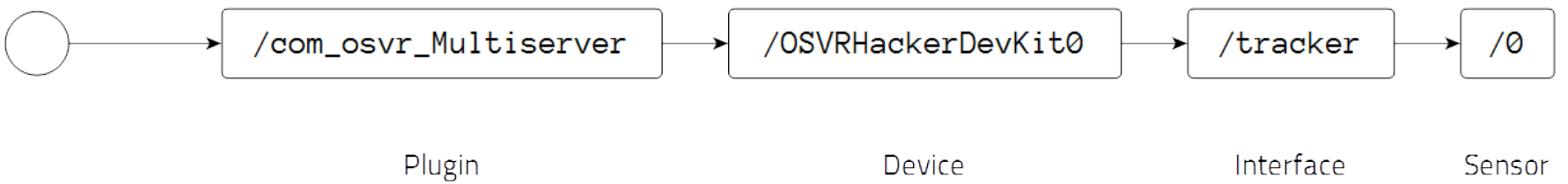
Actually, a “semantic path” (like a good URL)



So what?

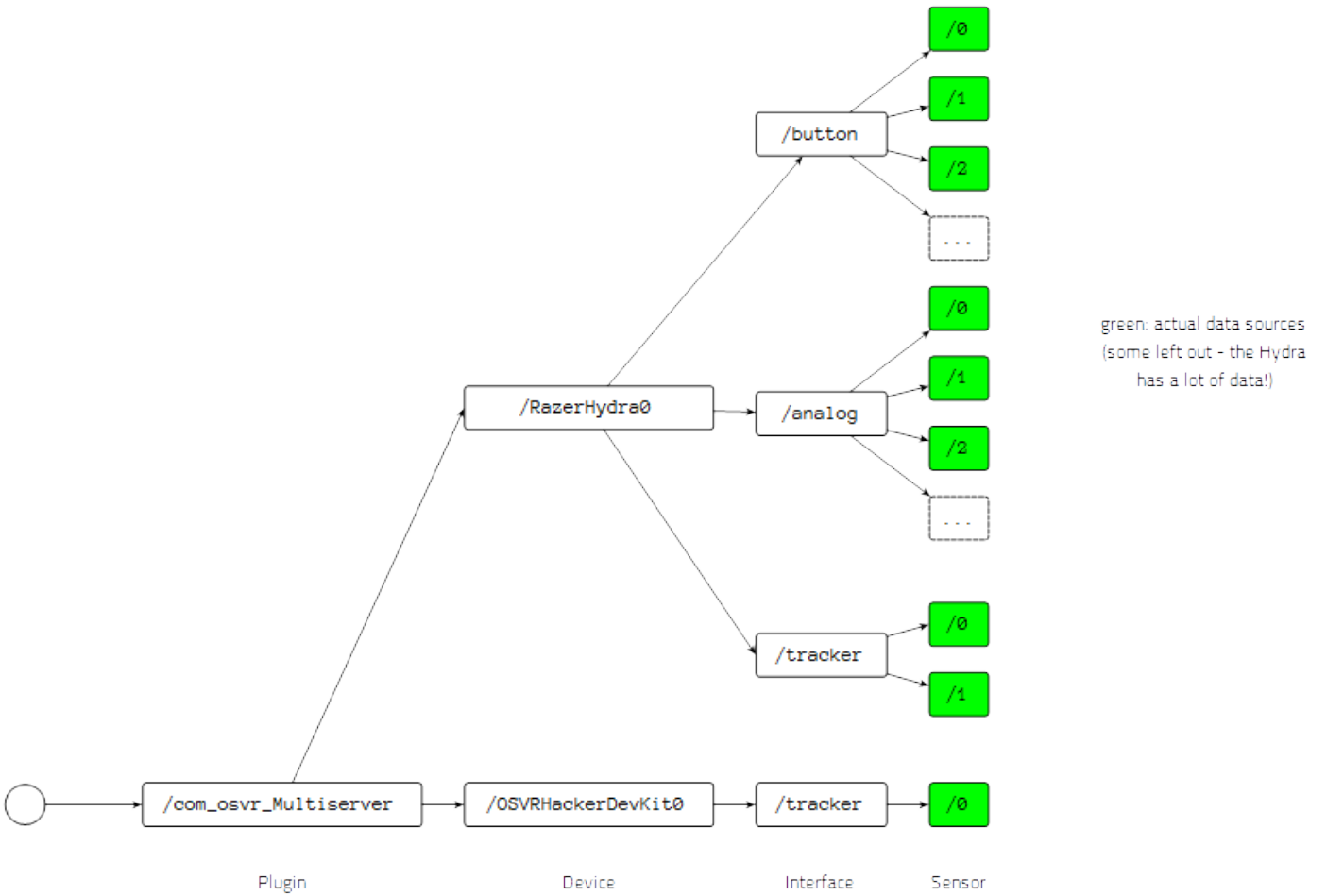
- This is a higher-level API: the game asks for what it wants by its *meaning*, not by its data source
  - Left hand position, not tracker data from Hydra sensor 0
- So we can make sure the game gets the data with the suitable meaning, with a variety of hardware
- To find the actual data source, we must look at another part of the path tree...

