

Deliverable

Project Acronym:	ImmersiaTV
Grant Agreement number:	688619
Project Title:	<i>Immersive Experiences around TV, an integrated toolset for the production and distribution of immersive and interactive content across devices.</i>

D3.8 Workflow integration & End-to-end tests

Revision: 0.15

Authors:

Maciej Glowiak, Szymon Malewski (PSNC)

Delivery date: M22

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 688619

Dissemination Level

P	Public	x
C	Confidential, only for members of the consortium and the Commission Services	

Abstract: This deliverable describes the process and methodology of integration components of the Immersia TV project. It contains information on how the releases were prepared and what functionalities appeared in each release. The document also contains description on testing process and lessons learned from the testing the integration process.

REVISION HISTORY

Revision	Date	Author	Organisation	Description
0.1	22-08-2016	M.Glowiak	PSNC	Deliverable structure, first contribution, list of releases
0.2	30-08-2016	Sz.Malewski	PSNC	Methodology and Testing procedures
0.3	31-08-2016	M.Glowiak	PSNC	Additional descriptions, styling, diagrams
0.4	6-03-2017	M.Glowiak	PSNC	Updated version
0.5	10-03-2017	J.Nunez	I2CAT	Additional improvements
0.6	16-03-2017	J. Lourenço	Lightbox	Tests of production tools
0.7	17-03-2017	M.Glowiak Sz.Malewski	PSNC	Integration, minor improvements
0.8	03-10-2017	M.Glowiak	PSNC	Include new changelogs and releases versions up to v.0.20
0.9	11-10-2017	Sz.Malewski	PSNC	Update for live scenario testing
0.10	12-10-2017	M.Glowiak	PSNC	Updates, corrections and comments for additional partner's contribution
0.11	13-10-2017	S.Valente	VS	Additional contribution regarding capture tools
0.12	18-10-2017	A.Kelembet	CGY	Additional contribution regarding live production
0.13	20-10-2017	J.Nunez	I2CAT	Additional contribution
0.14	20-10-2017	M.Glowiak	PSNC	Review and corrections
0.15	27-10-2017	M.Pacheco, H.Campos, J. Lourenço	LBOX	Review and corrections

Disclaimer

The information, documentation and figures available in this deliverable, is written by the **ImmersiaTV** (*Immersive Experiences around TV, an integrated toolset for the production and distribution of immersive and interactive content across devices*) – project consortium under EC grant agreement H2020 - ICT15 688619 and does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

Statement of originality:

This document contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

EXECUTIVE SUMMARY

This deliverable describes the workflow integration process and methodology as well as contains information of end-to-end testing procedures. The document is related to other project deliverables, especially D3.1 – Design Architecture, D3.2-D3.6 software deliveries and D4.4 (Technical evaluation), D4.5 (User evaluation). The intention was to create this deliverable incrementally and iteratively, building further versions basing upon this one. This version focuses on integration, delivery software releases and tests done for Pilot 1, what means it contains information dedicated to off-line production and playback workflow until release 0.6.

Chapter 1 contains introduction to the document and Chapter 2 is dedicated to description of components (2.1) and functionalities of particular releases (Section 2.2). In Chapter 3 there are information on workflow integration process (Section 3.1), methodology (Section 3.2) and End-to-end tests (Section 3.3). Document is shortly concluded by Chapter 4.

CONTRIBUTORS

First Name	Last Name	Company	e-Mail
Maciej	Głowiak	PSNC	mac@man.poznan.pl
Szymon	Malewski	PSNC	szymonm@man.poznan.pl
Juan	Nunez	I2CAT	juan.antonio.nunez@i2cat.net
João	Lourenço	Lightbox	joao@lightbox.pt
Maciej	Stróżyk	PSNC	mackostr@man.poznan.pl
Alexandr	Kelembet	CGY	kelembet@cinegy.com
Stephane	Valente	VS	stephane@orah.co

