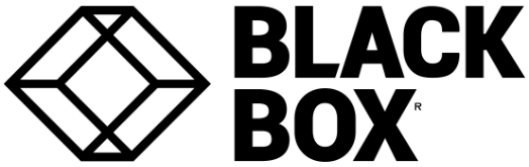

Emerald 4K 2K Interoperability

Architectural Overview

Document Number: ENG-0008-025
Author: John O'Sullivan
Status: Draft
Rev: 0.5
Date: 20/10/2020

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

Master Document Revision History

Date	Version	Description	Author / Editor
March 23rd, 2020	0.1	First Draft	John O'Sullivan
August 4th, 2020	0.2	Revised for Phase 2	John O'Sullivan
September 1st, 2020	0.3	Revised after review	John O'Sullivan
October 20th, 2020	0.4	Revised for final draft	John O'Sullivan
November 23rd, 2020	0.5	Updated after discussion with Ed	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 Assumptions

- A 2K Receiver can only make optimised connections.
- A 4K receiver can make lossless and optimised connections.
- Compatibility with older InvisaPC technology is outside the scope of this feature.
- A dual-head 2K receiver connecting to a 4K transmitter will display the video on the first head.
- Pixel Perfect Mode will not be supported in Phase 2. It will be dropped from the product.

7 Requirements

7.1 Connections

- A 2K RX can connect to a 4K TX operating in Optimised mode. If the 4K Transmitter is connected to a source greater than HD then the transmitter will scale the resolution down to HD when a optimised connection is made. (DC0035). [1]
- A 2K RX can connect to a shared 4K TX operating in Municast Optimised mode.
- Connections between Emerald4K Transmitters and Emerald2K Receivers shall be capable of supporting two channels of USB Redirection. (DC0013b)
- The following Connection modes between Emerald4K Transmitters and Emerald2K Receivers shall be supported as follows: (DC0018b)
    - Private
    - Shared (Multi-Unicast)
- The following Connection configurations between Emerald4K Transmitters and Emerald2K Receivers shall be supported as follows: (DC0019b)
    - Audio

[11] The system should scale to a resolution with same aspect ratio so that the picture is not distorted (in shape). There may be a need to review what happens in DCI mode (4096x2160) because it has a unique ratio. There are also proprietary resolutions that we support on DisplayPort.

- Persistent
- ViewOnly
- USBr

7.2 Configuration

4K and 2K devices receivers should be capable of configuring 4K and 2K transmitters. Certain constraints are detailed in the sections below, but in general:

- A 2K Receiver shall be capable of configuring a 4K transmitter with respect to settings and will also be able to retrieve diagnostic logs and restore the device.
- A 4K Transmitter shall be capable of configuring a 2K transmitter with respect to settings and will also be able to retrieve diagnostic logs and restore the device.

7.3 Discovery

Inter device discovery normally assumes a direct network connection between the two devices.

- A 2K Receiver shall be capable of discovering a 4K.
- A 4K Receiver shall be capable of discovering a 2K.

Incompatibility between 4K (10g SFP) and 2K (1G RJ-45) does not allow a direct connection currently[2]. We could allow the 4K 1G RJ-45 port to connect with 2K RJ-45 port, but this would require us bringing that port into service; currently, it is only used for debugging purposes.
We could also facilitate their connection using an isolated switch.[3]

8 Planning Considerations

In addition to the work following from the sections below, the following consideration should be factored into any planning.

- The reliability and maintainability of the product could be improved by introducing frame markers. There is no direct end-user requirement for this however, it is not considered part of the scope for phase 2 due to time constraints, but will follow in a future phase.

9 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

[2] Future versions of 4K will support 1G rates on SFP ports.
[3] Further discussion is needed on this.

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.[4]

9.1 Product Impacts

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

10 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.

10.1 4K Transmitter Perspective

Transmitter	First RX to Join	Second RX to Join	Behaviour
4K TX 2K Source	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.[5] The second 4K RX connection is accepted in municast mode.
4K TX 2K Source	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 Lossless.	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 or	4K RX with Shared Mode	4K RX with Shared Mode	The first 4K RX connection is accepted and the TX enters

[4] While the mode will not be available for the end user we will retain the logic in the code.

[5] 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 if the connection is configured for Optimised Mode.

2K Source	Enabled and Compression Mode set to Lossless.	Enabled and Compression Mode set to Lossless.	multicast mode. The second 4K RX connection is accepted in multicast mode.
4K TX **2K Source**	4K RX with Shared Mode Enabled and Compression Mode set to Lossless Mode.	4K RX with Shared Mode Enabled and Compression Mode set to Optimised.	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 Source**	2K RX with Shared Mode Enabled and Compression Mode set to Optimised.	2K RX with Shared Mode Enabled and Compression Mode set to Optimised.	The first 2K RX connection is accepted and the TX enters municast mode. The second 2K RX connection is accepted in municast mode.
4K TX **4K Source**	2K RX with Shared Mode Enabled and Compression Mode set to Optimised.	2K RX with Shared Mode Enabled and Compression Mode set to Optimised.	The first 2K RX connection is accepted and the TX enters municast mode and scales[6] the resolution to 2K. The second 2K RX connection is accepted in municast mode and it also receives the scaled down 2K connection.

