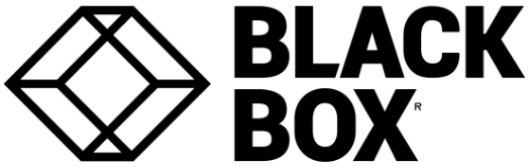

Emerald 4K 2K Interoperability Phase 1

Architectural Overview

Document Number: ENG-0008-025
Author: John O'Sullivan
Status: Draft
Rev: 0.1
Date: 24/03/2020

This document defines the Architectural Implications of the Interoperability Feaature on the Emerald Appliances

Master Document Revision History

Date	Version	Description	Author / Editor
March 23rd, 2020	0.1	First Draft	John O'Sullivan

1 Table of Contents

2 List of Figures

3 List of Tables

4 References

- Marketing Requirements
  https://bboxjira.atlassian.net/wiki/spaces/EM/pages/1230700564/Emerald+4K-+6.0
- ENG-0008-026 4K/2K Interoperability Analysis
- Emerald Detailed Product Specification DPS – ENG-0006-001

5 Terms and Definitions used in Blackbox

Term	Definition
CLU Image	A optimized encrypted file format used for 2K appliance software images.
Decoder	See Receiver
Encoder	See Transmitter
FIT Image	A optimized encrypted file format used for 4K appliance software images.
Municast	The use of multiple individual TCP streams to share a video from a single transmitter to multiple receivers
Multicast	The use of an individual Multicast UDP stream to share a video from a single transmitter to multiple receivers. Blackbox uses reliable multicast so that any dropped frames are resent and no data loss occurs.
Pixel Perfect Mode	The original scheme used by 4K products where the entire frame of the source video was captured continuously and sent with compression to the decoder.
RFX	RemoteFX Codec Extension developed by Microsoft which uses a lossy image codec to encode screen images with efficient and effective compression. This is the lossy scheme used in Blackbox products.
Receiver	A Blackbox appliance that receives and displays a video stream. A receiver is also referred to as a client or decoder.
RPU	Real Time Processing Unit. The Zynq Ultrascale+ MPSoC features two A5 RPUs. In our 4K products one of these is used for Display Port functionality using IP provided by Xilinx. The second on the encoder side acts a video processing feed between FPGA and TOE
Shared Mode	A generic term used in Blackbox to describe schemes that allow more than one receiver view the video source from a transmitter. See Municast and Multicast.
Tile Differencing Mode	A lossless scheme used to reduce bandwidth; only tiles (portions of the screen) that change are sent from encoder to decoder.

TOE	TCP Offload Engine: A hardware based TCP stack used to accelerate the transmission of data. This is essential in high bandwidth, low latency transmission of video and audio.
Technical Debt	Technical Debt is a concept in software development that reflects the implied cost of additional rework caused by choosing an easy (limited) solution now instead of using a better approach that would take longer.
Transmitter	A Blackbox appliance that captures and transmits a video from a source. A transmitter is also referred to as a server or encoder.
Video Source	A source of input for a Transmitter, normally the DVI or Display Port output of a PC.

Table 1 Terms and Definitions

6 Architectural Overview

There are two distinct technologies used in Emerald products. Our 2K products are based on Xilinx Zynq-7000 SoC series devices while the 4K product uses Zynq Ultrascale+ MPSoC devices. Prior to interoperability these technologies had the following characteristics.

	2K Product Family	4K Product Family
Technology	Zynq-7000 SoC	Zynq Ultrascale+ MPSoC
Network	1 Gig	10G and 1 Gig
Video Compression and Processing	Tile Differencing and RFX Compression	None- Pixel Perfect
Display Interfaces	DVI	Display Port

Table 2 Technology Differences

Customers need these technologies to be able to communicate with each other; this provides challenges because of the different video compression and processing techniques involved.
In order to support interoperability between the product families we propose to add the RFX video Compression scheme to the 4K products. The proposal in more detail is as follows:

- Add the RFX Compression decoder to the 4K Receiver Appliances.
- Add the RFX Compression encoder to the 4K Transmitter Appliances.
- Add a 4K capture differencing engine to the 4K Transmitter so that only changes in the video source are optimized and encoded.
- Remove the current Pixel Perfect mode from the 4K products and replace it with a scheme that uses Tile Differencing with the compression stage removed.