CONTENTS

Revision History	1
Executive Summary.....	3
Contributors	4
Table of Figures.....	7
List of acronyms.....	8
1. Introduction	9
1.1. Purpose of this document.....	9
1.2. Scope of this document	9
1.3. Relation with other ImmersiaTV activities.....	9
2. Releases.....	10
2.1. Components.....	10
2.1.1. Player	10
2.1.2. Production tools	10
2.1.3. Server	11
2.1.4. Capture.....	11
2.2. Description of releases	11
2.2.1. Releases 0.2 - 0.7	11
2.2.2. Releases 0.8 – 0.20.....	12
2.2.3. Release notes for Player.....	12
2.2.4. Release notes of Web Player	25
2.2.5. Release notes of Off-line Production Tools	25
2.2.6. Release notes of Live Production Tools	32
2.2.7. Release notes for Server	34
2.2.8. Release notes of VideoStitch Studio	37
2.2.9. Release notes of Vahana VR.....	41
3. Integration	45
3.1. Workflow integration	45
3.2. Methodology	48
3.3. End-to-end tests	49
3.3.1. Internal system tests.....	49
3.3.2. Testing workflow in demo environment.....	50
4. Conclusions	54

TABLE OF FIGURES

Figure 1: Role of T3.8 within the ImmersiaTV platform.....	9
Figure 2: Architecture for ImmersiaTV system for Pilot 1.....	46
Figure 3: Architecture for ImmersiaTV system for Pilot 2.....	47
Figure 4: Process of software integration	47
Figure 5: through the immersiaTV services, the content creator can convert his content to DASH and publish the content for synchronized playout during content consumption.....	50
Figure 6: TNC demo	52
Figure 7: TNC2017 demo on photographs. On the left there is CINEGY transport tools integrated with the Elmo Rig camera. On the right photograph, Immersia TV team is presenting live components.....	53
Figure 8: IBC demo.....	54
Figure 9: Immersia TV demo booth at IBC Future Zone in Sep 2017	54

LIST OF ACRONYMS

Acronym	Description
HMD	Head Mounted Display
WP	Work Package
D	Deliverable
T	Task

1. INTRODUCTION

1.1. Purpose of this document

In this report, partners of the Immersia TV consortium describes the outcome of the tests and system integration. The document explains the results of tests, current limitations and suggestions for architecture and implementation improvements. This document will describe the integration process and tests done iteratively across all 3 phases of the work conducted in the project.

1.2. Scope of this document

This document focuses on integration process, test methodology and tools used for management of development process. It contains short description of all releases.

The document bases on the D3.1 (Architecture design) and other deliverables of WP3 (dedicated to particular components provided by partners D3.2 – D3.6). As some tests were described in D4.4 (Technical Evaluation) and D4.5 (User Evaluation), this deliverable just refers to them if necessary.

1.3. Relation with other ImmersiaTV activities

This deliverable is part of task T3.8 (Workflow integration & End-to-end tests) in WP3 (Immersive Broadcast Platform). The relationship between this task and the other WP tasks is shown below.