10.2 2K Transmitter Perspective

Transmitter	First RX to Join	Second RX to Join	Behaviour
2K TX **2K Source**	2K RX with Shared Mode Enabled and Compression Mode set to Optimised.	4K RX with Shared Mode Enabled and Compression Mode set to Optimised.	The first 2K RX connection is accepted and the TX enters municast mode.[7] The second 4K RX connection is accepted in municast mode.
2K TX **2K Source**	2K RX with Shared Mode Enabled and Compression Mode set to Optimised.	4K RX with Shared Mode Enabled and Compression Mode set to Lossless.	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.

[6] There is an open question on how scaling would work. In the current architecture the Transmitter does not know what the preferred resolution of the Receiver is. We could by default scale to 2K 1920 X 1080 or, for more flexibility; we could send the default Receiver monitor resolution when we first make the connection.

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

2K TX **2K Source**	4K RX with Shared Mode Enabled and Compression Mode set to Optimised.	4K RX with Shared Mode Enabled and Compression Mode set to Lossless.	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 **2K Source**	4K RX with Shared Mode Enabled and Compression Mode set to Lossless.	4K RX with Shared Mode Enabled and Compression Mode set to Lossless.	The first 4K RX connection is rejected because 2K does not support Lossless Mode. The second 4K RX connection is rejected because 2K does not support Lossless Mode.
2K TX **2K Source**	4K RX with Shared Mode Enabled and Compression Mode set to Lossless.	4K RX with Shared Mode Enabled and Compression Mode set to Optimised.	The first 2K RX connection is rejected because 2K does not support Lossless Mode. The second 2K RX connection is accepted in municast mode.
2K TX **2K Source**	2K RX with Shared Mode Enabled and Compression Mode set to Optimised.	2K RX with Shared Mode Enabled and Compression Mode set to Optimised.	The first 2K RX connection is accepted and the TX enters municast mode. The second 2K RX connection is accepted in municast mode.

Table 3 Use cases

11 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.

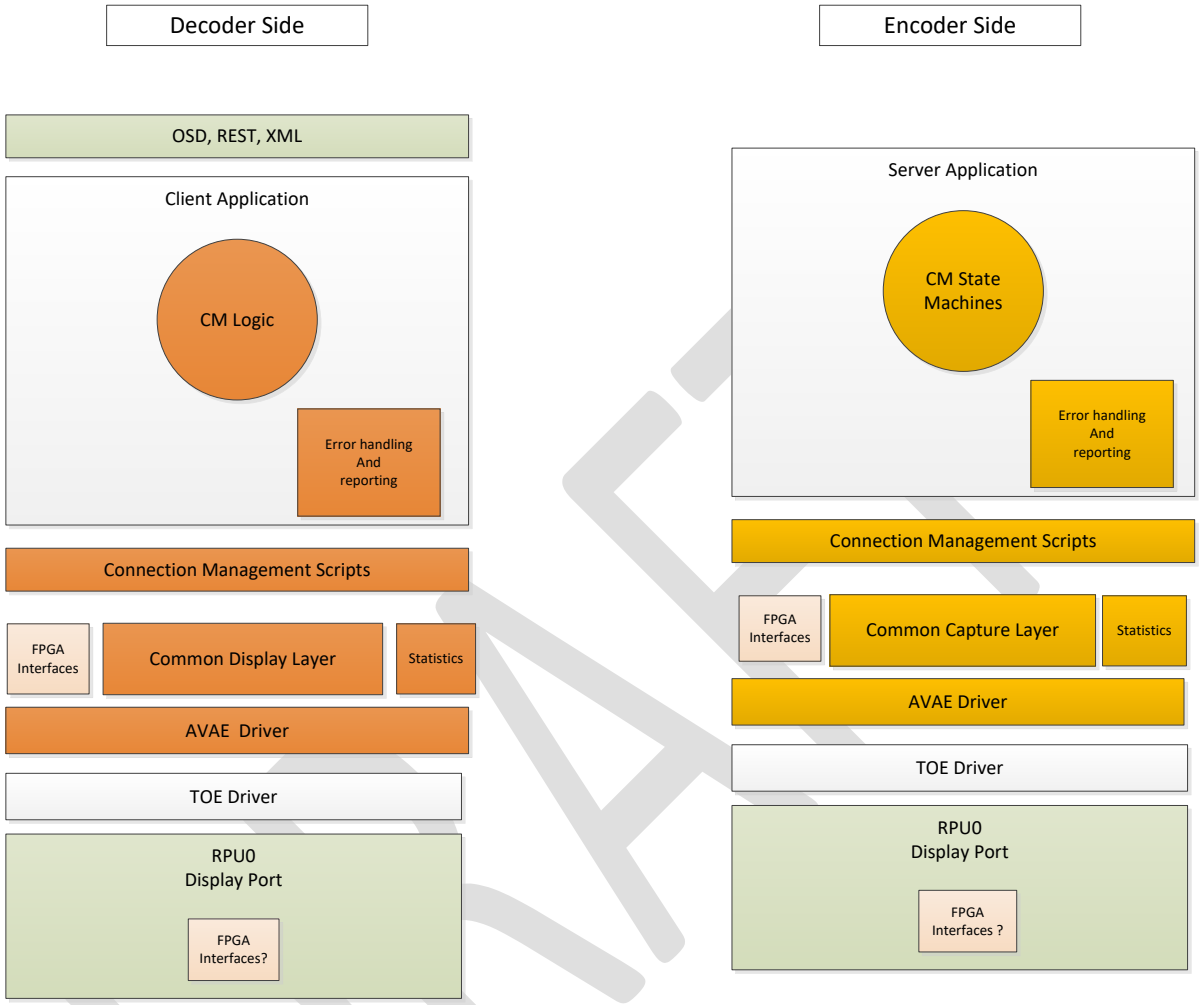

Figure 1 Logical View of proposed architecture

11.1 Main impacts to decoder side components

- The connection state machine on the decoder side was modified in phase 1 to handle two different connection types.
  - Lossless Mode – Tile Differencing Mode (or Pixel Perfect Mode[8])
  - Optimised – Compression using RFX
- The decoder can now handle three different connection mode variations; a system can operate in shared mode or private mode using lossless or shared connection types. No further changes are envisaged at this point. The connection variations are:
  - A *private optimised* connection.
  - A *shared* optimised connection.
  - A *shared lossless* connection.
  - A *Private lossless* connection.