6.1 Product Impacts

This feature will impact existing 4K devices, all members of the 2K families, the Remote App product and Boxilla.

7 Use Cases

The following use cases over a representative selection of possible connection scenarios between a 2K and 4K device; they are not meant to be exhaustive.

7.1 4K Transmitter Perspective

Transmitter	First RX to Join	Second RX to Join	Behaviour
4K TX	2K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	4K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	The first 2K RX connection is accepted and the TX enters municast mode.[1] The second 4K RX connection is accepted in municast mode.
4K TX	2K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	4K RX with Shared Mode Enabled and Compression Mode set to Pixel Perfect.	The first 2K RX connection is accepted and the TX enters municast mode. The second 4K RX connection is rejected because it is not possible to mix modes.
4K TX	4K RX with Shared Mode Enabled and Compression Mode set to Pixel Perfect Mode.	4K RX with Shared Mode Enabled and Compression Mode set to Pixel Perfect.	The first 4K RX connection is accepted and the TX enters multicast mode. The second 4K RX connection is accepted in multicast mode.
4K TX	4K RX with Shared Mode Enabled and Compression Mode set to Pixel Perfect Mode.	4K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	The first 4K RX connection is accepted and the TX enters multicast mode. The second 4K RX connection is rejected because it is not possible to mix modes.
4K TX	2K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	2K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	The first 4K RX connection is accepted and the TX enters municast mode. The second 4K RX connection is accepted in municast mode.

[1] There is a general issue here with regard to the second connection to a shared instance; the second connection will join as a shared connection, but it will not know what mode to join in so it will have to join in whatever mode it is configured to use in the connection. This may the wrong mode for the given session. This suggests that we should have a preferred mode of operation for shared connections and that this preferred shared mode should be municast on 4K rather than Multicast.

7.2 2K Transmitter Perspective

Transmitter	First RX to Join	Second RX to Join	Behaviour
2K TX	2K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	4K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	The first 2K RX connection is accepted and the TX enters municast mode.[2] The second 4K RX connection is accepted in municast mode.
2K TX	2K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	4K RX with Shared Mode Enabled and Compression Mode set to Pixel Perfect.	The first 2K RX connection is accepted and the TX enters municast mode. The second 4K RX connection is rejected because it is not possible to mix modes.
2K TX	4K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	4K RX with Shared Mode Enabled and Compression Mode set to Pixel Perfect.	The first 4K RX connection is accepted and the TX enters municast mode. The second 4K RX connection is rejected because it is not possible to mix modes.
2K TX	4K RX with Shared Mode Enabled and Compression Mode set to Pixel Perfect Mode.	4K RX with Shared Mode Enabled and Compression Mode set to Pixel Perfect.	The first 4K RX connection is accepted and the TX enters multicast mode. The second 4K RX connection is accepted in multicast mode.
2K TX	4K RX with Shared Mode Enabled and Compression Mode set to Pixel Perfect Mode.	4K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth Mode.	The first 4K RX connection is rejected because 2K does not support Pixel Perfect Mode. The second 4K RX connection is accepted in municast mode.
2K TX	2K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth	2K RX with Shared Mode Enabled and Compression Mode set to Low Bandwidth	The first 4K RX connection is accepted and the TX enters municast mode. The second 4K RX connection is accepted in municast mode.

[2] At some point in the future we would envisage using: multicast for 2K to 2K sharing.

	Mode.	Mode.

Table 3 Use cases

8 Logical View

The figure below shows the logical view of the proposed architecture. On the left the decoder (or receiver) appliance is shown and on the right the encoder (or transmitter) application.
The areas shown in orange are areas most heavily impacted by the interoperability feature.