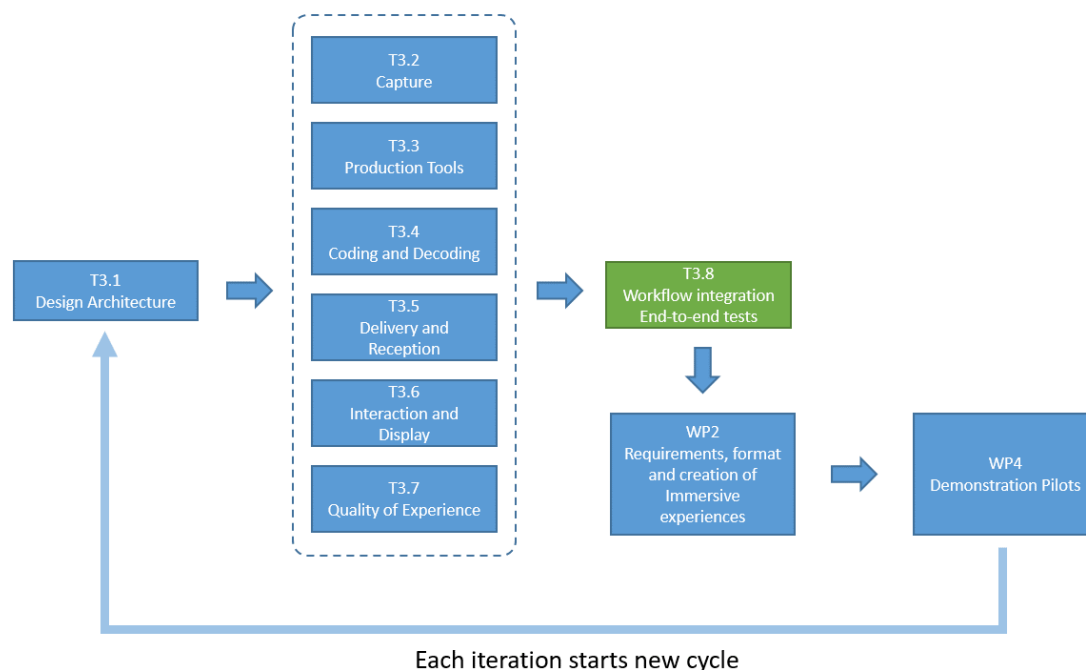


Figure 1: Role of T3.8 within the ImmersiaTV platform

2. RELEASES

2.1. Components

ImmersiaTV aims to distribute omnidirectional and directive audiovisual content simultaneously to head mounted displays (HMD), companion screens and the traditional TV. The content distributed is constituted of one or more omnidirectional videos, complemented with several directive shots, and metadata detailing how to merge these streams in an immersive display, in coordination with directive and omnidirectional videos also shown in traditional TVs and tablets.

As detailed in Deliverable D3.1 - Architecture Design, the ImmersiaTV platform involves the following modules: Video Capture, Video Stitching, Post-Production, Encoding, Distribution, Reception & Interaction and QoE component.

In order to achieve Pilot 1 functionality (off-line workflow), not all components were published as releases available for testing. The content capture was handled using camera rigs, and for stitching VideoStitch Studio and AutoPano were used, as it was described in details in D3.1. While partners responsible for advanced encoding, live production tools and QoE were developing their modules for next phases, the first releases focused for off-line scenario.

The Release 0.1 was made internally by I2CAT for testing tools and the process of creating release package. It was not published for partners, as it's not mentioned on the list below. The releases 0.2 - 0.7 contained two most important components for Immersia TV system for Pilot 1: Player (part of the Reception and Interaction) and Off-line production tools. Release 0.6 and later contained also web DASH server for use in demonstration environment. The detailed description of this group of releases is mentioned in Section 2.2.1.

Releases 0.8 and later focused mostly on bug fixing (stability) and introducing new features in existing services. These releases were prepared also for live scenario and implemented functionalities required for Pilot 2. Capture and live post-production tools were not part of the release, as they were prepared separately by VIDEOSTITCH and CINEGY. The integration of these tools were done using interfaces and common protocols, so one package bundled together all the component wasn't necessary. The detailed description of this group of releases is mentioned in Section 2.2.2.

2.1.1. Player

This component handles the end-user's reception side and display. It takes care of selecting proper video streams, receiving, decoding and displaying them. It also handles the synchronization of multiple received streams in order to present them to the end-user. The user can interact with the received content by selecting streams, choosing the device and performing basic playback actions. The client application provides logging functionality of playback parameters, which can be analysed off-line.

2.1.2. Production tools

This component encompasses a set of tools and plugins for video editing with functionality of synchronization and combining multiple 2D and omnidirectional video sources and auxiliary data together. These data come from Capture and Stitching blocks.

Off-line production tools operate on video files from omnidirectional and directional cameras and allows the editor to select, modify and combine streams using various transition effects and produce metadata. Off-line production tools produce video files as well as XML metadata that needs to be uploaded to Content Distribution server.

Live production tools have similar functionality, however they operate on network streams, switching them and if necessary transcoding. They output final streams as MPEG-DASH ones with complementary XML metadata.

2.1.3. Server

This component is responsible for streaming out DASH streams to the Player. In Phase 1 server takes files from local repository and in Phase 2 it receives external streams from Live Production tools.

The content is always streamed as MPEG-DASH adaptive streams taking advantage of the computing power and screen resolution of the different devices where the player consume the contents, transcoding is performed in this phase.

For Live scenario Cinegy Transport machine is responsible for streaming MPEG-DASH content. So, there is no requirement for intermediate proxy server, although it may be used during Open Pilot to provide scalability. In such case vanilla Nginx server is expected to be used as proxy.