[8] Pixel Perfect Mode will be discontinued in this phase and Tile differencing will become the default mode.

- The Error handling subsystem was modified in phase 1 of the project to accommodate errors associated with the connection rejections highlighted in the use cases above. There may be some refinements of this in phase 2, but no significant architectural changes are anticipated.
- In our appliances the applications and driver layers are abstracted by a series of connection scripts. These scripts were changed in phase 1 of the project to accommodate the different connections types and modes and now significant changes are anticipated for phase 2.
- Phase 1 of the project created a common display layer. No significant changes are anticipated for phase 2 but there may be some changes required to resolve ongoing issues with tile differencing mode.
- During phase 1 development mode the AVAE Driver on the decoder was refactored to handle the Tile Differencing lossless mode. This still needs to be fully evaluated during phase 2, but no significant changes are anticipated.
- The XML had a new compression mode field added during phase 1 development. There will be a requirement to filter the XML file on the 2K receiver before presenting the connection list on the OSD on Emerald 2K devices.

11.2 Main impacts to Encoder side components

- The connection state machine on the encoder side was modified extensively in phase 1 to handle the three different connection types. No further impacts are anticipated at this point in the core state machine which now handles the following connection types.
    - Lossless Mode – Tile Differencing Mode (or Pixel Perfect Mode[9])
    - Optimised – Compression using RFX
- The Error handling subsystem was updated in phase 1 of the project. This may have to be updated to accommodate any new errors associated with the connection rejections from a 2K receiver connecting to a 4K transmitter, but no significant refactoring is envisaged.
- In our appliances the applications and driver layers are abstracted by a series of connection scripts[10]. These scripts will now have to accommodate the tile differencing mode of operation.
- 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 and some logic is also embedded directly into server-emerald. Depending on how we want to handle it the scaler on 4K will also have significant architectural impact. 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 were made in phase 1:
    - These added provision for the setup for the Tile Differencing mode.
    - These also added a configuration function for video data processing
    - These changes allow the removal of the current RPU1 based solution used by pixel perfect mode.
- Incorporating the frame marker feature[11] will require changes in the handling of municast and multicast in the AVAE driver.

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

[10] At the time of writing this is changing to an architecture that replaces the scripts with C code. The C code uses system calls.

[11] The use of the frame marker does not have a direct user impact so its inclusion in the scope of this phase is optional. Additional analysis is required to understand the full architectural impact of this.

- Support 4K to 4K TCP SYN/FIN was added in phase 1, but the ability to update MSS dynamically must be completed if we retain the RPU for any reason[12]; this will also require changes in the AVAE for tile differencing operation.

[12] If we moving away from RPU to AVAE for video, then a fixed MSS limitation should not be an issue.

12 Process View

12.1 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 was refactored in phase 1. 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 had support for Tile Differencing added in phase 1 of this feature.
- The Connection Management scripts[13] will need to add support for the tile differencing based connection but much of this was added in phase one and remaining work may be minimal.
- The Error handling subsystem on the encoder will be extended to accommodate new errors associated with the new use cases involving 4K receiver connections to a 2K transmitter.

[13] At the time of writing this is changing to an architecture that replaces the scripts with C code. The C code uses system calls.