Device driver in plugin



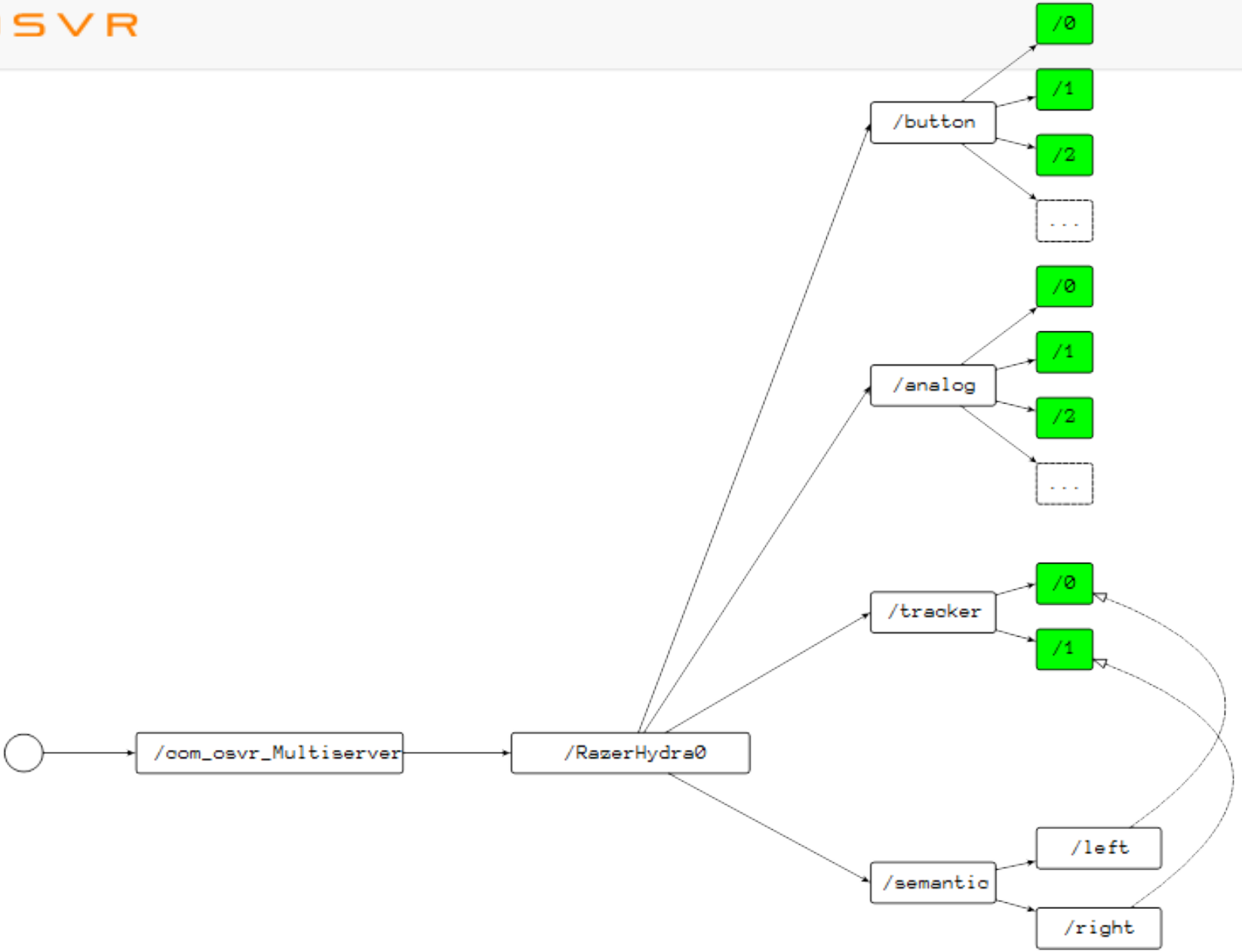
OSVR

Add another input device

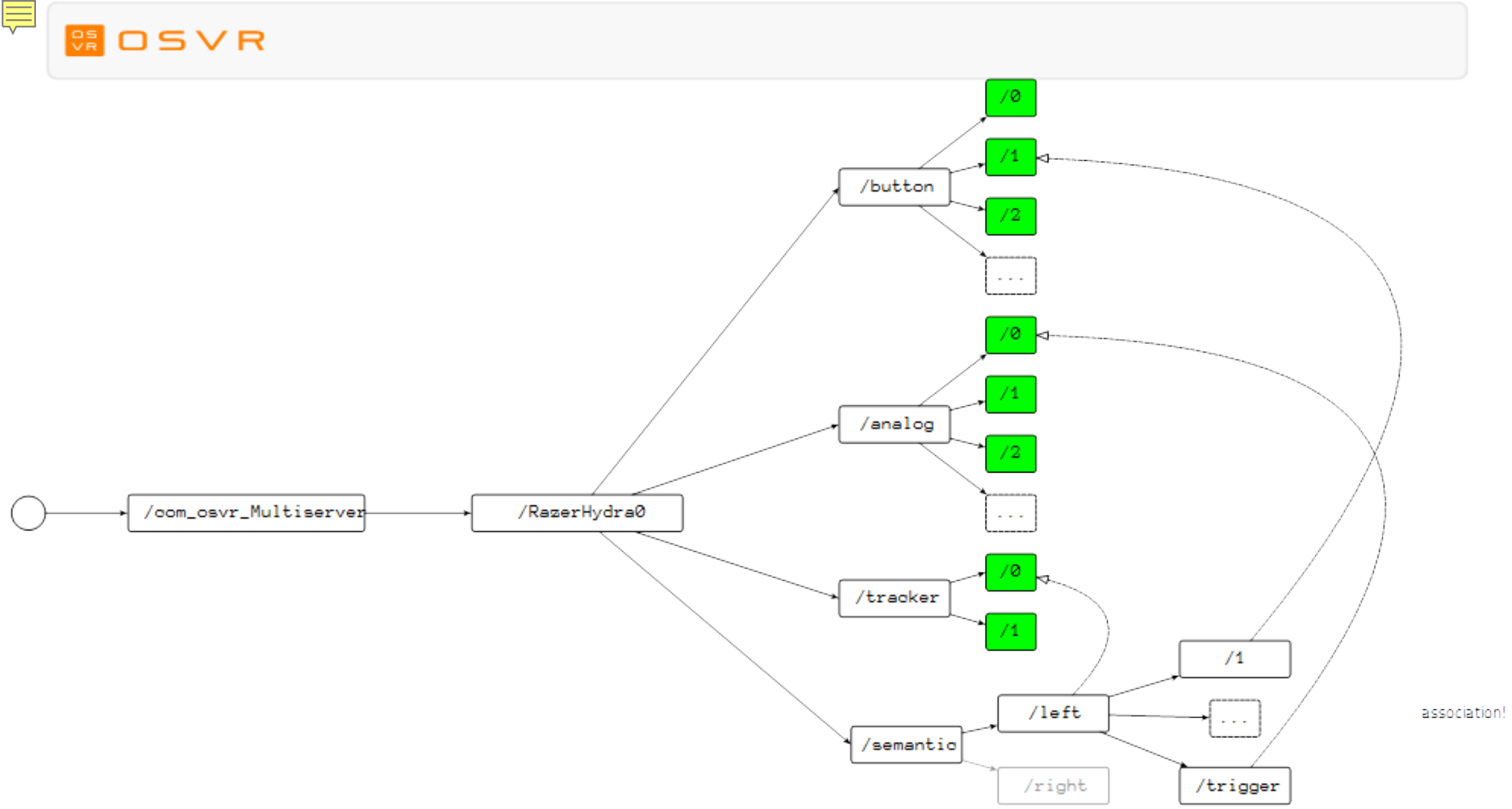


green: actual data sources (some left out - the Hydra has a lot of data!)