2.1.4. Capture

Capture in Phase 1 was done by recording omnidirectional video using Elmo Rig and GoPro rig, and then processing raw video files using Video Stitch Studio and Auto Pano. For Phase 2, although we may use Vahana VR (with Elmo Qbic rigs), live spherical camera systems were preferably used, like imec EDM's Studio.One and Orah 4i cameras.

2.2. Description of releases

2.2.1. Releases 0.2 - 0.7

In this period the main goal was to achieve functional demo. Releases were introducing new components from the simplest player and production tools to full set realizing most of requirements for Pilot 1 (release 0.6 – demo on IBC 2016, release 0.7 – demo on NEM 2016).

Released software contained components such as:

- ImmersiaTV player for android, windows and android TV
- Premiere Pro extension with mac and windows installers
 - Multiplatform content generation (tablet, HMD, TV)
 - Transitions based on luma masks (with basic interaction).
 - Automatic generation of DASH files.
- Dash conversion and content publication services
- Tutorial explaining how to create scenes covering the main cases of ImmersiaTV, and how to use the ImmersiaTV services
- Session Manager

2.2.2. Releases 0.8 – 0.20

In the next releases major changes were less rapid, however we improved stability and robustness of components. There were twelve new releases in that time, because the consortium performed a lot of tests using their components, so they need all the bugs, optimizations and missing functionalities implemented and released quickly. New features were tested and steadily developed.

For the production tools, experiences from Pilot 1 preparation were taken into account, better solutions for discovered problems were tried. Luma mask based transitions were replaced by fixed list of transitions, portals definitions were reviewed and last but not least, the user interface was greatly improved.

In the player, besides performance and synchronization improvements, adjustments for supporting live scenario were done. Releases included also enhanced controls and interaction support. The interactions and user referenced directive insertions were improved in order to provide a better user experience.

Since release 0.12 detailed logs for QoE analysis are available and since release 0.18 player is integrated with QoE module.

2.2.3. Release notes for Player

Table below contains change log for the Unity Player

Release v 0.2 2016-06-16
Functionalities <ul style="list-style-type: none"> - Works with GStreamer Unity Bridge 1.2 version - Support for Android Cardboard (Android only) - Support for HTC Vive (PC version only) - New contents json input interface - New content and device selection interface - Base XmlEngine Parser for media content - Base JSON Parser for json content file - Shape Types: Rectangle (directive content), SphereCaps (omnidirectional content) - Display modes: TV, Cardboard, Tablet, HTC Vive - On PC: Cardboard/tablet move around functionality (pressing ALT + moving the mouse) - Back functionality while displaying media - Exit functionality Known Bugs/Issues <ul style="list-style-type: none"> - The app crashes sometimes while displaying high resolution videos - Render issues with high resolution videos on android devices - Videos with audio have unexpected behaviour - On exiting application crashes (audio issues)
Release v 0.3 2016-07-04
Functionalities

- add: InteractiveObject script and Interactive base actions
- add: InputManager and basic mouse and touch managers
- add: callback actions functionality to Spheres and Rectangles (setVisibility, UnsetVisibility, PlayTransition)
- add: set InputManager on Hmd scene
- add: hmd input manager for hmd scene
- add: target selection by shapeld, add mediaObject to playTransition initialization
- add: media video extension constant
- add: mediaElement as communication layer between gubs
- add: ExternalAlphaColor shader to control fade in & fade out
- add: fade in functionality to the rendered mediaObjects
- add: media element initialization on callbacks
- add: set a marker point to hmd scene
- modify: set sphere radio with anchors distance
- modify: check visibility in a modified shape
- modify: camera background color to black
- modify: setup black color to texture when start
- modify: StreamManager, works without ApplicationManager
- modify: material using the new shader and prefabs using the matShaderColorMask material
- fix: remove MonoBehaviour from HmdInputManager
- fix: adding the delegates to a list and clear the list after delegates are removed
- fix: change state of the interactive object only if there is a callback
- fix: get the media element if it's not set yet

Known Bugs/Issues

- Inputs are not working properly on SamsungGear VR
- Render issues with high resolution videos on android devices
- Videos with audio have unexpected behaviour
- On exiting application crashes (audio issues)