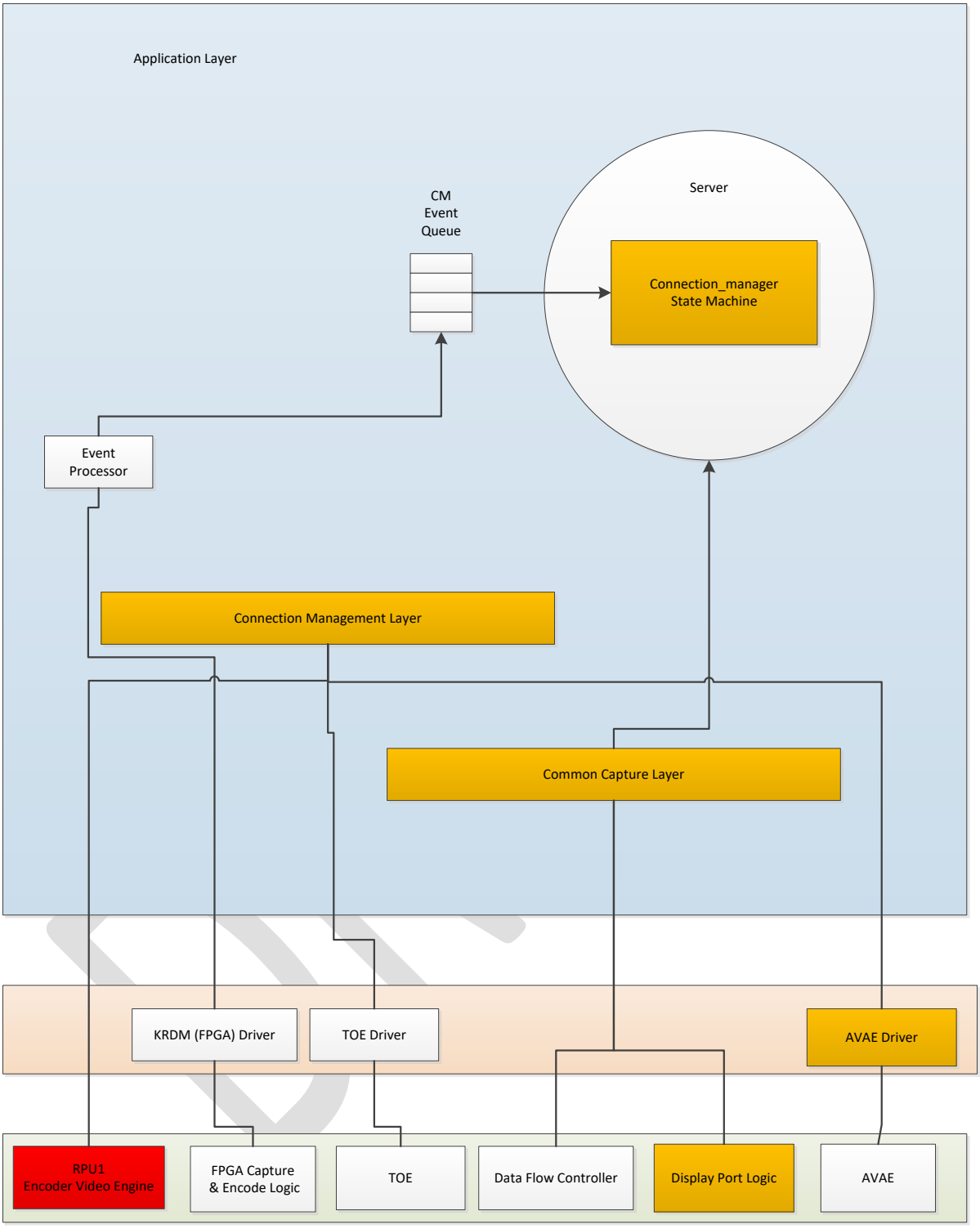

Figure 2 Encoder Process View

12.2 Decoder Side

Decoder side changes for connection logic and configuration were completed in Phase 1 of interoperability. No further changes are anticipated in this area, but there are changes required to Transmitter Settings to allow the 4K and 2K receivers interoperate with the transmitters so that settings, upgrades, diagnostics and system restore functionality can be retrieved.

12.2.1 Transmitter Settings

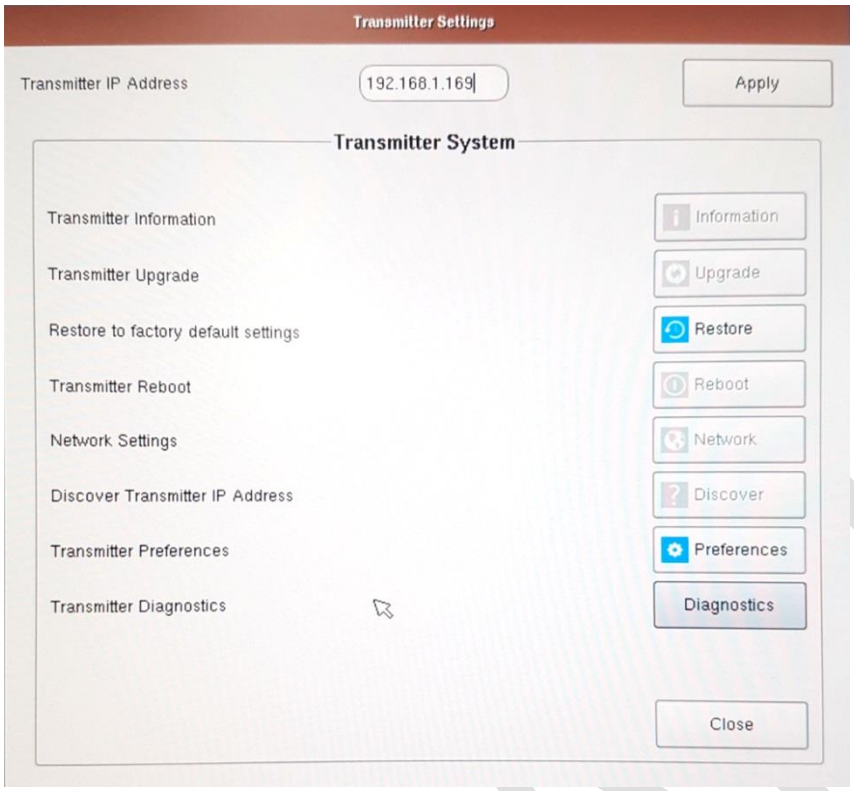

Figure 3 Transmitter Settings

12.2.1.1 System Restore

Invoking system restore should cause the targeted transmitter to restore to its default state.

12.2.1.2 Diagnostics

Invoking diagnostics should cause the targeted transmitter to send its default diagnostics to a connected 4K or 2K receiver.

12.2.1.3 Discovery

A receiver should be capable of discovering a connected 4K or 2K transmitter.

12.2.1.4 Transmitter Preferences

Clicking on Transmitter Preferences should cause the receiver to display the appropriate preferences page for the target transmitter.