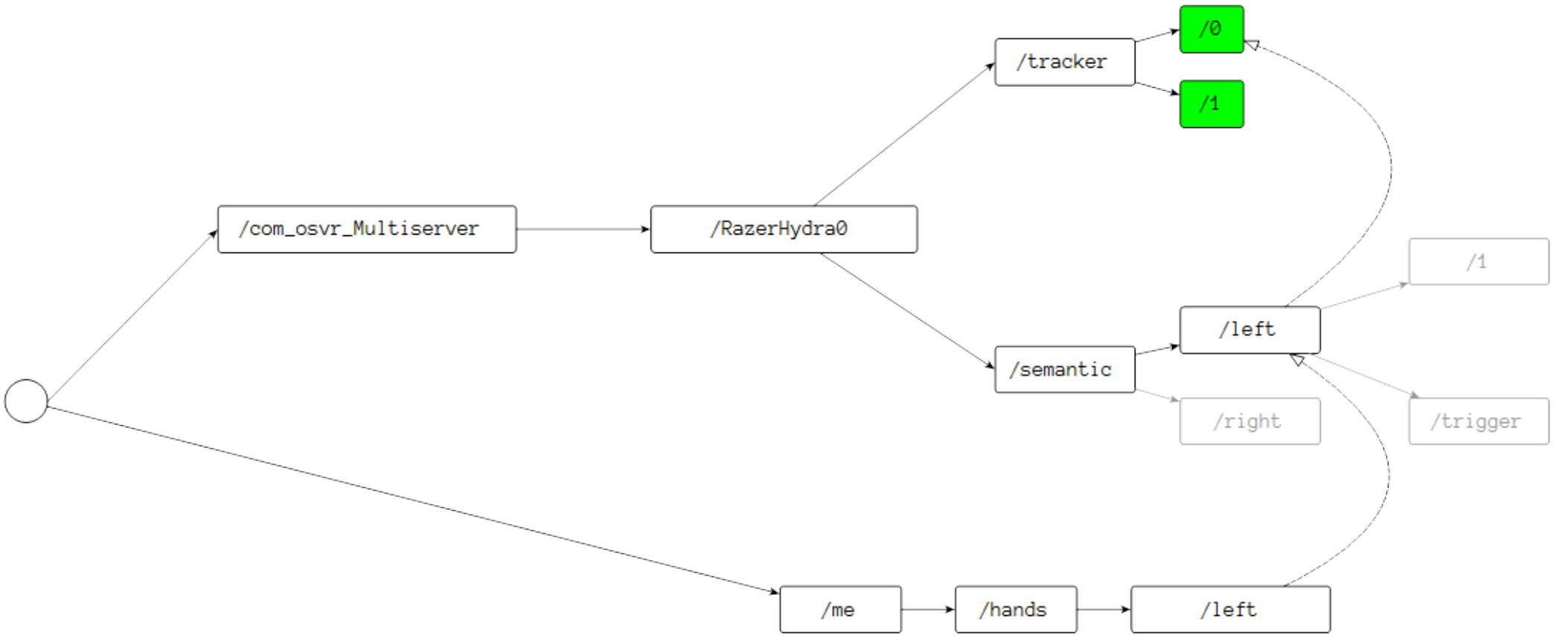
OSVR

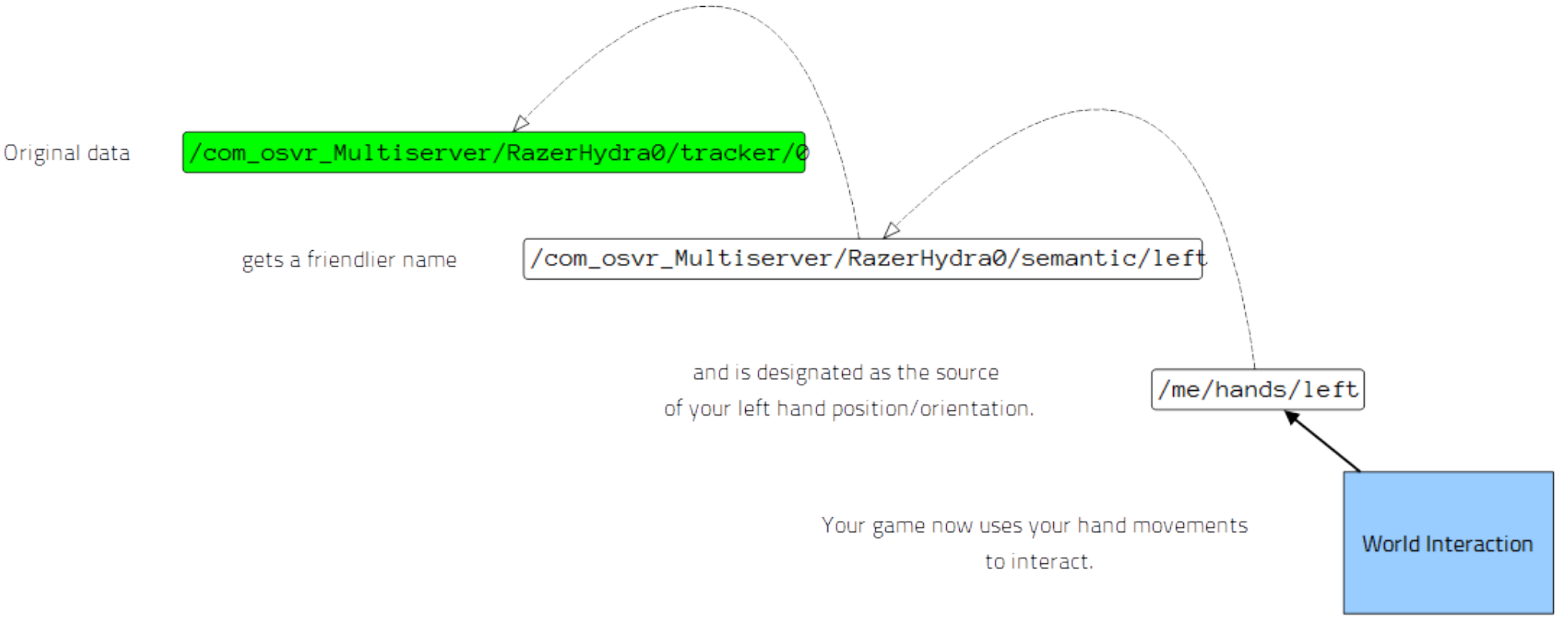