Release v 0.4 2016-07-21

Functionalities

- add: base for user and world frame reference
- add: new crosshair marker for Samsung GearVR scene
- add: new input controller for Samsung GearVR device
- add: set Samsung Gear scenes with new input
- add: new tools:
 - OBJExport, script to export scene meshes to obj
 - CubeSphere generator scripts
- add: README file describing the scripts inside Tools folder
- add: new projection sphere
- add: new immersiaTv icon and splash-screen
- add: unsetVisibleAction script
- modify: SetVisible action to always be true and set the correct action on "unsetVisibility" callback parse
- modify: Use CubeSphere in sphereCaps prefabs, textures are smoother
- modify: transform the direction vector with transformMatrix and set the new position to the user frameReference

- modify: HmdInputManager to accept Samsung Gear input
- modify: move OBJExport to Tools folder
- fix: missing dll's for Oculus builds
- fix: check if the shape Id is the same as the target of the action
- fix: calculate the size of the rectangle being the max with the wide of the reference background sphere
- fix: the z position of the panel button
- fix: apply the new formula to get the output size of the rectangle shapes
- Fix: visualization size needs to be multiplied with a magic number
- fix: remove the white point of the screen for Samsung GearVR HMD scene
- fix: remove the unused buttons from the menu canvas panel
- fix: remove warning messages, replacing Application.loadLevelName and logging the exception message
- fix: Readme file for tools folder
- fix: sphere correct rotation for uv mapping
- fix: change rectangle shape's factor (magic number) to see as a baked video
- fix: visibility actions comments

Known Bugs/Issues

- Synchronization not working properly for offline and online content
- The App crashes when coming back from video reproduction
- Render issues with high resolution videos on android devices
- Videos with audio have unexpected behaviour
- On exiting application crashes (audio issues)

Release v 0.6 2016.08-25

Functionalities

- add: new synchronization with mediaTime inside the player
- add: new sync type var to switch between dvbcsswc, hack and no_sync in the application manager
- add: MVC for the Menu Manager.
- add: Load videos in local avoiding server download
- modify: save prefabs setup, broken when the project is cloned for the first time
- modify: changes in the scenes with the new mvcViewBase
- modify: update comments, use defaults local and remote json strings from ApplicationManager
- modify: remove spaces and commented lines
- modify: apply sync type on SessionManagerBridge and StreamManager scripts
- modify: set the content scene to use the Synchro Hack
- modify: remove unused isSamsungGear var, go directly to SamsungGearScene without showing the DeviceSelection panel
- modify: setup the starting angle for SamsunGear media menu by buttons length
- modify: update menu scenes, removing debugmanager gameobject
- modify: add features for non-overlap android applicaciotion in same device.
- modify: add new methods to check looping
- modify: access to the ContentTimeManager from the applicationManager
- modify: use of the MediaTime and MediaTimeInSeconds from the ContentTimeManager
- fix: recover device buttons

fix: make a pause&play to make it work on android the force-sync
 fix: checkers of vars that are not null
 fix: events using correct time for sync
 fix: we dont have to change the sync type if there is not previous content
 fix: remove default uri from StreamManager in tablet scene
 fix: documentation and sync part onStart event invocation
 fix: Control if is local and take the absolute path resources
 fix: Back button
 fix: update SessionManagerClient in submodule to recover the mediaTime
 fix: remove MenuManager as singleton

Known Bugs/Issues

- Live Streaming issues, doesn't reproduce
- Render issues with high resolution videos on android devices
- Videos with audio have unexpected behaviour
- On exiting application crashes (audio issues)

Release v 0.7 2016.11-21

Functionalities

add: a little minimap for tablet devices showing the position of the portals
 add: builded GUB with gstreamer-1.11 and playbin3
 add: new TransitionUpdaters with iTransitionUpdater as abstract class

update: integration with live streaming
 update: add finite state machine for stream logic management (for live updates)
 update: prebuild the scene to draw at the current event time
 update: move and set shape and anchor update methods to their classes (refactor)
 update: comment warning to do the checks before changing values on shapes (refactor)
 update: remove toggle visibility
 update: add TransitionModeTypes.NormalMode for shapes without a transition defined
 update: Parse Helper for TransitionMode adds NormalMode
 update: disable the collider to remove interaction after remove shape is doing
 update: fix the user reference shapes to the camera, so now they dont make a slerp movement and are fixed
 update: use public vars to activate the gstreamer debug mode in the editor

fix: live streaming starts from index 1
 fix: transition updater for image shader transitions and use it for unset transitions shapes
 fix: the videos for transitions working with the new methods
 fix: Separate the gstreamer loading from GstUnityBridgeTexture.

