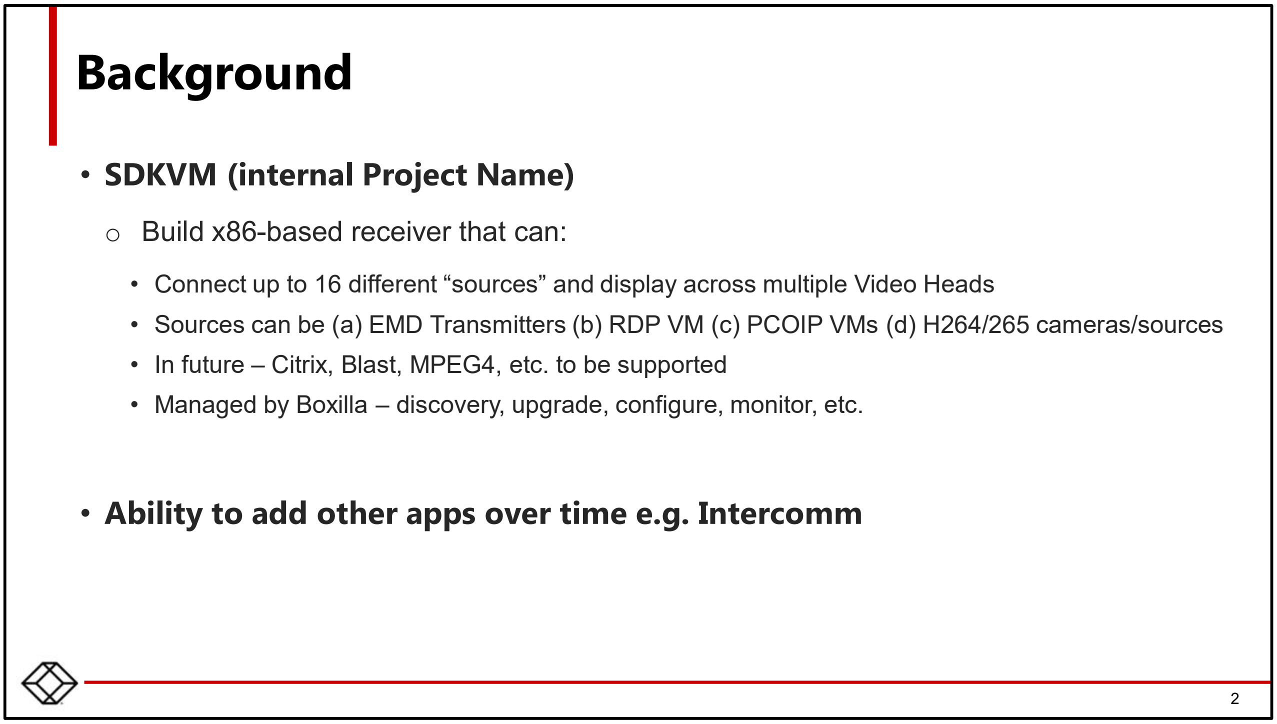
Black Box – SQ – Flutter POC

REV 1

28Jul22

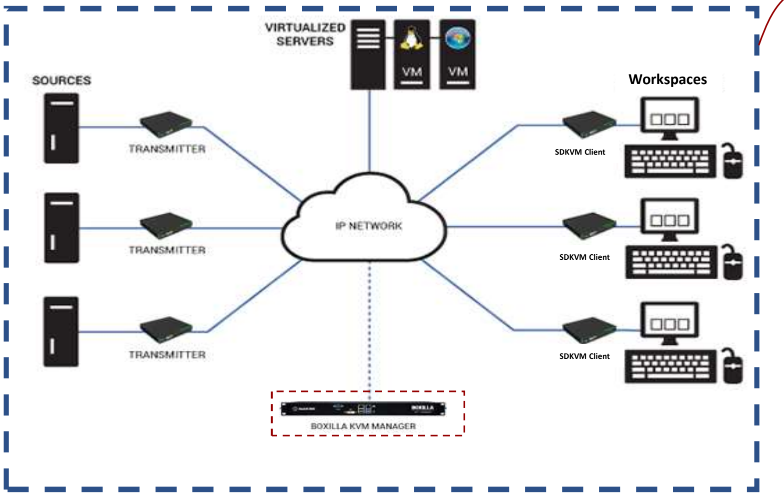


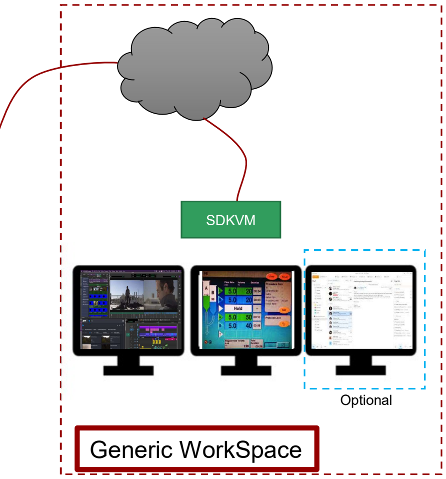
Background

- **SDKVM (internal Project Name)**
  - ○ Build x86-based receiver that can:
    - Connect up to 16 different “sources” and display across multiple Video Heads
    - Sources can be (a) EMD Transmitters (b) RDP VM (c) PCOIP VMs (d) H264/265 cameras/sources
    - In future – Citrix, Blast, MPEG4, etc. to be supported
    - Managed by Boxilla – discovery, upgrade, configure, monitor, etc.