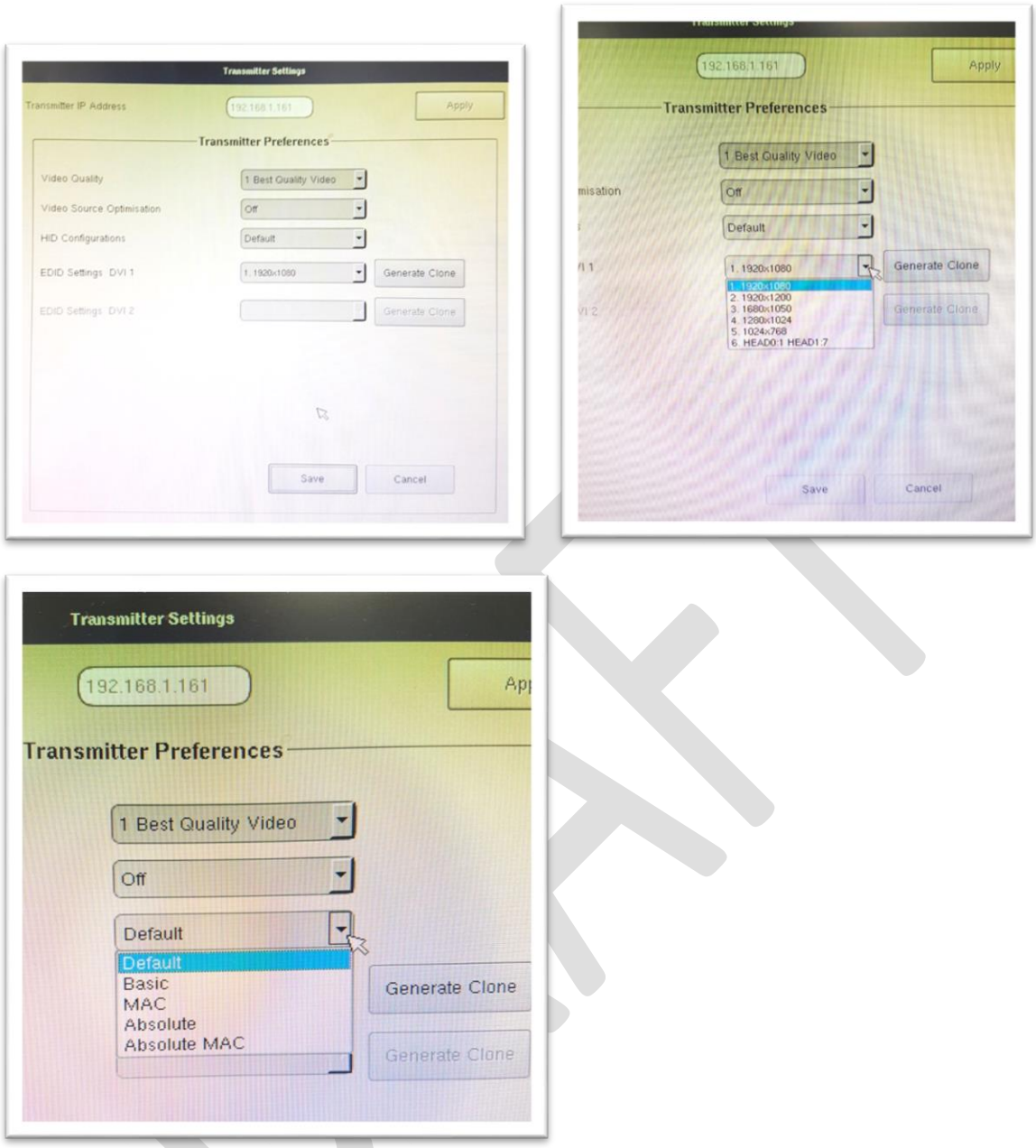

Figure 4 Transmitter Preferences 2K

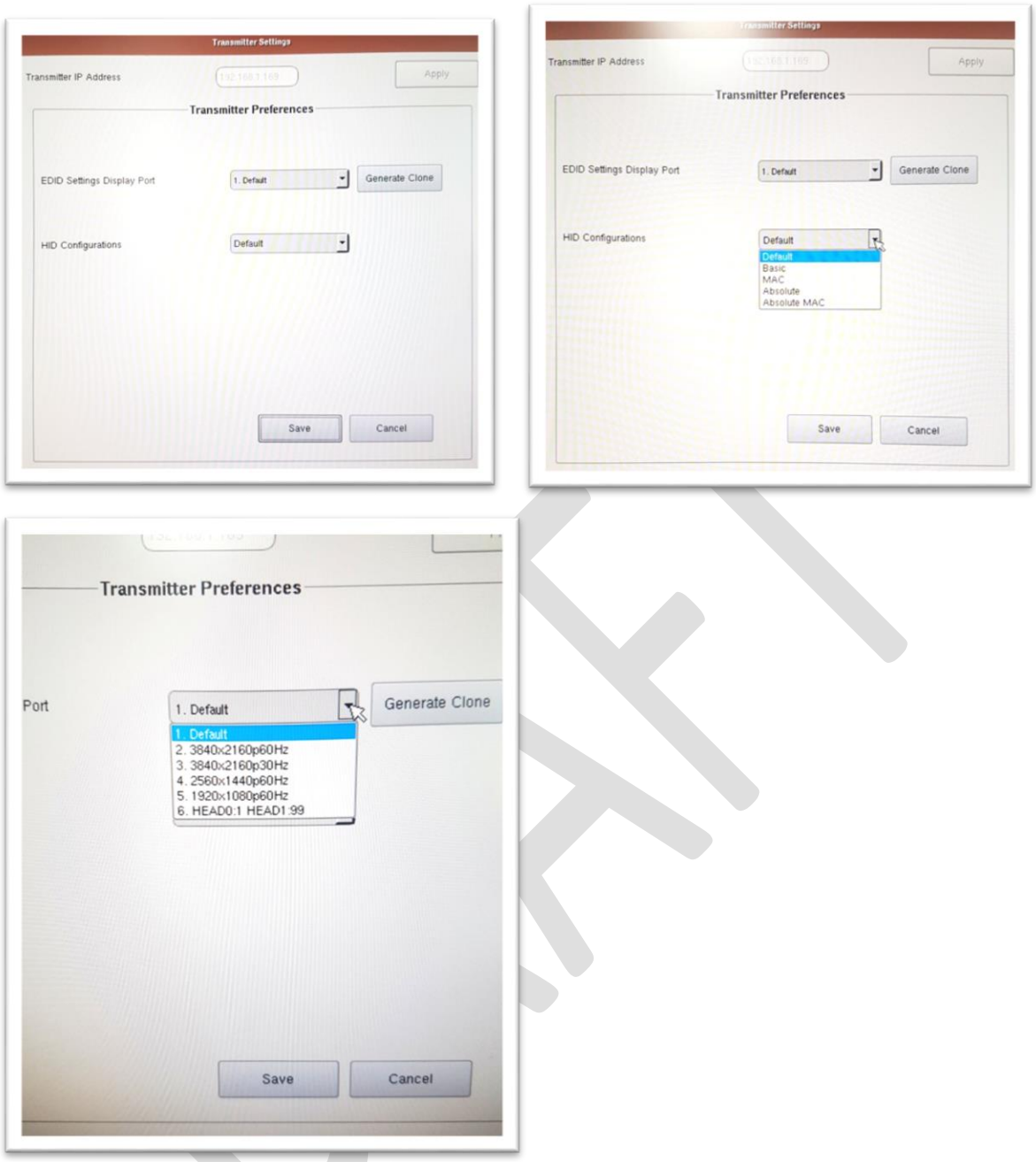

Figure 5 Transmitter Preferences 4K

12.2.1.4.1 Changes required

The 4K Transmitter settings page will need to change to provide configuration of Video Quality and Video Source Optimisation.

It may be possible to create a single common page with settings disabled as appropriate, for example:

For both 4K and 2K.

- Video Quality – Enabled.
- Noise Settings - Enabled.

For 2K

Note; It will not be possible to clone a 2K device from a 4K device. [14]

- EDID DVI 1 and Clone
- EDID DVI 2 and Clone

For 4K

Note; It will not be possible to clone a 2K device from a 4K device. [15]

- EDID Display Port

The OSD will need to implement the necessary logic to prevent EDID cloning from different technology types.

12.2.2 Software Upgrade

The Software Upgrade function on the OSD should be capable of handling both 4K and 2K image formats for encrypted images for Transmitters:

- 2K Transmitter Firmware – CLU File
- 4K Transmitter Firmware – .fit File[16]

In the case of Transmitter upgrade the OSD should only present image options for the relevant technology type.[17]

12.3 Boxilla

- Boxilla should only allow the provision of compatible connection types.

Boxilla will have to be upgraded to include additional logic implemented by the OSD.

12.4 Remote App

Any changes in the message exchanges of message fields will have to be updated in the Remote App. No additional changes are anticipated at this point but requirements to change these may still emerge.

It is expected the Remote App will present itself to a 4K Transmitter as 2K capable device in the same manner as it would present itself to a 2K Transmitter. The

4K Transmitter would scale a 4K video source to 2K for such a connection. Only YUV data is supported between a Remote App and a 4K Transmitter.

[14] In this phase of the project it will not be possible to clone a monitor connected to a 2K device from a 4K device and vice versa. This may be desirable at some point but any such analysis should be included as part of a more general analysis of our EDID cloning requirements.

[15] It this phase of the project it will not be possible to clone a monitor connected to a 4K device from a 2K device and vice versa. This may be desirable at some point but any such analysis should be included as part of a more general analysis of our EDID cloning requirements.

[16] Some feasibility may be required to determine if we can support 4K upgrade from 2K. We may be constrained by flash size or ram disk size.

[17] A common upgrade format could be considered at some point, but this is outside the scope of this feature.

13 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.