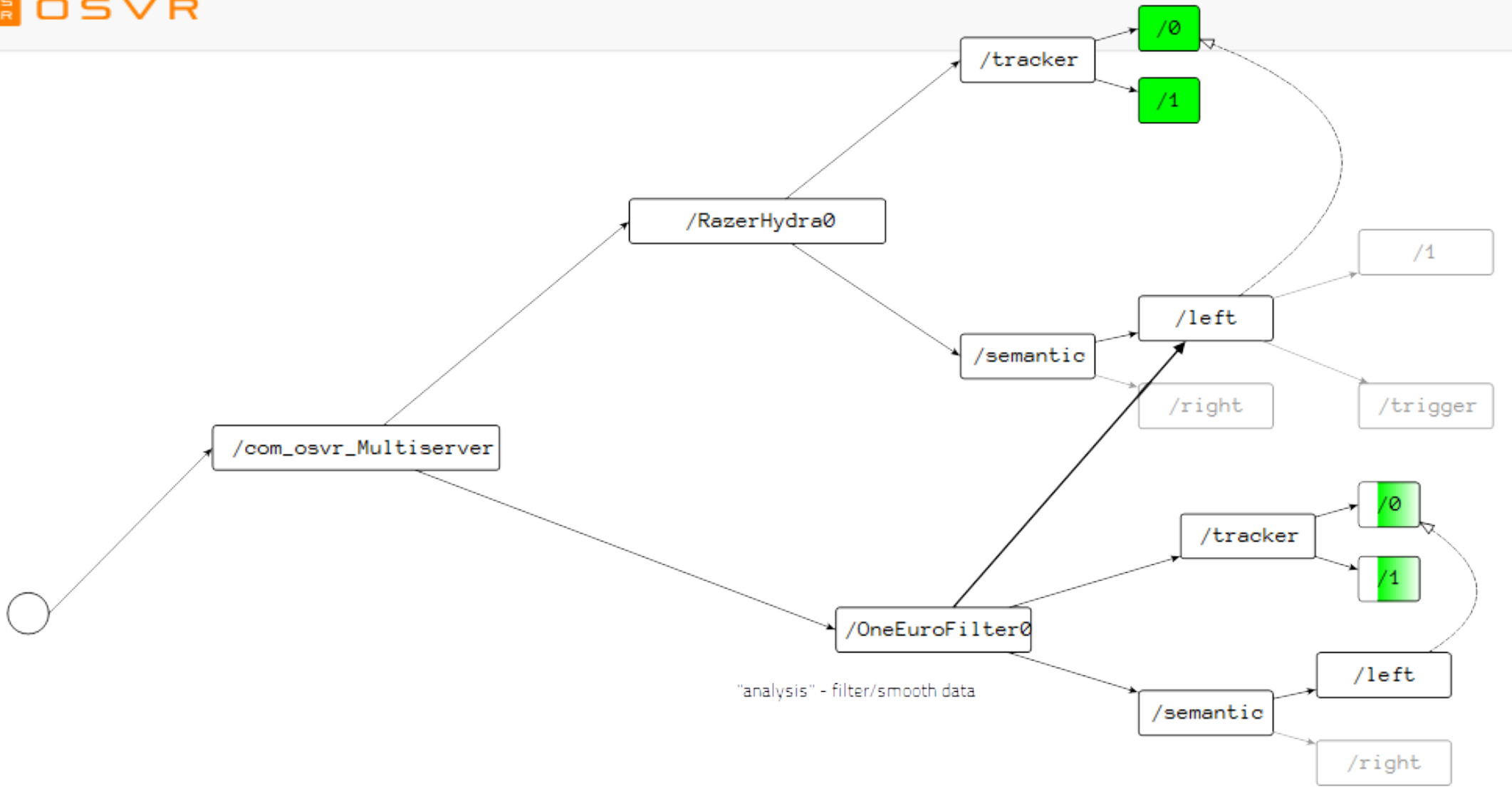
aliases!