- **Ability to add other apps over time e.g. Intercomm**

SDKVM Deployment

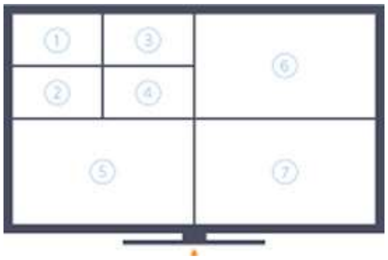




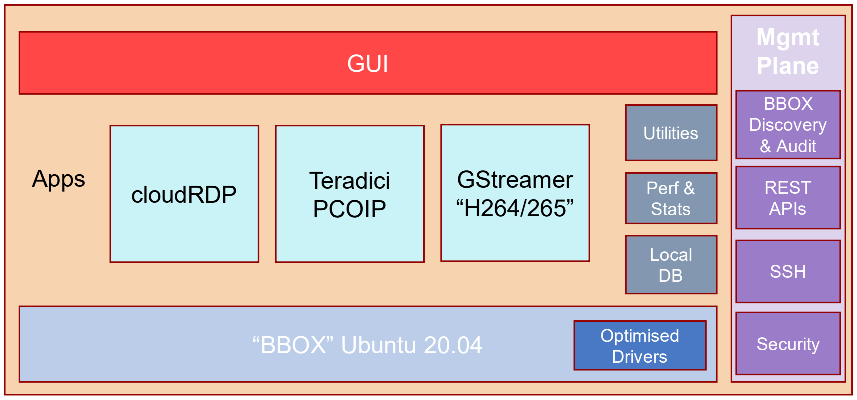
User-Login creates desk profile:

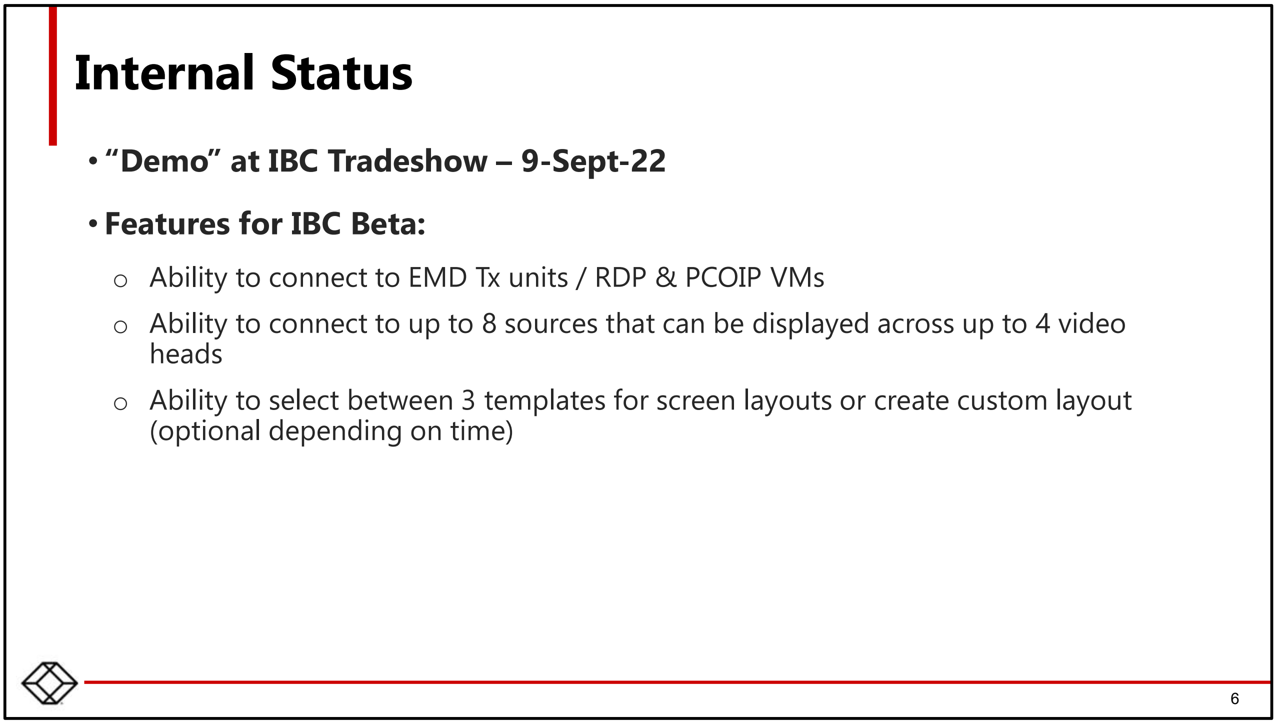
1. Standard-user (locked down user)
2. Power-user (ability to update templates)
3. Admin user (full access to change settings, templates, users)