Known Bugs/Issues

- Touch controller not working properly on Tablet scene
- Pixel C tablet doesn't get the correct orientation while video reproduction

Release v 0.8 2016.12-20

Functionalities

add: Player icons for video controlling
 add: UI interactions for video controller
 add: the video player menu mvc base files
 add: Player control icons canvas prefab
 add: PlayerManager script to control video activity
 add: merge mouse and touch in the same input manager
 add: DVBCSS for synchronization
 add: session manager server into player client

update: add speed to scaleObjUpdater script
 update: setup indexes for rtsp and dash streaming
 update: minimap move to right-bottom corner
 update: relocate scripts and make static ButtonSetup method
 update: add buttons methods, create PlayerMenuManager as singleton, save into prefab
 update: add simple navigation among panels
 update: add player controller logic to control video flow without synchro
 update: save Tablet scene with the last changes of the video player controls
 update: set as default mouseTouch input manager
 update: remove the touch section and use only the mouse part.
 update: add new interactive object icon
 update: remove the user reference icons from the minimap
 update: remove the doubletap option in vrcamerarotation
 update: dont disconnect from sessionmanager when exiting the device scene
 update: remove the ref from methods
 update: use Discovery server on player
 update: gub.jar to the last version
 update: launch the sessionmanager with a coroutine and use the Discovery service
 update: add fixed loop time
 update: connect through Discovery Services

experimental: use sessionmanager server inside the player. Only on android tv's and PC

Known Bugs/Issues

- Android Players can't use discovery service to find the sessionmanager, multicast is not working properly on some devices
- Pixel C tablet doesn't get the correct orientation while video reproduction

Release v 0.9 2017.01-19

Fucntionalities

add: Doxygen configuration file for DoxyWinzard app
 add: new structure for live events
 add: xml dynamic update on live events

update: add NTP server time petition and StreamManager refactoring to use the NTP UTC time
 update: use utc time from xml file in live events

update: set new images for the release removing the xmas hat
 update: remove SessionManagerServer folder, the .dll will be include in SessionManager
 update: Connect directly to session manager hardcoding the ip, instead of using Discovery service

fix: exit from PreBuild loop if currentIndex is out of range
 fix: use the correct time parser to unix time, rename the script NTPTTimeManager to NetworkTimeManager.

Known Bugs/Issues

- Android Players can't use discovery service to find the sessionmanager, multicast is not working properly on some devices
- Pixel C tablet doesn't get the correct orientation while video reproduction
- Android devices are not connecting correctly to the session manager

Release v 0.10 2017.02-09

Functionalities

add: SteamVR 1.2.0
 add: update content on the devices when new one is selected
 add: set dynamic loop to SessionManagerServer through xml TV data

update: add a range of view for the minimap in Tablet scene
 update: connect through discovery service, wait 2 seconds for masters and 0.1 for clients.

fix: dont download the xml content twice
 fix: GetLocalIPv4 method only works on windows systems, use LocalIpAddress for Android
 fix: check if content xml is downloaded before entering select device screen.

Known Bugs/Issues

- Android Players can't use discovery service to find the sessionmanager, multicast is not working properly on some devices
- Pixel C tablet doesn't get the correct orientation while video reproduction
- Synchronization is not working properly, video losing synchro and doesnt loop
- Loading and DeviceSelection panels appears at the same time when coming back from the video

Release v 0.11 2017.03-03

Functionalities

add: more documentation

update: doxygen config file

fix: commenting the json validation mends the loop and the synchronization fault
 fix: check the current navigation panel to avoid showing DeviceSelection panel anytime
 fix: take the correct events from TV

Known Bugs/Issues

- Android Players can't use discovery service to find the sessionmanager, multicast is not working properly on some devices
- Pixel C tablet doesn't get the correct orientation while video reproduction
- SamsungGear VR clicks are not working on menu buttons
- Live videos checks milliseconds instead of seconds to launch events

Release v 0.12 2017.03-16

Functionalities

- add: logger module
- add: enabled logging to database
- add: log texture resolution and video address