OSVR
/com_osvr_Multiserver
/RazerHydra0
/button
/0
/1
/2
...
/analog
/0
/1
/2
...
/tracker
/0
/1
/semantic
/left
/1
...
/right
/trigger
association!

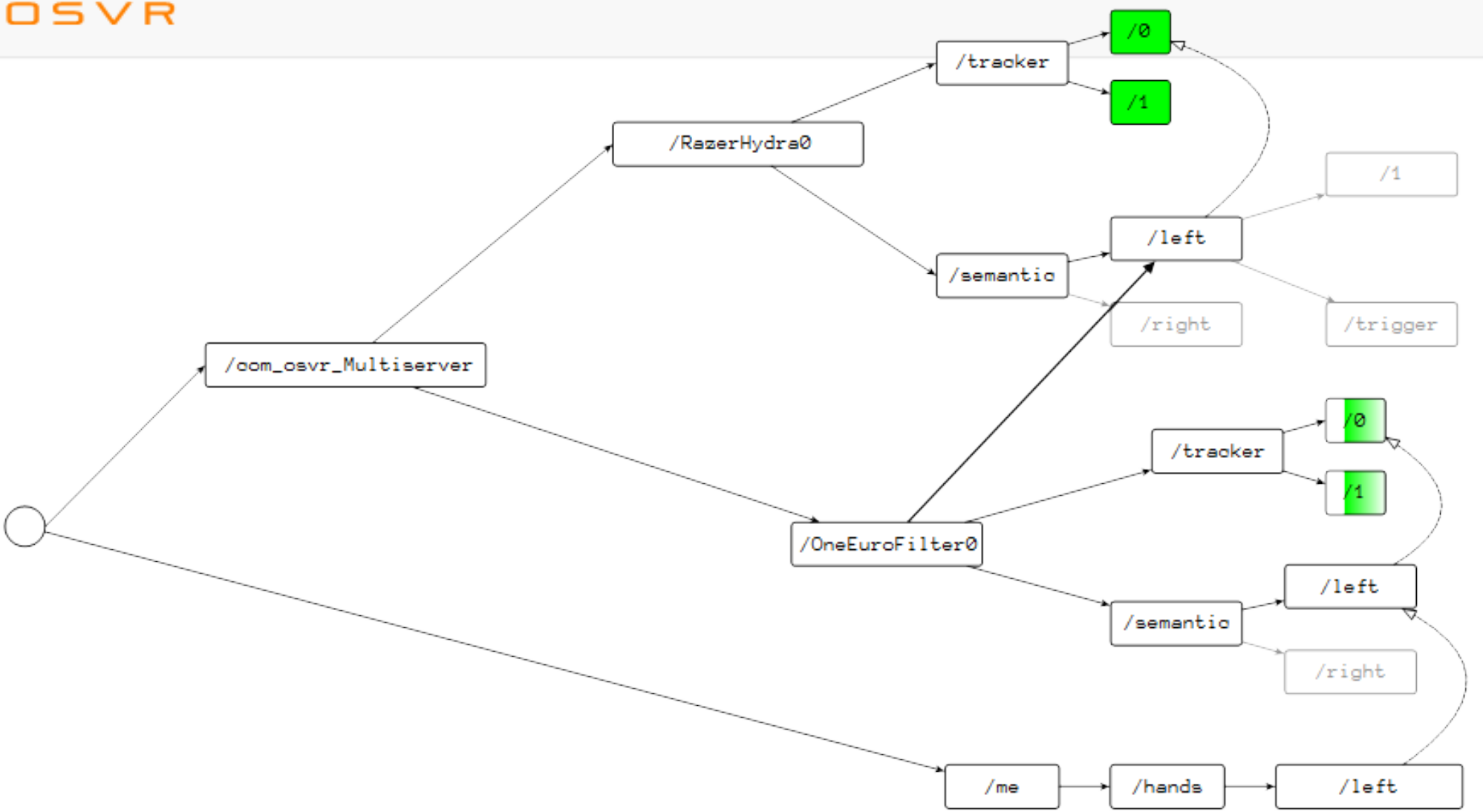
OSVR


/com_osvr_Multiserver
/RazerHydra0
/tracker
/0
/1
/semantic
/left
/1
/right
/trigger
/me
/hands
/left