13.1 Physical Deployment

13.2 Unicast Mode

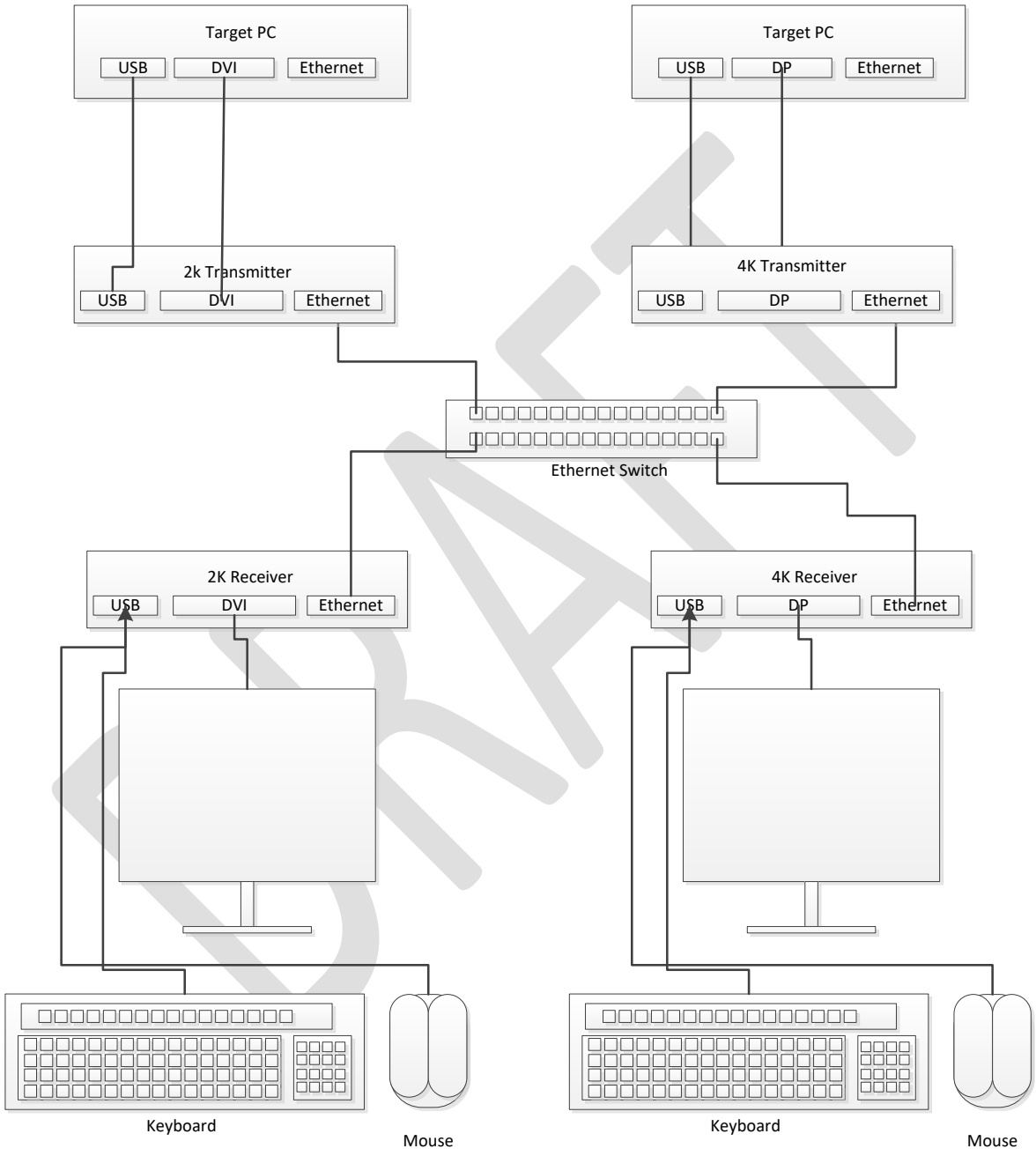

Figure 6 Basic Interoperability Deployment

13.3 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.