Templates



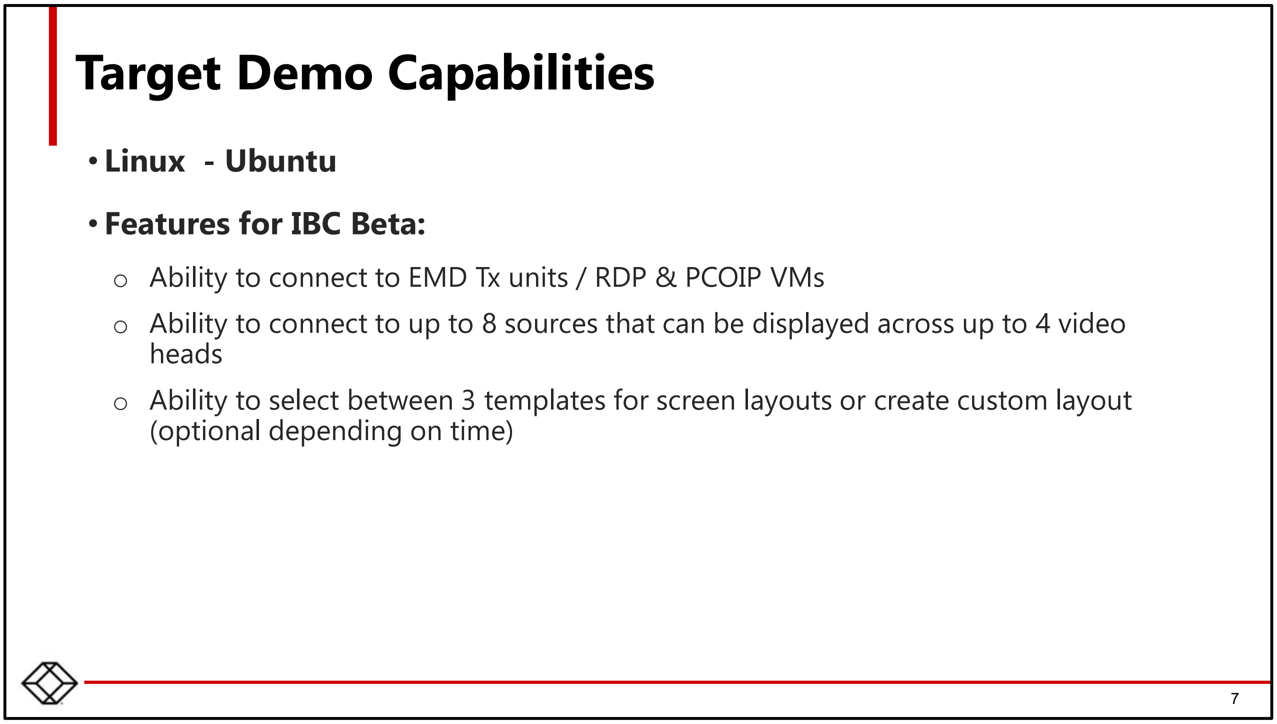
Architecture





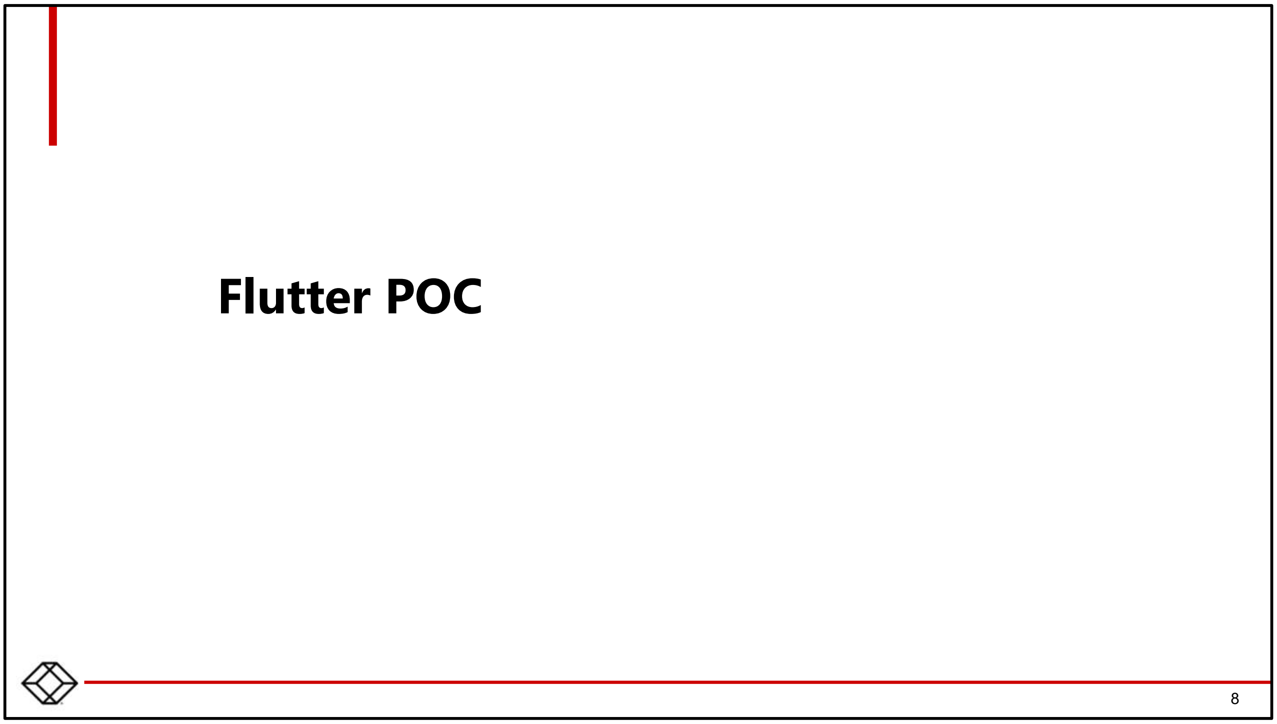
Internal Status

- “Demo” at IBC Tradeshow – 9-Sept-22
- **Features for IBC Beta:**
  - Ability to connect to EMD Tx units / RDP & PCOIP VMs
  - Ability to connect to up to 8 sources that can be displayed across up to 4 video heads
  - Ability to select between 3 templates for screen layouts or create custom layout (optional depending on time)



Target Demo Capabilities

- **Linux - Ubuntu**
- **Features for IBC Beta:**
  - Ability to connect to EMD Tx units / RDP & PCOIP VMs
  - Ability to connect to up to 8 sources that can be displayed across up to 4 video heads
  - Ability to select between 3 templates for screen layouts or create custom layout (optional depending on time)