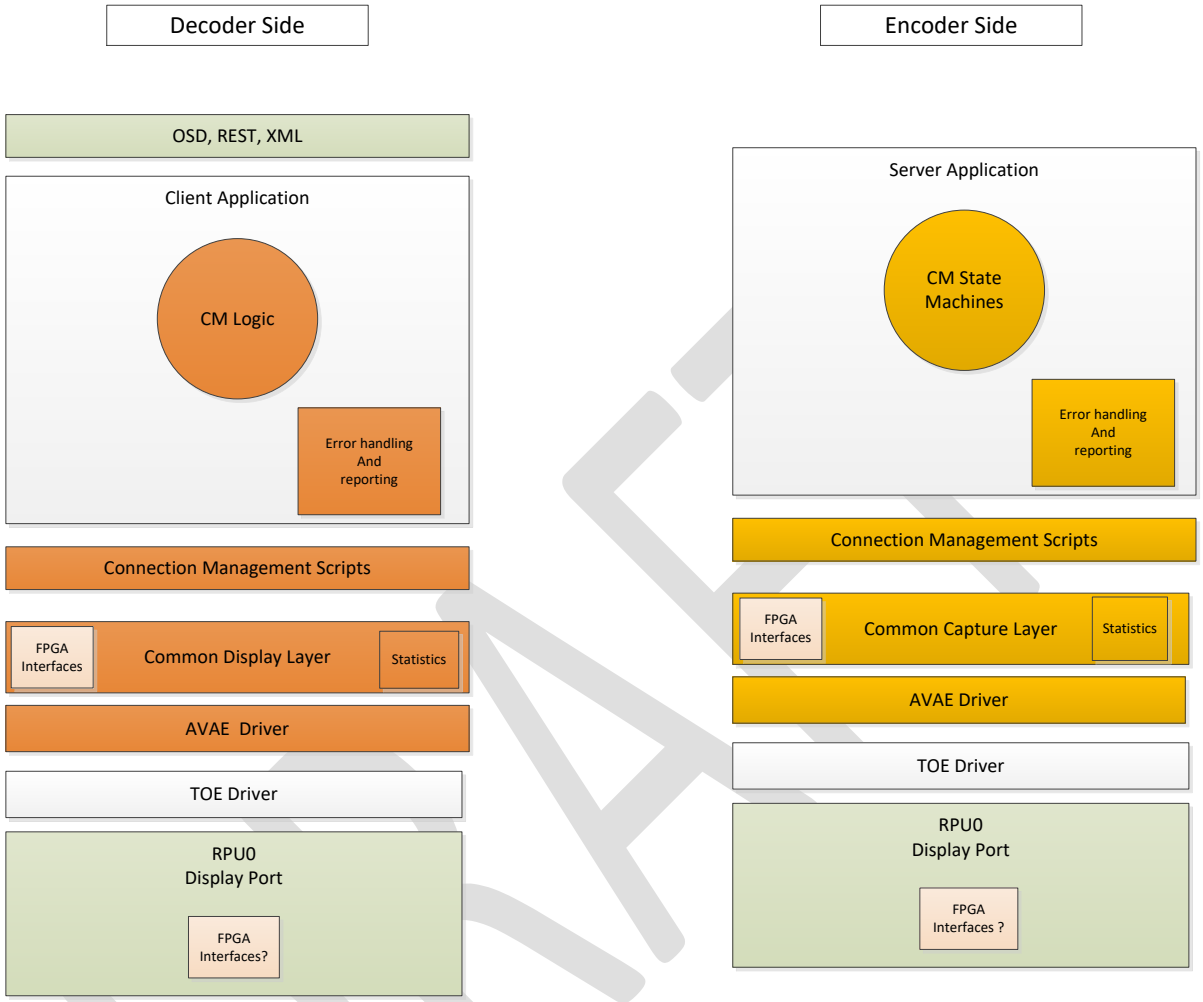

Figure 1 Logical View of proposed architecture

8.1 Main impacts to decoder side components

- The connection state machine on the decoder side will now have to handle three different connection types. The types, shown below, will be modelled as connection attributes. So a connection must be provisioned for a specific connection type.
  - Pixel Perfect Mode[3]
  - Lossless Mode – Tile Differencing Mode
  - Optimized – Compression using RFX
- Connection Modes: We already have different modes of operation; a system can operate in shared mode or private mode. These additional connection types further refine this concept of modes. So, for example, one could have the variations shown below:
  - A *private optimized* connection.
  - A *shared* optimized connection.