- update: change the default sphere effect time to 7 seconds
- update: documentation and code relocation
- update: add the CHANGE SPHERE functionality, having only 2 sphere on transition and then only 1
- update: MediaObjectData into several components (Gub, Position, Transition, Interaction)
- update: [SessionManager] remove the mediatimecallback and use CurrentMediaTime property
- update: check by constant the console log showing, add comments and remove the Debug.Log of the Running MediaTime
- update: add Logs folder to the gitignore

- fix: minimap camera icon box collider remove
- fix: disable logs from GUB that cause crash
- fix: move and rotate the directive portals to the correct place
- fix: there were 2 InputManagers attached to StreamManager
- fix: save logs in PersistentDataPath in android devices
- fix: void memory leak with the database on android devices
- fix: take the correct events from TV
- fix: [SessionManager] solved conflict in logging config when server and client are together

Known Bugs/Issues

- Logger callbacks crashes the app in android
- clicking in multiple portals create multiple spheres
- Android Players can't use discovery service to find the sessionmanager, multicast is not working properly on some devices
- Pixel C tablet doesn't get the correct orientation while video reproduction
- SamsungGear VR clicks are not working on menu buttons
- Live videos checks milliseconds instead of seconds to launch events

Release v 0.13 2017.03-31

Functionalities

- add: (GUB) cache log messages to prevent crash
- add: (Logger) CameraLogs, modify: log levels setup
- add: (Logger) function to log messages with given timestamp
- add: (Logger) log tablet camera position

add: image rendering functionality on shapes
 add: new images for transitions
 add: MediaValueType field for the render type (video, image, audio)
 add: fields startClip and endClip for Transitions, from the xml, parse and update
 add: Quality field, as the relation of resolutions and bitrates

update: (SessionManager) discovery workaround using broadcast udp, instead of upnp based on multicast
 update: (logger) log camera fov and end of frame
 update: use different shader for Image Spheres
 update: recover onDeactivate field in the xml
 update: refactor internal MediaObject class and controllers.

fix: correct change sphere, avoid activating more than 2 spheres at the same time
 fix: Logger callbacks crashes the app in android
 fix: use the correct way to re-scale the images, avoiding orientation issues
 fix: Images in the spheres need a flip in x, removeshape modata missing
 fix: disable the correct meshCollider when removing a shape from scene
 fix: dont apply interaction on non interactive shapes
 fix: interaction marker shows and hides
 fix: the mesh Quad was missing in the Mesh Collider component

Known Bugs/Issues

- Multiples clicks on portals makes an undesired behaviour on them
- Pixel C tablet doesn't get the correct orientation while video reproduction
- SamsungGear VR clicks are not working on menu buttons
- Live videos checks milliseconds instead of seconds to launch events

Release v 0.14 2017.04-13

Functionalities

update: send GUB Pipeline decoding parameters by struct from unity player

fix: Transitions between spheres when a new one is loaded by event

Known Bugs/Issues

- Multiples clicks on portals makes an undesired behaviour on them
- Pixel C tablet doesn't get the correct orientation while video reproduction
- SamsungGear VR clicks are not working on menu buttons
- Live videos checks milliseconds instead of seconds to launch events
- black screen while loading new shapes

Release v 0.15 2017.04-28

Functionalities

MAYOR UPDATE: Change to Unity 5.6.0f3

add: new Google VR SDK plugin 1.30
 add: Oculus Utilities for Unity5, version 1.13.0 for SamsunGear VR

add: new event system prefab used for Gaze inputs
 add: Steroscopic cam script and new HMD scene with it
 add: new Tablet_vr scene for testing Google VR SDK on it
 add: sphere and directive transitions with TV sets
 add: new 3D environment for androidTV/SamsungGear VR menu scenes

update: change from bundleIdentifier to applicationIdentifier
 update: project version and change textureImporterType
 update: shader and scripts from GoogleVR (unity 5.6)
 update: remove prebuilt scenes and fix GvrViewerEditor script
 update: remove google vr sdk on Tablet scene and use VRCameraRotation with gyroscope
 update: improve gyroscope movement on tablets
 update: BuidMaker performs now the correct VR sdk selection
 update: add gaze cursor and correct transitions
 update: little workaround to avoid display issues on TV scene
 update: relocate the directive tv shape to 0.3 value
 update: Gyroscope optimization
 update: make transitions only when the gub is loaded and synchronized

fix: add UNITY_STANDALONE_LINUX flag in FileHelper.cs for linux builds, re-locate the menu buttons for SelectMediaHMD view
 fix: the render issue on TV scene in android devices.
 fix: material renderqueue number should be less than the TV material render queue
 fix: Pixel C and other tablets Natural orientation
 fix: Samsung Gear VR clicks