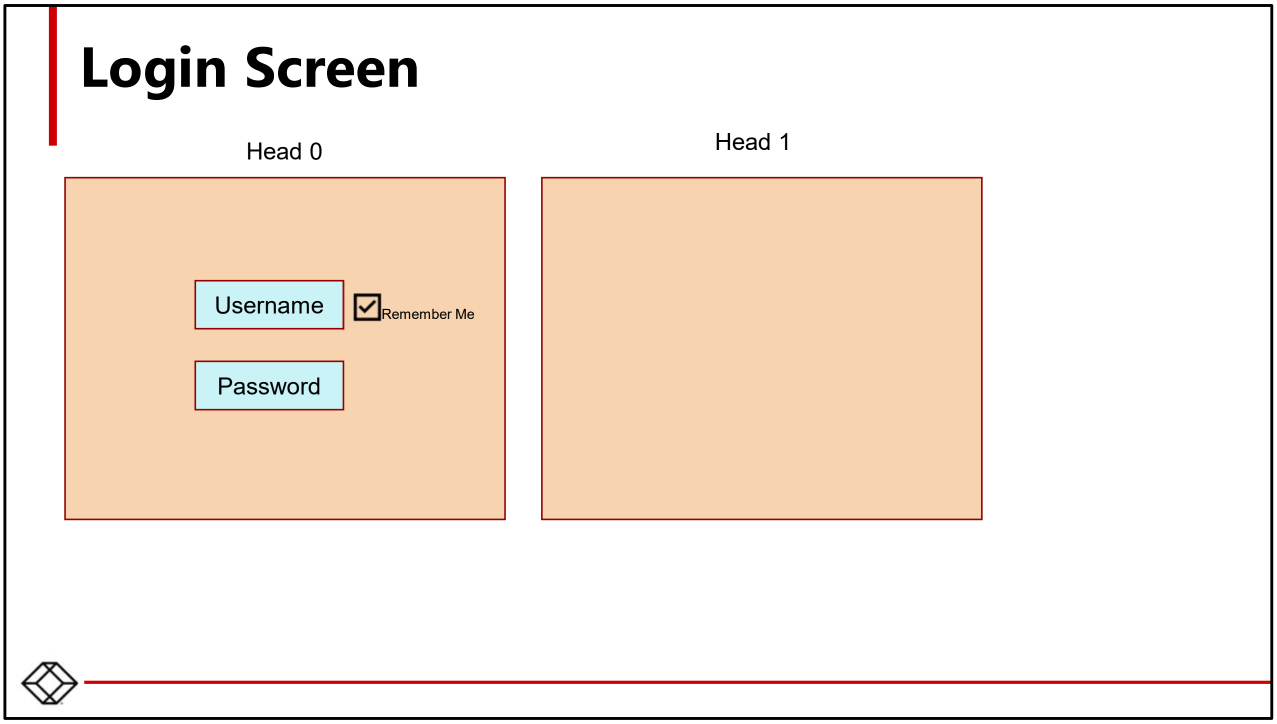
Flutter POC

Notes

- **All screens assume dual-head video**
  - Generally initial work will focus on one-screen (so second will be “blank” most of the time);
- **Colours/Text font are only for illustrative purposes – feel free to use any other others or text type/font/etc. for the initial POC work**
  - Initial POC is more focused on checking Flutter is capable of doing what we want – no real focus on UX side of things.
- **Head 0 and Head 1 are labels used to identify video head on PC running the application**
  - Head 0 is considered the ”primary” head

- **Head 0 and Head 1 are labels used to identify video head on PC running the application**
  - Head 0 is considered the ”primary” head

- **Read notes section of slide to check for more information on workflow**

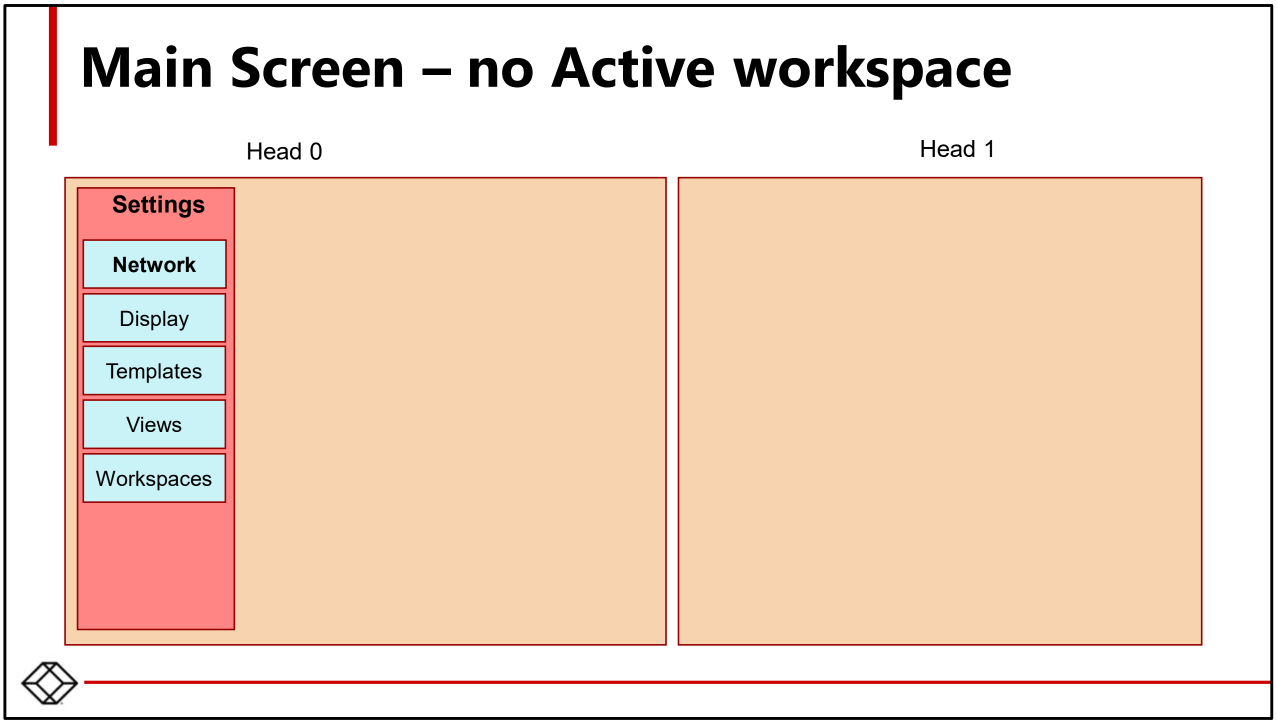


Login Screen

This is a typical login screen. User enter username/password – application checks – can hardcode that only username: admin and password: admin will be allowed to login.

If login fails a pop-up window is display – says “Username/Password incorrect” with a “Ok” button on window. User has to check “ok” to dismiss the pop-up window.