[3] Pixel Perfect Mode will be discontinued in the longer term

  - A shared lossless connection.
- The Error handling subsystem will have to change to accommodate the errors associated with the connection rejections highlighted in the use cases.
- In our appliances the applications and driver layers are abstracted by a series of connection scripts. These scripts will now have to accommodate the different connections types.
- Due to technical debt our display layer is currently fragmented into a 2K layer and a 4K layer. This has driven code duplication and presented numerous maintenance issues. We propose a common display layer to remove some of this debt and make the selection of operating modes more straightforward for interoperability.
- The AVAE Driver on the decoder will change to handle the Tile Differencing based solution.
- The OSD will need to incorporate a new connection attribute for the compression mode setting.
- The XML file will require a new compression mode field for each connection. There may be a requirement to filter the XML file before presenting the connection list on the OSD.
- Other changes to system libraries, device trees, memory maps will not be shown in detail in this document.

8.2 Main impacts to Encoder side components

- The connection state machine on the encoder side has a complex state machine. Due to technical debt this is distributed throughout the application and is as a result very difficult to maintain. In addition to the existing states management for shared mode and private mode the scheme will now have to handle the three different connection types.
  - Pixel Perfect Mode[4]
  - Lossless Mode – Tile Differencing Mode
  - Optimized – Compression using RFX
- The Error handling subsystem will have to change to accommodate the errors associated with the connection rejections highlighted in the use cases above.
- In our appliances the applications and driver layers are abstracted by a series of connection scripts. These scripts will now have to accommodate the different connections types.
- Due to technical debt our Capture layer is currently fragmented into a 2K Capture layer and a 4K Capture layer. This has presents numerous maintenance issues. Some of the FPGA interface functionality in 4K resides in the Display Port subsystem in RPU0. We propose a common display layer to remove some of this debt and make the selection of operating modes more straightforward for interoperability and, as part of this refactoring; we propose to encapsulate the FPGA functionality in a single location.
- The AVAE Driver changes on the encoder side are more extensive:
  - It must provide the setup for the Tile Differencing based solution.
  - It must allow configuration of video data processing so that the RPU1 based solution can be removed.
- Other changes to system libraries, device trees, memory maps will not be shown in detail in this document.

[4] Pixel Perfect Mode will be discontinued in the longer term

9 Process View

9.1 Encoder Side

As shown below, the main processes on the encoder side remain the same except for the highlighted areas shown below.

- The Connection Manager State Machine will evolve to handle the combination of operating modes and connection types. No new threads are anticipated and existing queues and sub systems should be sufficient.
- The capture layer will merge into a single common capture layer. Technology differences between 4K and 2K will be abstracted via a common interface so that the technology differences are hidden from the application layer. No new threads are anticipated.
- The Encoder Video Engine in the RPU (shown in red) will be made obsolete and the core functionality it provided will be replaced by the AVAE.
- The AVAE driver will use the current 2K model becoming a common driver across technologies and support will be added for Tile Differencing.
- The Connection Management scripts will need to add support for the new connection types and may need to split the existing Pixel Perfect logic into separate functions to allow selection between that mode and the new lossless tile differencing based mode.
- The Error handling subsystem on the encoder will be extended to accommodate the errors associated with the new use cases. These errors are shared between decoder and encoder. The encoder generates the error in the connection state machine and sends it back to the decoder.