Original data
/com_osvr_Multiserver/RazerHydra0/tracker/0
gets a friendlier name
/com_osvr_Multiserver/RazerHydra0/semantic/left
and is designated as the source
of your left hand position/orientation.
/me/hands/left
Your game now uses your hand movements
to interact.
World Interaction

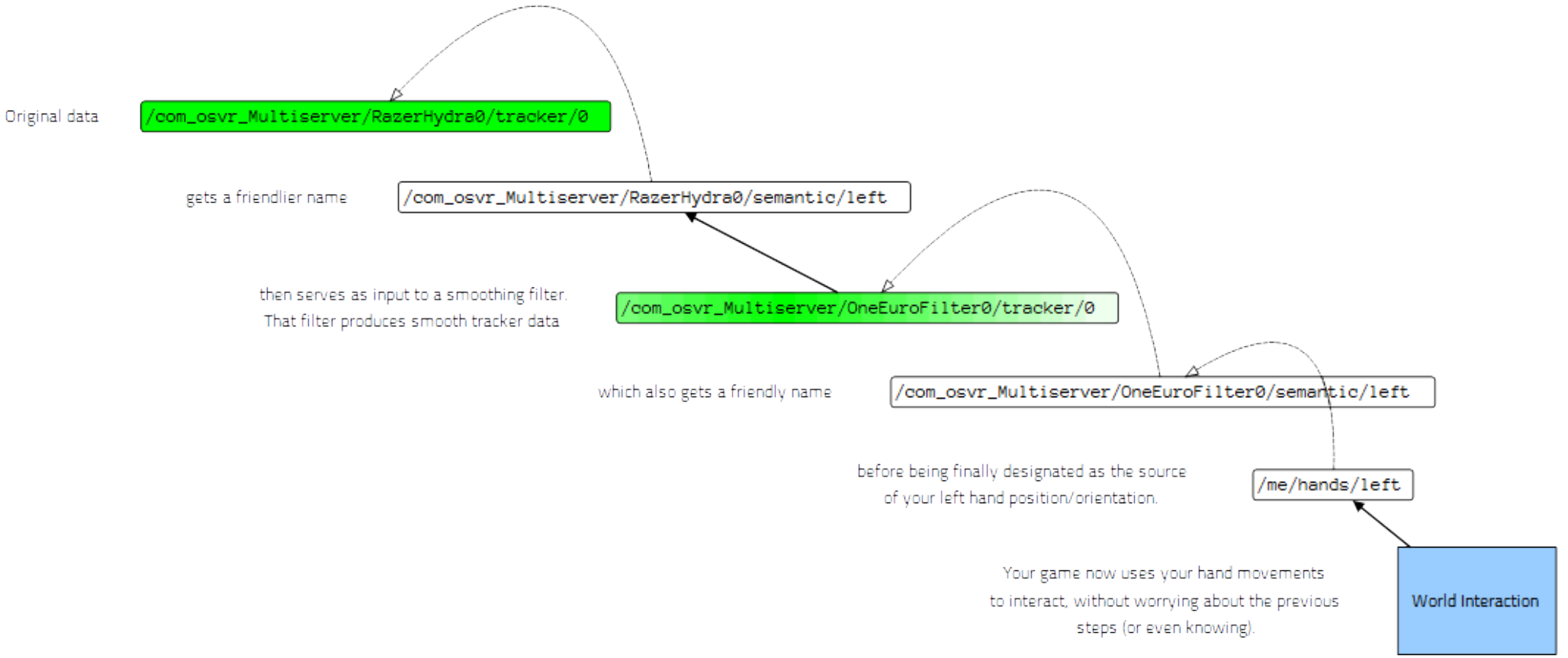

/com_osvr_Multiserver
/RazerHydra0
/tracker
/0
/1
/semantic
/left
/1
/right
/trigger
/OneEuroFilter0
/tracker
/0
/1
/semantic
/left
/right
"analysis" - filter/smooth data

OSVR



Ah, that's better tracking!