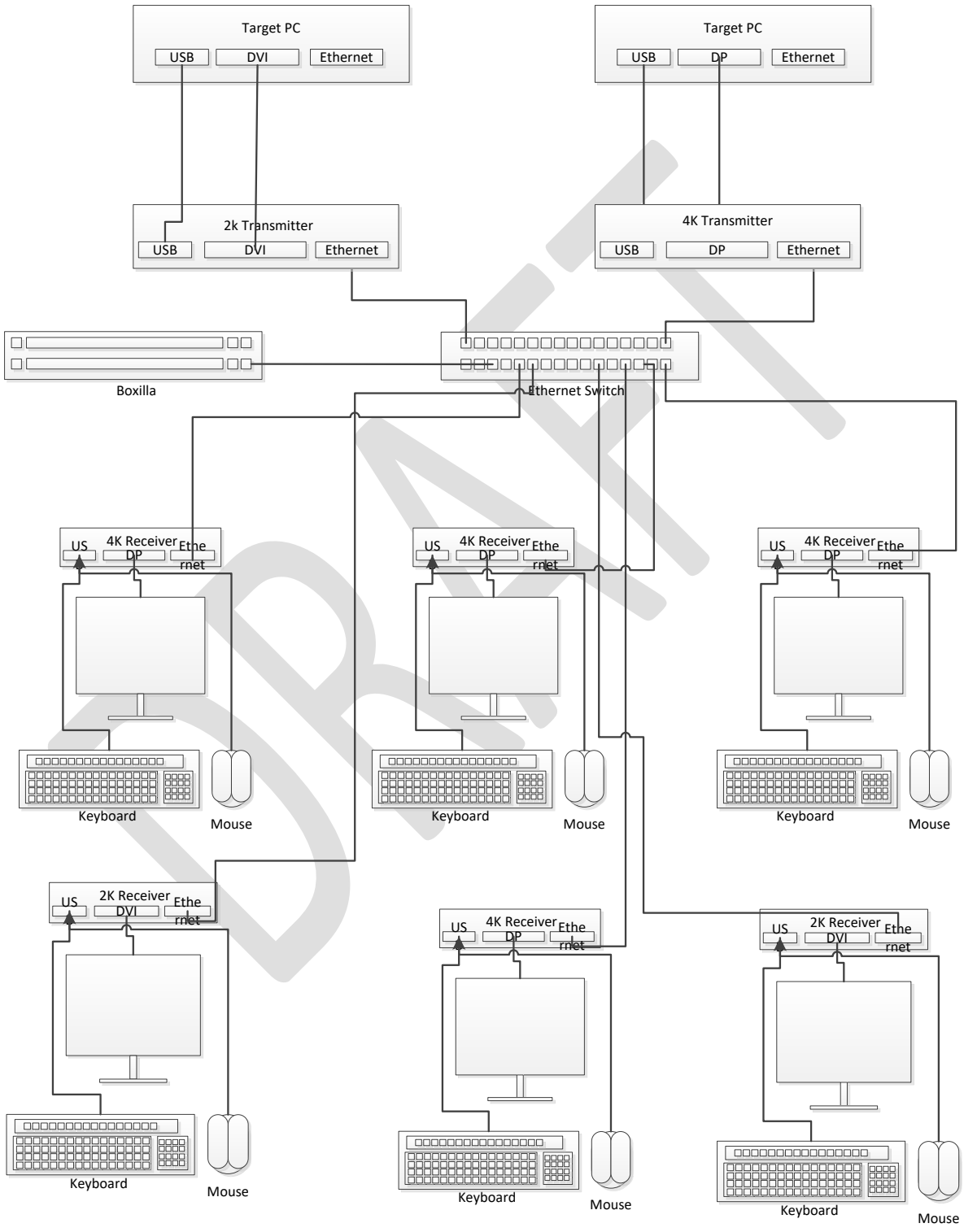

Figure 7 Shared Mode Deployment

13.4 Customer Deployment

13.4.1 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.

13.5 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

13.6 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.

As part of phase 2:

- An Emerald 4K receiver should be capable of upgrading a 2K Transmitter?
- An Emerald 2K receiver should be capable of upgrading a 4K Transmitter?

13.7 Phasing

This version of the document focuses on phase 2 of the project.

13.8 Estimates

13.8.1 Documents Effort

Document	Type	Estimate Weeks	Comments
Update to the Remote App specification	SDD	.5	See Section 12.5
Summary of Rest Changes in REST API document	API Definition	.5	See Section 12.3
Capture Layer	SDD	1	Micheal/John
Connection Management Refactoring for speed	Architecture	.5	

and efficiency			
EDID management	SDD	.5	Tom/Micheal//John
Scaling API	SDD	.5	Micheal
Total		3.5	

13.8.2 Coding Effort

Functional Area	Description of Work	Estimates Weeks	Comments	
REST API	1. Review current interfaces with respect to unified TX settings 2. Possibly look at mechanism for filtering connections which cannot be completed (This would have to be estimated)	.50		Marcus
OSD	1. 4K Transmitter settings page changes.	.25		Olga
OSD	1. Diagnostics needs to correctly invoke and the corresponding functionality on 2K or 4K to transfer required diagnostics files. It should work so this is largely a developer validation effort	.25		Olga
OSD	1. System restore needs to correctly invoke and the corresponding functionality on 2K or 4K. It should work so this is largely a developer validation effort	.25		Olga
Software Upgrade	1. Feasibility to determine if 2K devices can support upgrade of 4K devices.	1		Marcus

Networking	Connection Management scripts refactor and review	1		Dan
Capture Layer	1. 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.	2	Work on this has been progressing in the background. The estimate represents the remaining work.	Micheal
Error Handling	The Error handling subsystem will need to be updated to handle errors associated with non supported connection modes	1		Olga/Manas
Scaling API	The scaler in the FPGA is largely untested. The functionality proposed is per the DPS. We estimate 2 weeks work to develop a suitable API and integrate it into the Encoder functionality. But we also anticipate significant prototyping in conjunction with the FPGA team to stabilise this functionality. This is catered for in the schedule estimates.	2		Micheal
EDID Management	The interoperability between 2K and 4K poses particular challenges for EDID	2		Micheal/Tom

	management. The 4K EDID will contain resolutions and audio properties that a 2K device will not be able to support. We need a scheme that manages this based on yet to be defined marketing requirements.			
Networking	Remove and disable Encoder Video Engine RPU	.50		Manas
Networking	Add support for video frame markers	3		Manas
Networking	Connection Management Refactoring for speed and efficiency	1	Work on this has been progressing in the background. The estimate represents the remaining work of integrating this into the interop_p2 branch	Dan/Manas/Micheal
Total		15.5		

13.8.3 Development Testing Effort

Functional Area	Description of Work	Estimates Weeks	Comments	
REST API	3. Review current interfaces with respect to unified TX settings	1		Marcus
OSD	2. 4K Transmitter settings page changes.			Olga
OSD	2. Diagnostics needs to correctly invoke the corresponding functionality on 2K or 4K to transfer required diagnostics files. It should work so this is largely a developer validation effort	.25		Olga

OSD	2. System restore needs to correctly invoke and the corresponding functionality on 2K or 4K. It should work so this is largely a developer validation effort	.25		Olga
Software Upgrade	2. Feasibility to determine if 2K devices can support upgrade of 4K devices.	1		Marcus
Networking	Connection Management scripts refactor and review	1		Manas
Capture Layer	2. 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.	3		Micheal
Error Handling	1.The Error handling subsystem will need to be updated to handle errors associated with non-supported connection modes	1		Olga/Manas
Scaling API		1		Micheal
EDID Management		1		Tom/Micheal
Networking	Remove and disable Encoder Video Engine RPU			Manas
Networking	Add support for video frame markers	5		Manas
Networking	Connection Management Refactoring for speed and efficiency	8		Dan/Manas
Total		22.5		

Estimated Total Development and Development Testing Effort: 37.5 weeks. The longest individual effort is Micheal's at 12 weeks.

14 Notes

Notes: Handling DP resolution with a refresh rate > 60Hz needs to be analysed.