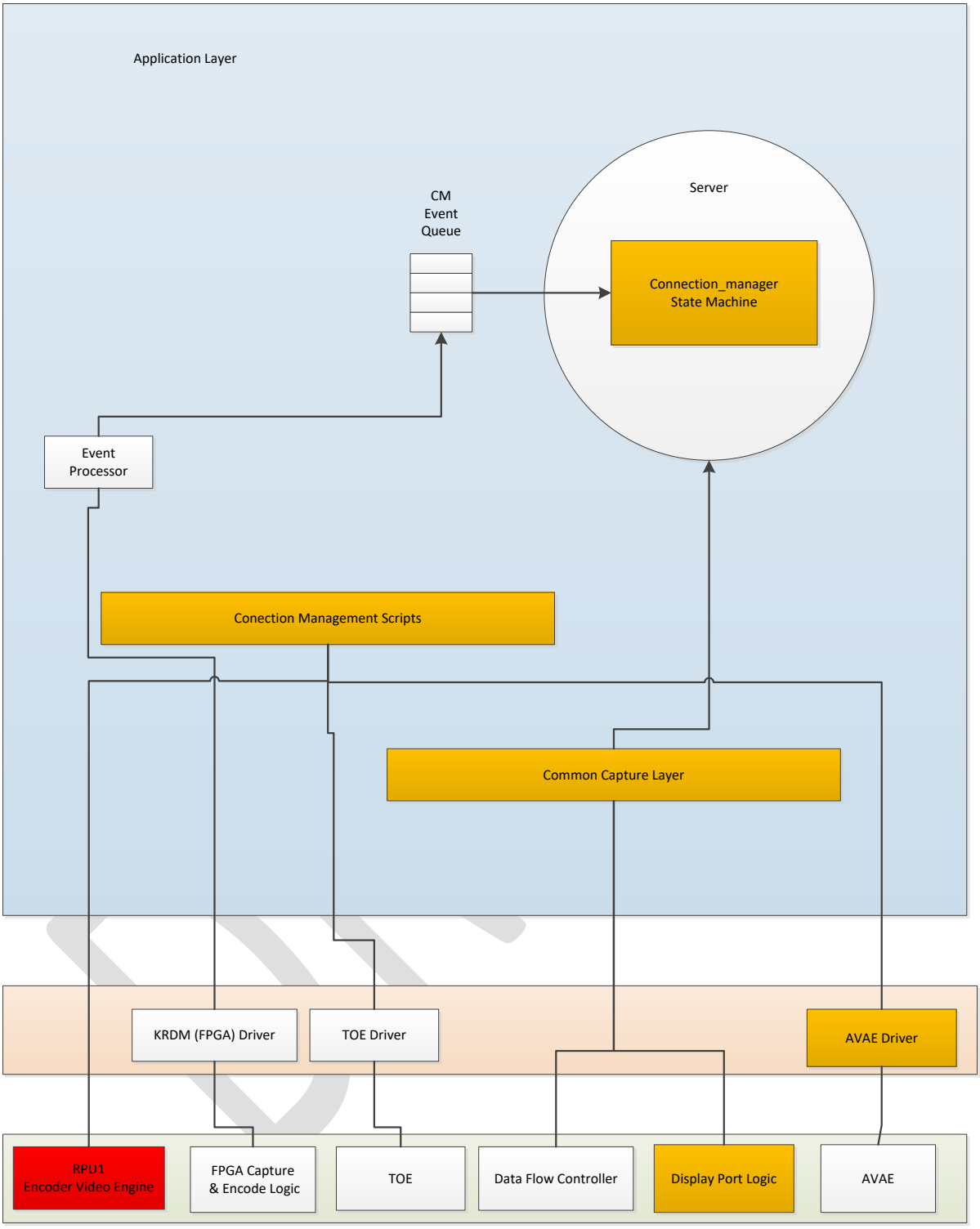

Figure 2 Encoder Process View

10 Decoder Side

As shown below, the main processes on the Decoder side remain the same except for the highlighted areas shown below in orange.

- The Connection Manager Logic in bbfreerdp will evolve to handle the combination of operating modes and connection types. No new threads are anticipated and existing queues and sub systems should be sufficient.
- The display layer will evolve into a single common display layer. Technology differences between 4K and 2K will be abstracted via a common interface so that the technology differences are hidden from the application layer. No new threads are anticipated but the design of this component needs to confirm this.
- The AVAE driver on the decoder side requires minimal changes. The support for Tile Differencing features will be added and is common to both sides; these new functions will be invoked via the proc file system.
- The Connection Management scripts will need to add support for the new connection types. There may be a need to split the existing Pixel Perfect logic into separate functions to allow selection between that mode and the new lossless tile differencing based mode.
- The Error handling subsystem on the decoder will be extended to accommodate the errors associated with the new use cases. The impacts here will be in the global bbdefines and the components in the OSD and connection manager that use them.