OSVR

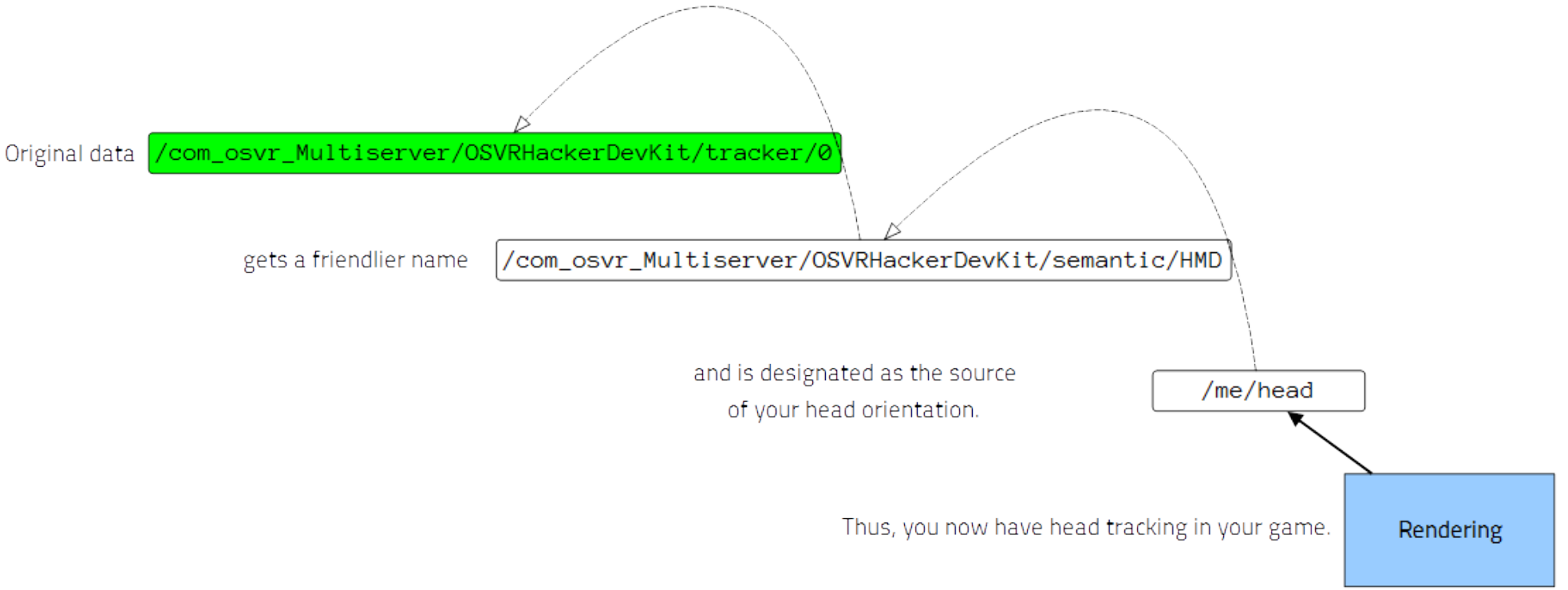

Original data
/com_osvr_Multiserver/RazerHydra0/tracker/0
gets a friendlier name
/com_osvr_Multiserver/RazerHydra0/semantic/left
then serves as input to a smoothing filter.
That filter produces smooth tracker data
/com_osvr_Multiserver/OneEuroFilter0/tracker/0
which also gets a friendly name
/com_osvr_Multiserver/OneEuroFilter0/semantic/left
before being finally designated as the source
of your left hand position/orientation.
/me/hands/left
Your game now uses your hand movements
to interact, without worrying about the previous
steps (or even knowing).
World Interaction

And thus...

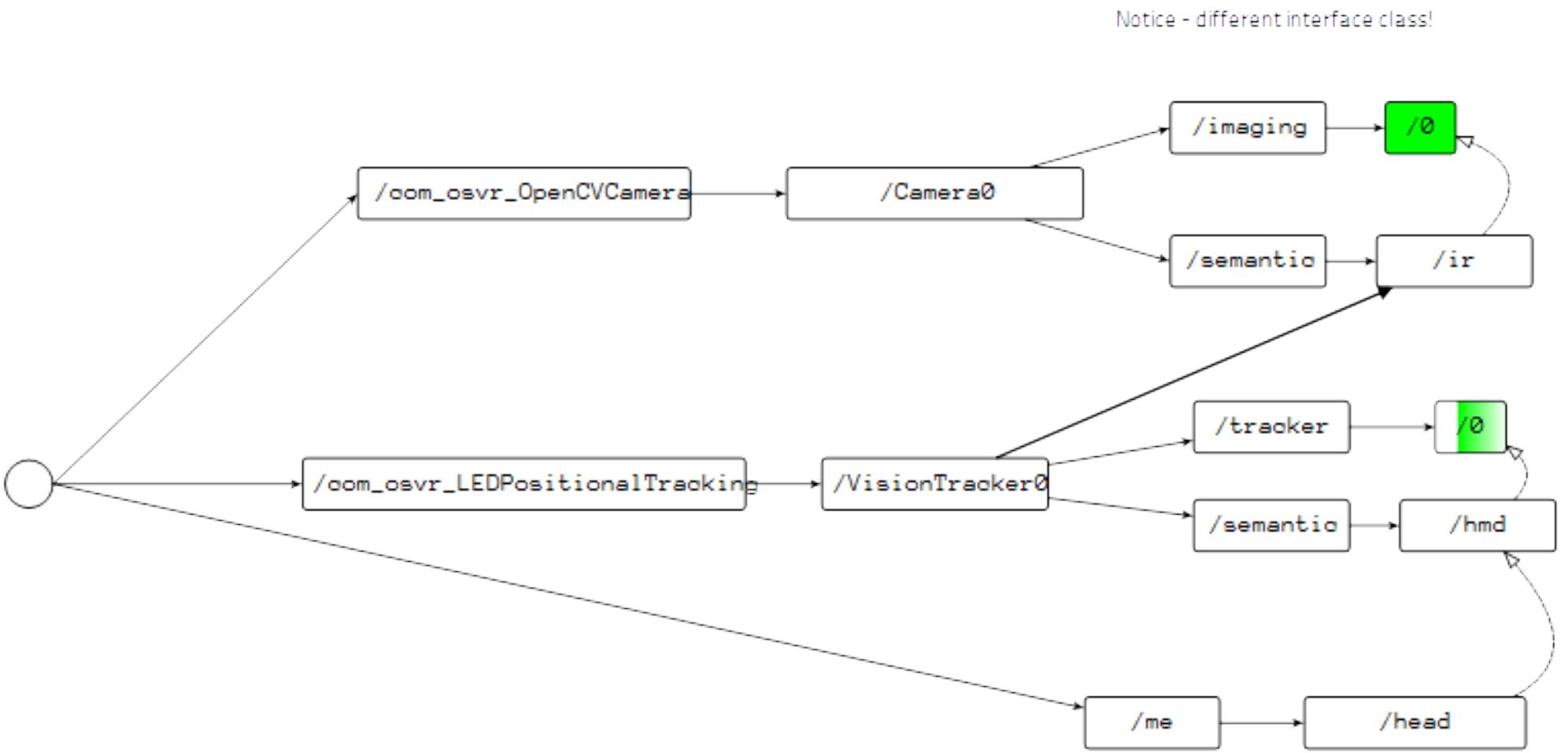
- You can in fact use anything with data resembling a tracker to provide /me/hands/left, now or in the future, even with a game from today.
- Generalizes to:
  - many different devices (including VRPN support!)
  - any conceivable analysis plugin(s)
  - whatever semantic names come into common usage