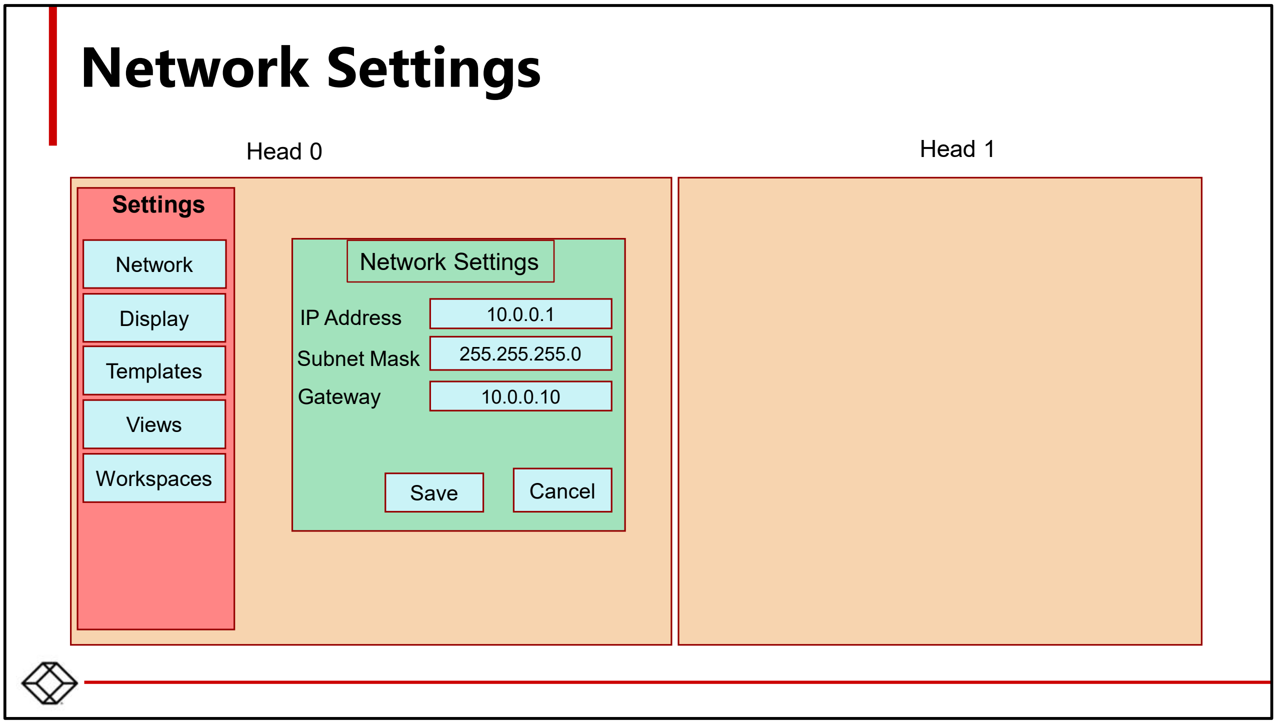
The remember me button is a check-box. Starts up “unchecked”. It user clicks on button – “check” is displayed and remembered by App. What this means if the check box is set – the last password that successfully passed login is remembered and displayed when login screen is display. The status of this check-button should be preserved between application launches (ie is remembered/is persistent on device).



Main Screen – no Active workspace

After a successful login – the user is brought into the Main-screen. On the left is a menu bar – called the Settings menu. The rest of the video 0 & video 1 are initialised in some default background across the screen (should be full screen).

Only Network and Workspace items on the menu will have "action" behind them for initial POC. Look/feel of Settings menu not important initially – just needs to be functional.



Network Settings

IP Address 10.0.0.1

Subnet Mask 255.255.255.0

Gateway 10.0.0.10

If the user clicks on “Network” item – a pop-up window is displayed called Network Settings Window.

This window reads the network settings from the device it is running on (all IPV4 addressing to be used) and display them as the values in the IP Address, Subnet Mask and Gateway fields.

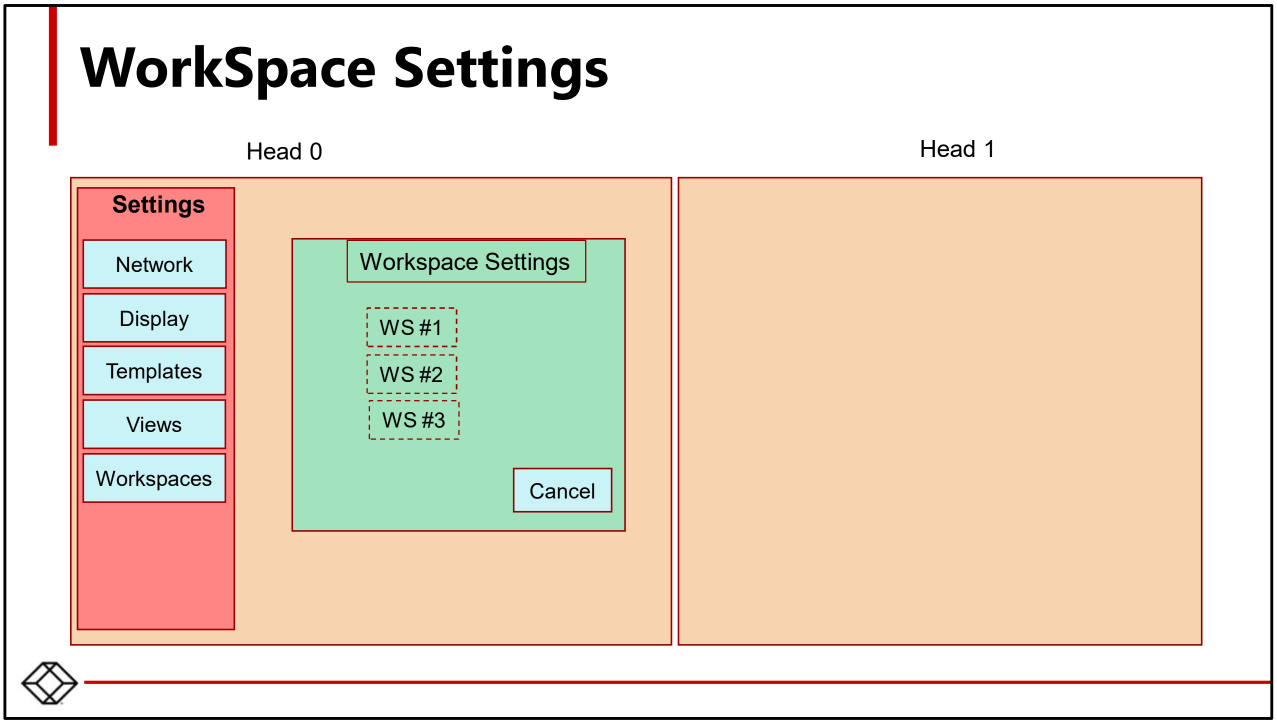
These 3 fields are editable by user. When the Save button is clicked – the application checks (a) the IP address is a valid IPv4 address (uses 4 values between 0 and 255 separated by a “.” (b) that Gateway is on same subnet as new IP address (i.e. when subnet is applied the network subnets match for IP address and gateway)). If address is validated then it should be stored on device as new IP address details.

For the purpose of the POC – initially these can be created in App / Saved to App – each time App loaded back to initial IP address settings.

Hitting the save button commits validated changes or pop-ups an error message with explanation on why validation failed. OK button on Error pop-up window to allow Error Message to be dismissed and user brought back to Network Settings again to update.

If Cancel is clicked – Network Settings screen is dismissed with no changes to IP addressing settings and user is brough back to Main Screen.

