

OEM specific HID architecture overview

HU

SDL HMI

- Handle HU specialized device. Properly interpret special devices' event to SDL RPC. If required, proprietary RPC shall be used. The RPC can be handled by proper plugin.

SDL core

- Unnecessary to add new RPC (SDL Core 4.4 can be used)

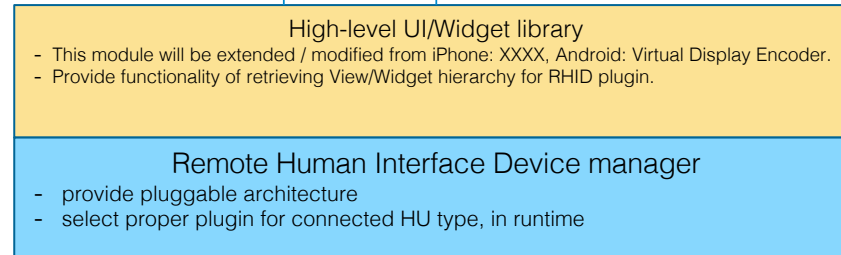
Specific Device

HS

SDL app

- Unnecessary to customize for each HU's special device
- Easy to migrate iOS/Android app to SDL VPM support
- Unnecessary to draw focus highlight
- Unnecessary to control focus movement (If app hope to control then, it may be controlled)
- If Apps using themselves UI/Widget (current Sygic app) then Apps must be customized with using each plugin directly.

SDL proxy



----- Remote Human Interface Device plugin I/F -----

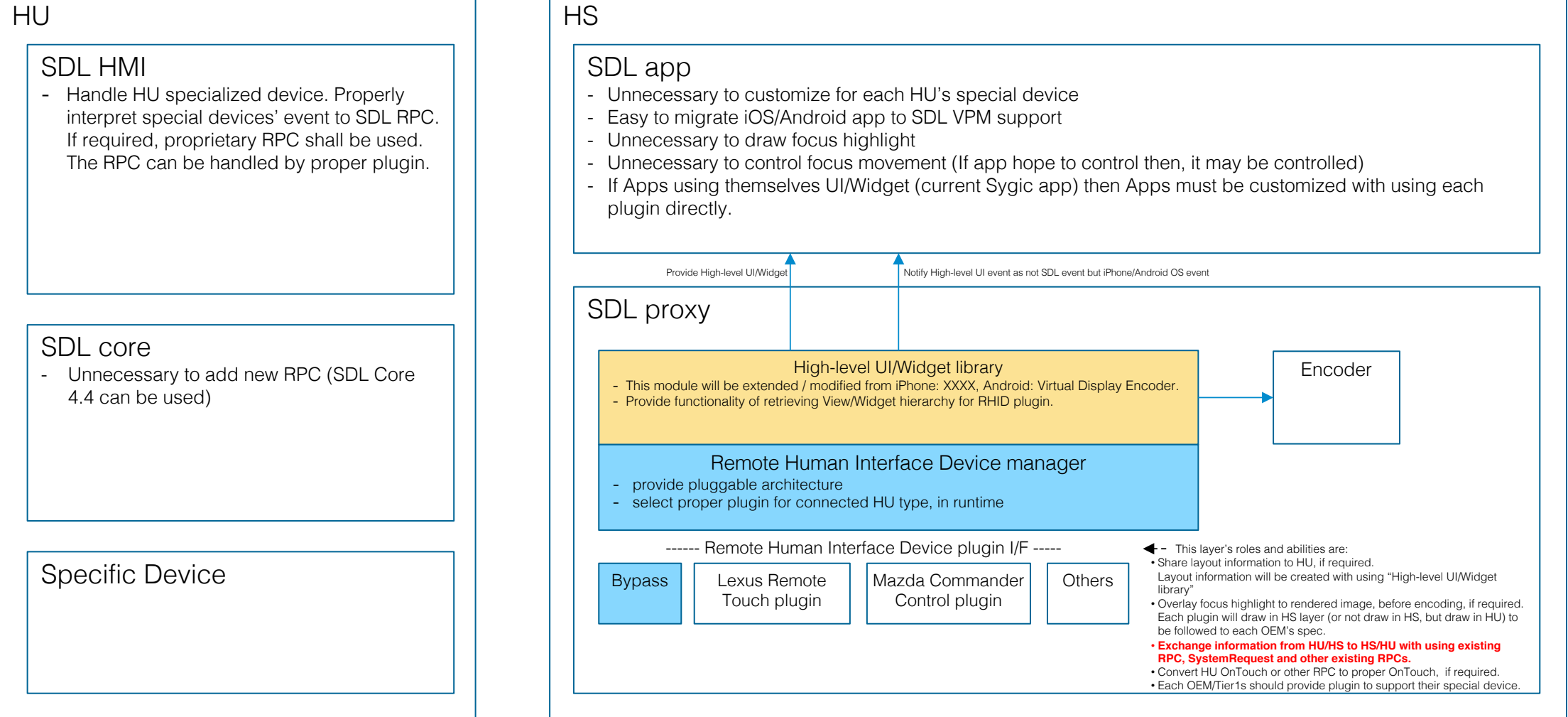


Encoder

- ← This layer's roles and abilities are:
- Share layout information to HU, if required. Layout information will be created with using "High-level UI/Widget library"
 - Overlay focus highlight to rendered image, before encoding, if required. Each plugin will draw in HS layer (or not draw in HS, but draw in HU) to be followed to each OEM's spec.
 - Exchange information from HU/HS to HS/HU with using new RPC for exchange some of proprietary information to HU (SendOEMSpecificData)
 - Convert HU OnTouch or other RPC to proper OnTouch, if required.
 - Each OEM/Tier1s should provide plugin to support their special device.

OEM specific HID architecture Alternative#1

Alternative #1: No new RPCs are added to support the plug-in interface. Rather, SystemRequest and other existing RPCs will be repurposed. Since the custom behaviors will be encapsulated in the plug-in and will only work in the repurposed way with the OEM head unit, there is no impact on other SDL implementations.



OEM specific HID architecture Alternative#2

Alternative #2: All of I/F which plug-in requires, will be implemented as MOBILE RPC. Plugin architecture will not be required, SDL HMI may be implemented OEM specific device control with using these RPC. However, this functionalities may be depend on Android/iOS architecture. So, it will be difficult to generalize because of another architecture.