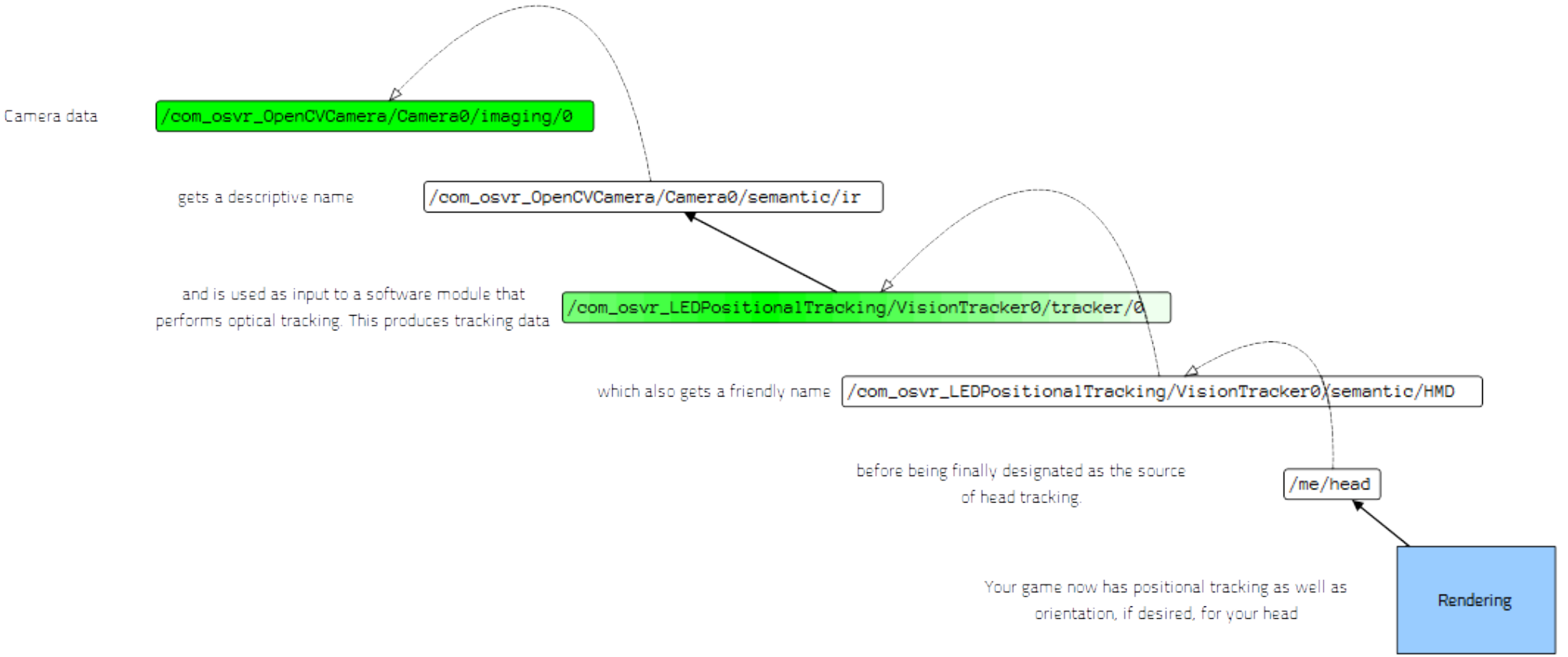
OSVR

Or, for instance,



OSVR


Notice - different interface class!
/oom_osvr_OpenCVCamera
/Camera0
/imaging
/0
/semantic
/ir
/oom_osvr_LEDPositionalTracking
/VisionTracker0
/tracker
/0
/semantic
/hmd
/me
/head


Camera data
/com_osvr_OpenCVCamera/Camera0/imaging/0
gets a descriptive name
/com_osvr_OpenCVCamera/Camera0/semantic/ir
and is used as input to a software module that performs optical tracking. This produces tracking data
/com_osvr_LEDPositionalTracking/VisionTracker0/tracker/0
which also gets a friendly name
/com_osvr_LEDPositionalTracking/VisionTracker0/semantic/HMD
before being finally designated as the source of head tracking.
/me/head
Your game now has positional tracking as well as orientation, if desired, for your head
Rendering

For additional information:

- OSVR developer portal
  - osvr.github.io
- Sensics – Founding contributor to OSVR, experts working in VR/AR for over a decade
  - www.sensics.com