When the Network Settings window is active – the Settings Menu is “inactive” i.e. cannot select any item on it until Network Settings window is closed.



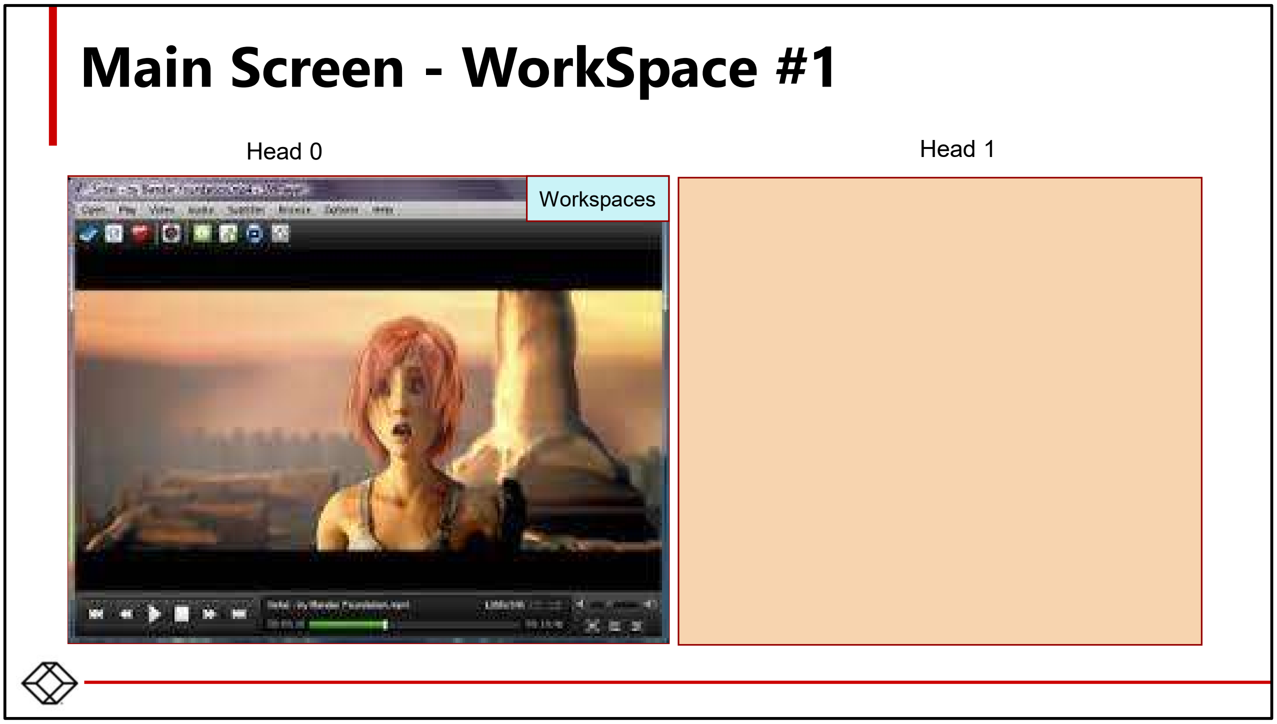
WorkSpace Settings

If the user clicks on "Workspaces" item – a pop-up window is displayed called Workspace Settings Window. This has 3 items – WS #1, WS #2, WS #3.

The user can click on one of these 3 items and then this causes that Workspace to become active and be the "top layer" on the screens (see next slide).

If the user clicks on the Cancel button – no other action is taken other than closing the Workspace Settings Window.

When the Workspace Settings window is active – the Settings Menu is "inactive" i.e. cannot select any item on it until Workspace Settings window is closed.

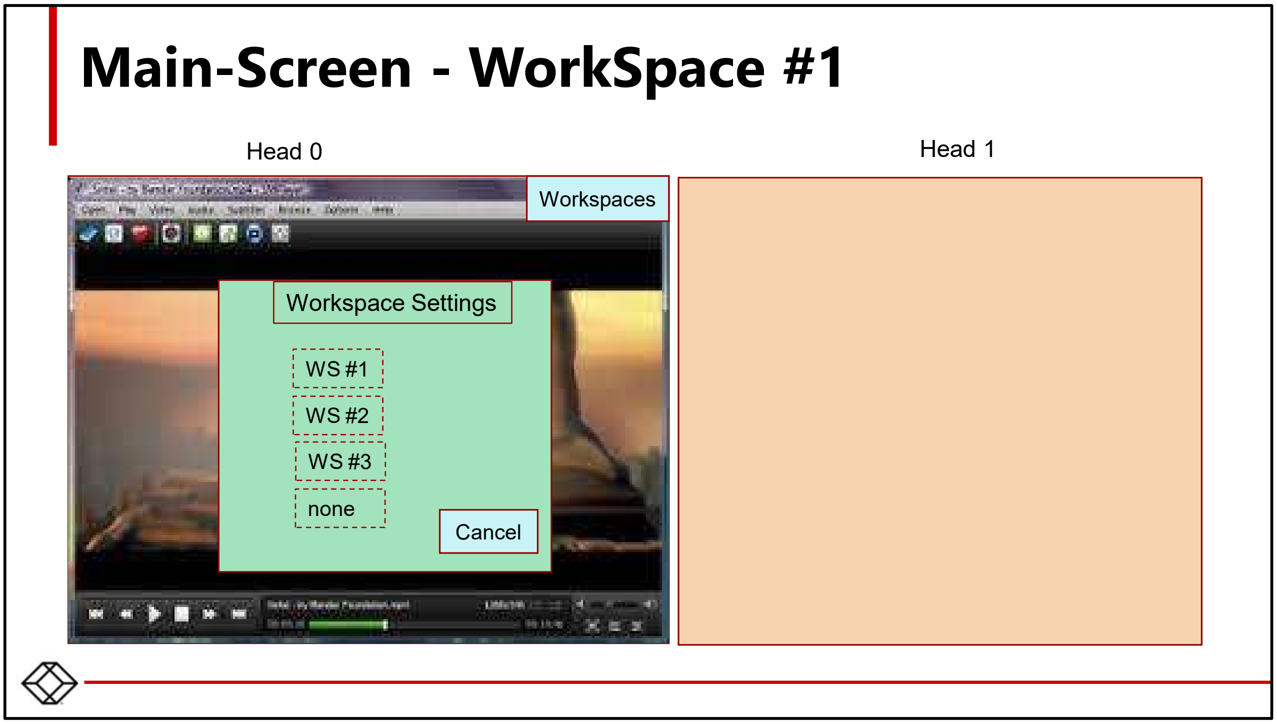


Main Screen - WorkSpace #1

If the user clicks on “WS #1” item – the screen is laid out as shown i.e. a video player (running on OS) is launched and placed to be full screen on Head 0. There is a “button” that is added on top of the video player on top right corner (can be put in another place if needed – just need to ensure it does not cover a needed button to operate player).