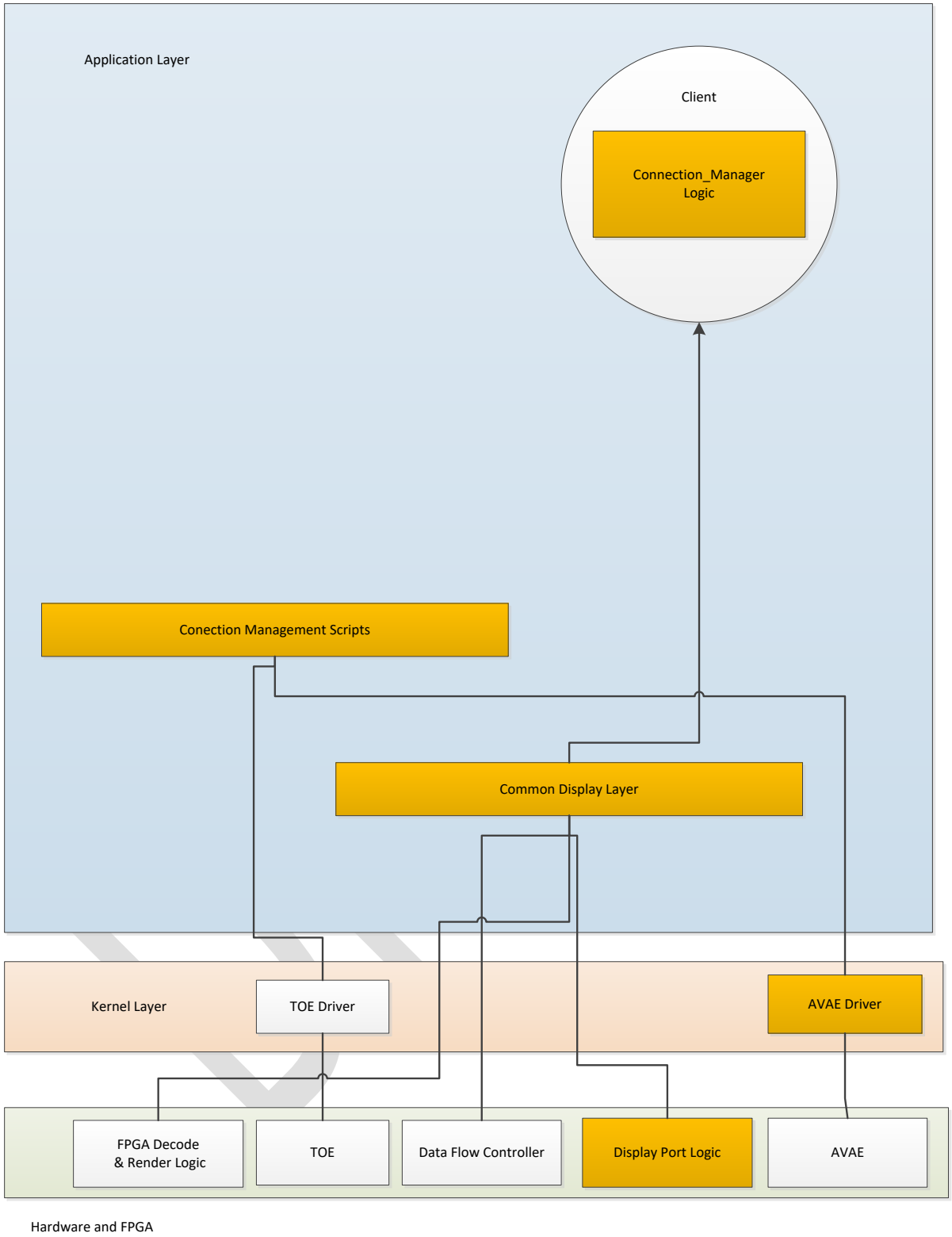

Figure 3 Decoder Process View

11 Deployment View

This view describes the environment within which the Interoperability Feature will operate. It describes the physical distribution of the system and the structure of the hardware components that

execute the software. This view also shows the system from the user's point of view and how the feature will be packaged into deployable units.
The most basic deployment is shown below with 2K transmitters and receivers interoperating in the same environment.