The user should be able to start/stop the video in the player – open file menu on player and select another file to be played.

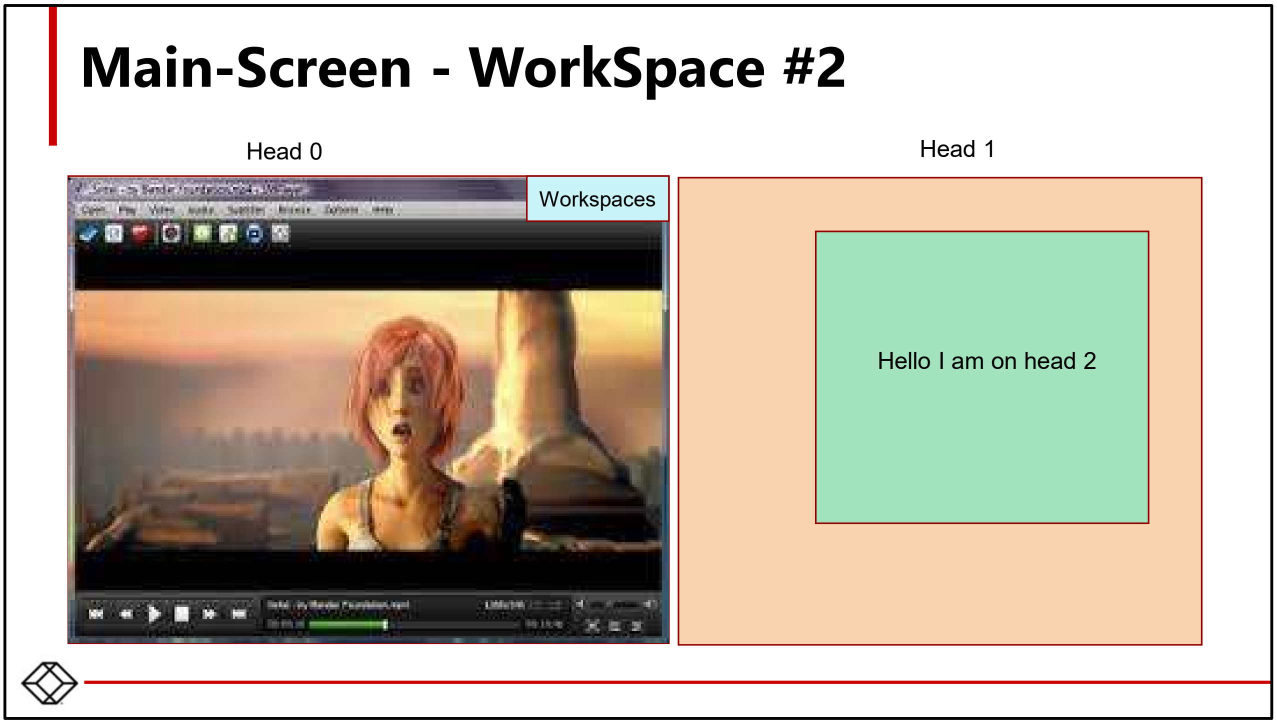
The user should be able to click on the Workspaces window to be able to select a new work space.



Main-Screen - WorkSpace #1

If the user selects Workspaced button – a pop-up window overlays the video player as shown. The user can select to move to a new workspace by selected from the items in the list. Select “none” – takes the user back to Main-Screen – no active workspace layout as shown before.

Clicking on “cancel” button means the Workspace settings pop-up window is closed and no changes made – i.e. user stays on workspace #1.



Main-Screen - WorkSpace #2

Head 0

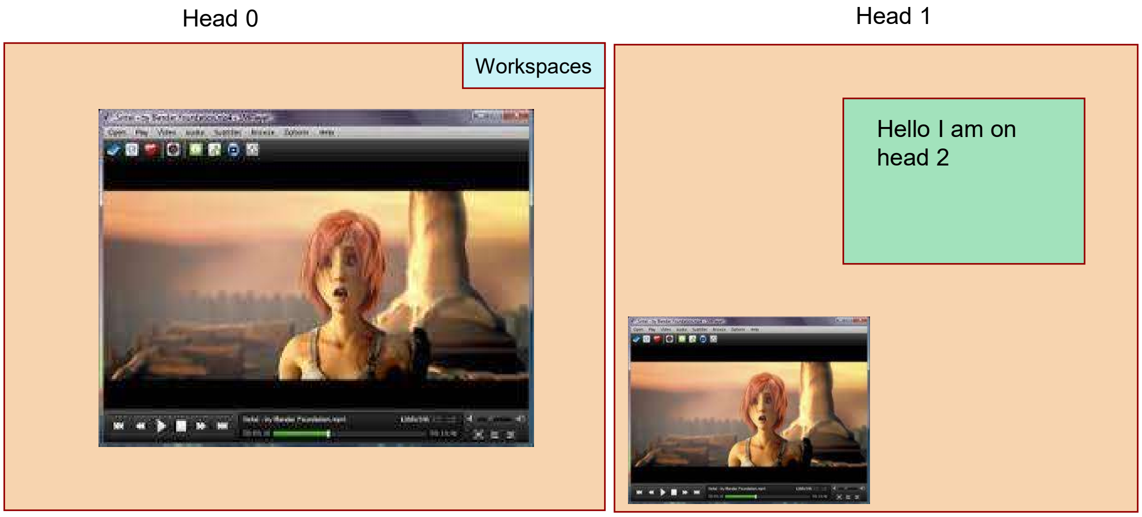
Workspaces

Head 1

Hello I am on head 2

If the user selects WS #2 – the layout is as shown above. Really only adding an item to be displayed on Head1.

Main-Screen - WorkSpace #3



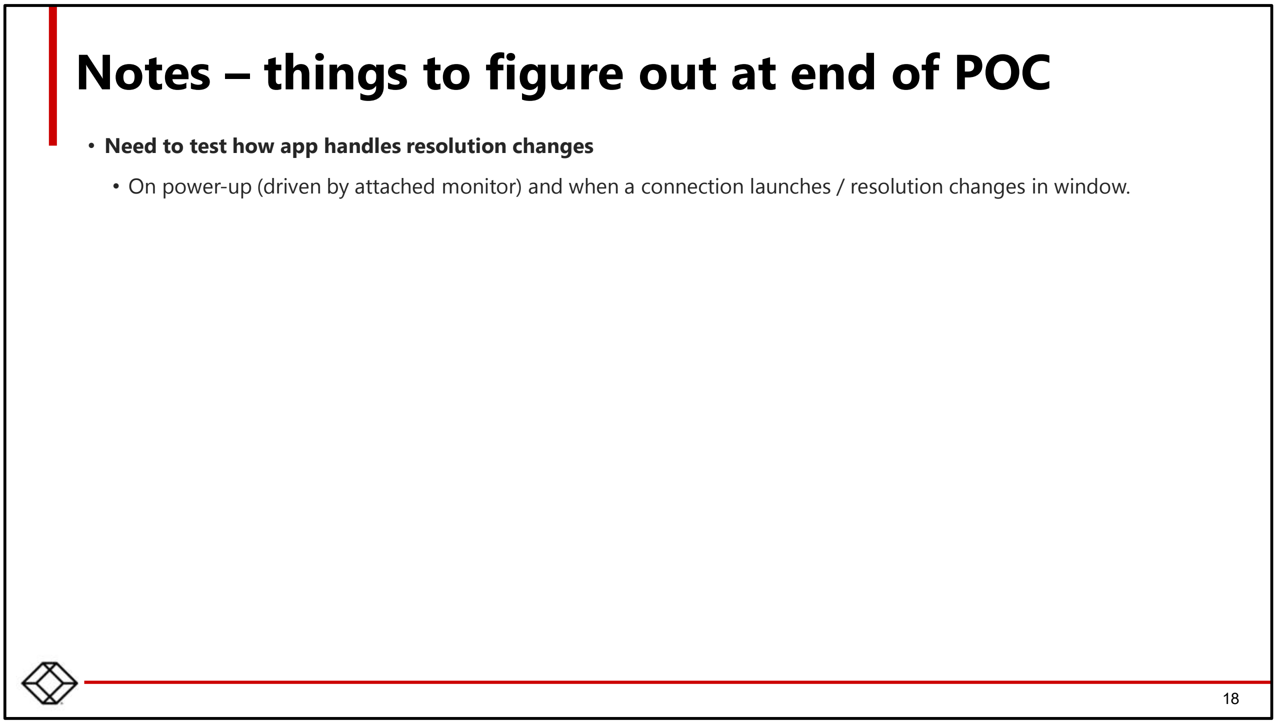
Head 0

Head 1

Workspaces

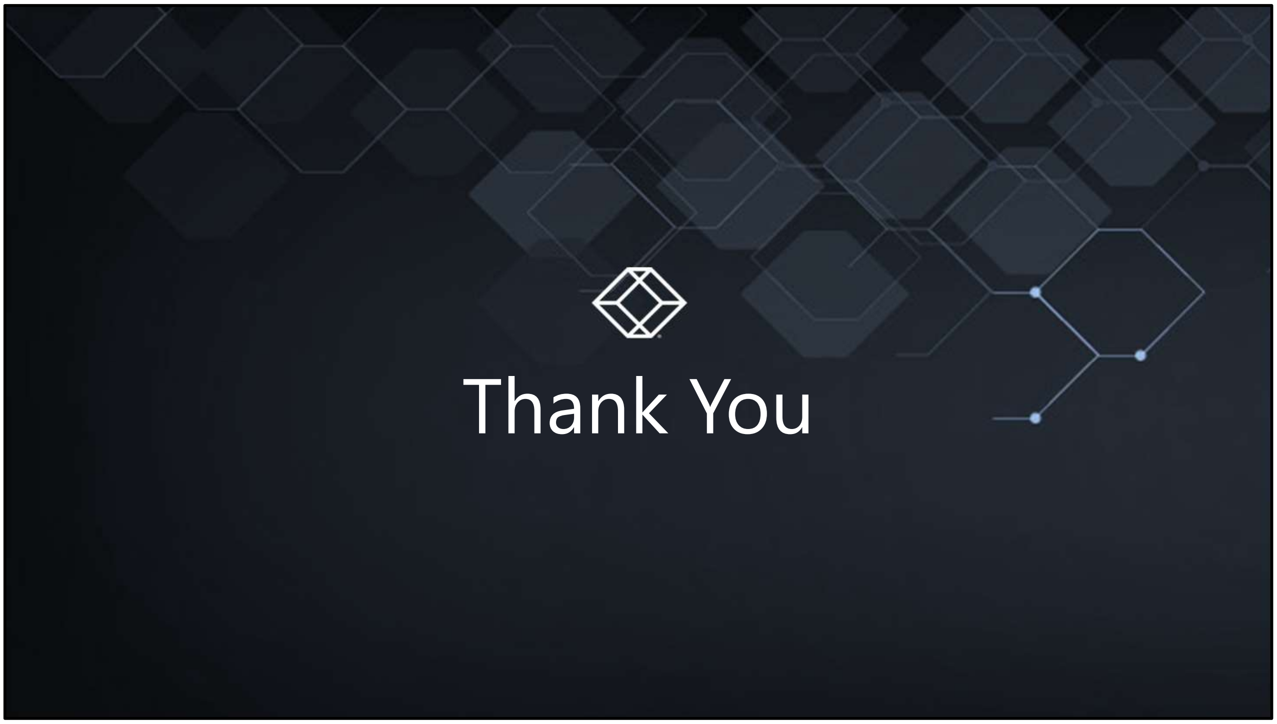
Hello I am on head 2

If the user selects WS #3 – the layout is similar – just that player on Head0 is reduced to less than full screen - 2/3 of screen. While a player that is roughly ¼ of head 1 is launched as well.



Notes – things to figure out at end of POC

- **Need to test how app handles resolution changes**
  - On power-up (driven by attached monitor) and when a connection launches / resolution changes in window.



Thank You