11.1 Physical Deployment

11.2 Unicast Mode

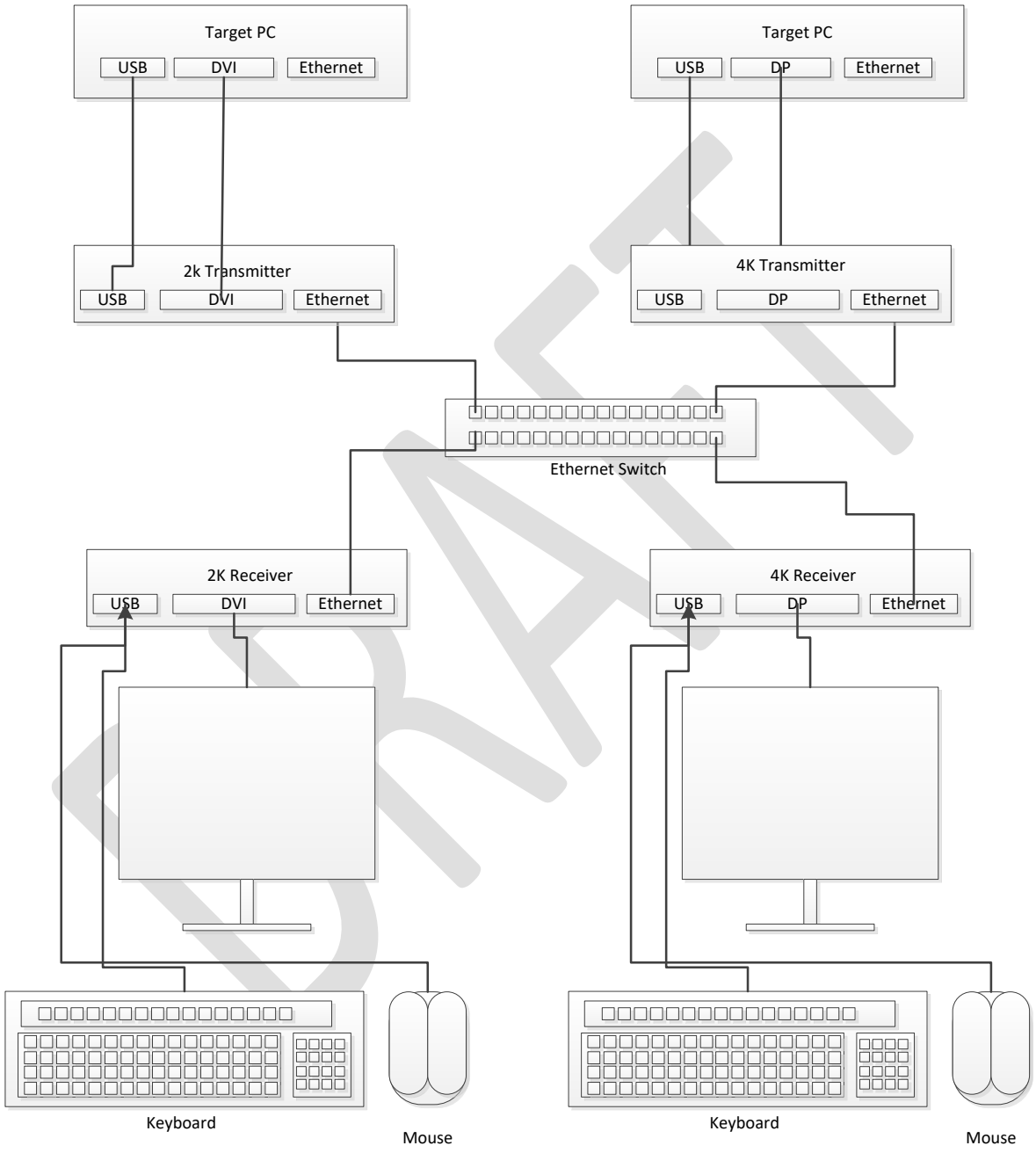

Figure 4 Basic Interoperability Deployment

11.3 Shared Mode

A shared mode deployment is shown below for an installation using 4K transmitters connected to a mix of 2K and 4K receivers with Boxilla Managing the system. 4K and 2K devices can interoperate once the appropriate connection type is selected.

Boxilla should only allow the provision of compatible connection types.

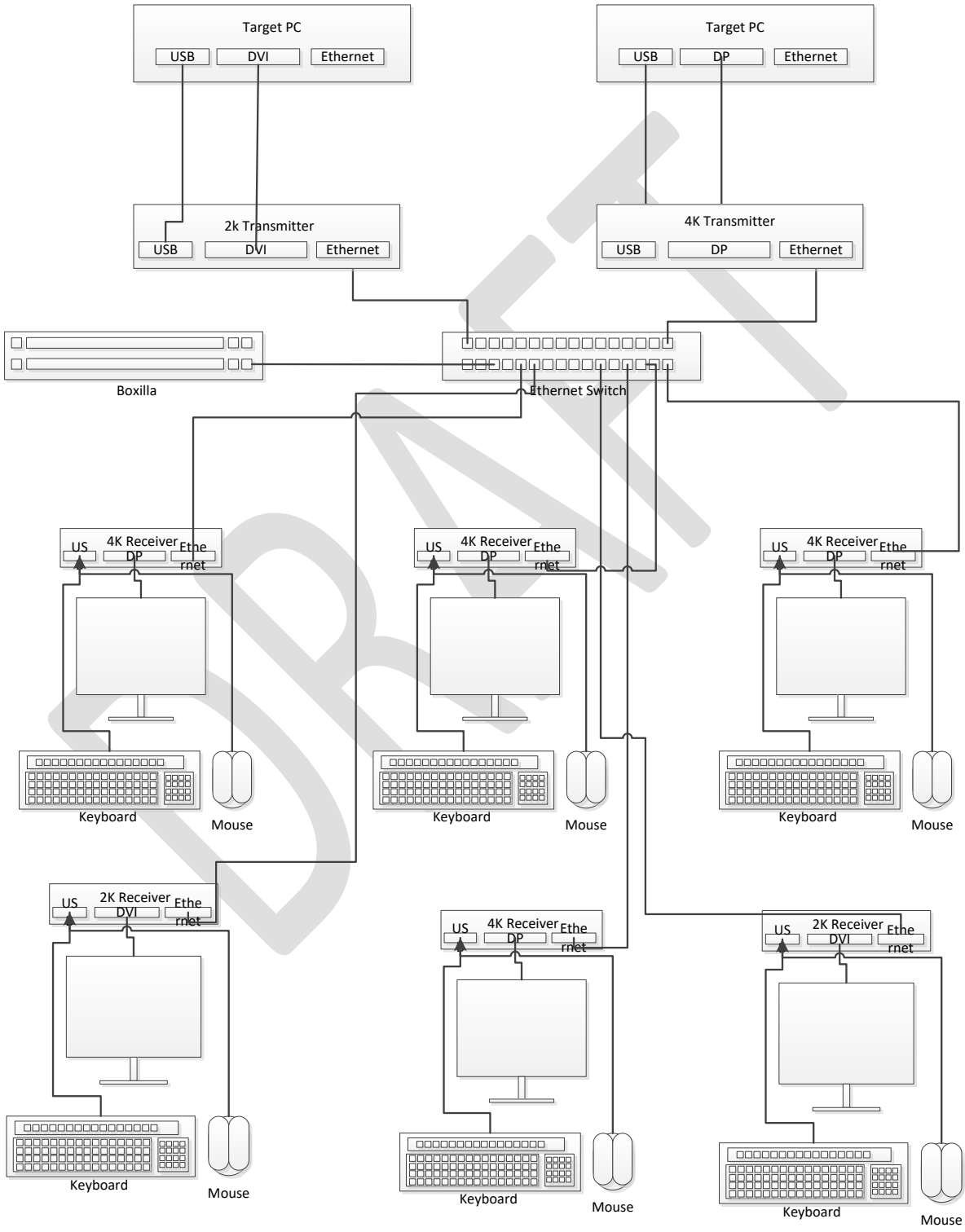

Figure 5 Shared Mode Deployment

11.4 Customer Deployment

11.4.1 Technical Support

The introduction of the interoperability feature will increase the complexity of product configuration and Technical Support personal will need to be made aware of the implications of the feature for customer environments.

11.5 Software Packaging

The interoperability feature requires changes to:

- 4K Receiver Firmware – .fit File
- 2K Transmitter Firmware – CLU File
- 2K Receiver Firmware – CLU File
- 4K Transmitter Firmware – .fit File
- Boxilla Firmware - .bbx (encrypted .deb files).
- Remote App Software - .msi

11.6 Software Upgrade

It is envisaged that the software upgrade format and capability will not change. There are no plans to create a common merged format for now, but it would be worth looking at combining the software for all devices so that Boxilla could work with a single image.

11.7 Phasing

The delivery of this feature may be phased over a number of releases. The phasing of the work is outside the scope of this document.