Known Bugs/Issues

- Multiples clicks on portals makes an undesired behaviour on them
- Live videos checks milliseconds instead of seconds to launch events
- black/white screen while loading new shapes
- there is no loop with the session manager
- delays connecting with the session manager
- low response in the session manager

Release v 0.16 2017.05-17

Functionalities

add: new image shader for transitions

update: base for camera vertical rotation
 update: changes in VRCamera
 update: raycast all elements and interact with the first one where its hit in a pixel without transparency
 update: change spheres priority order (less distance, more priority)
 update: change sprite animation controller and scenes
 update: Add TabletUIManager logic to StreamManager to control scene HUD items (loading icon and minimap)
 update: change default times for BasicTransitions to 0
 update: live content get unix time in seconds

update: add a threshold for the transitions effect in the ExternalAlphaGuide shader, making them smooth and less pixelated
 update: use the new Image shader on ImageMat material
 update: prevent creating the TV directive on tablet devices
 update: remove shape when finishing transition
 update: hide loading icon when resetting the basetime
 update: constants and add banner icon for android tv in project settings
 update: (SessionManagerClient) comment logger for unknown behaviour

fix: execute action logic and prevent states change if the action has not been executed
 fix: tv default distance
 fix: recover touches for buttons
 fix: get time from ntp server once and manage the time internally
 fix: avoid clipping effect on RemoveShape transitions
 fix: missing onDeactivate fields, duplicate from onActivate
 fix: missing sprite key on loading animation
 fix: Multiples clicks on portals, waits till the transition is done
 fix: (SessionManagerServer) remove mongodb address to avoid undesired behaviours
 - delays connecting with the session manager
 - low response in the session manager

Known Bugs/Issues

- black/white screen while loading new shapes
- GStreamer empty buffer causes application crash
- GazeInput doesnt change on HMD scenes in SamsungGearVR

Release v 0.17 2017.07-07

Functionalities

add: cubemap projection prefabs and test scene
 add: [tools] cubemap generator script, texture splitter script and starter
 add: VibrateManager and AndroidManifest with Vibrate permission
 add: Preview player setup and build maker
 add: ExternalListener script to manage external websocket calls
 add: Save application preferences base logic. Maintains the json content uri
 add: app settings json file and documentation
 add: simple VR Camera rotation controller without gyroscope, only touches
 add: new prefab using gub texture rendering for Unity UI system
 add: Oculus and HTC Vive builds
 add: Preview build

update: upgrade to SteamVR 1.2.2 version
 update: oculus and htc vive defines changes
 update: automatically go to hmd scene on Oculus and HTC Vive builds
 update: navigation system, back doesnt exit from the app on oculus devices
 update: go to content selection scene when pressing back, and enter the hmd scene if there is a content selected
 update: gub new attributes
 update: play/pause the media objects when the speed changes
 update: update the portals position taking in count the main sphericalCap (targetId)

update: remove Vertical scroll from VRCameraRotation script
 update: merge gyro and manual camera rotation in VRCameraRotation script
 update: integrate the external listener with the player
 update: GStreamer 1.11.2 for android in the player
 update: (GUB) last compiled libraries
 update: (SessionManagerClient) send the video reproduction speed when is changed

fix: Stop automatically updating the VR content from SteamVR. Disable the [InitializeOnLoad] on SteamVR_Settings to avoid conflicts with other plugins
 fix: set correct values for PichZoom script and add it to tablet scene
 fix: add eventsystem to detect correctly the touches on main menu buttons
 fix: solve crash targeting the active main media when there is no one
 fix: seeks on live streams dont work
 fix: remove console warnings and add logs messages
 fix: (GUB) add d3d11 graphic backend to make it work on windows
 fix: (GUB) docker build for android with newer gstreamer
 fix: (SessionManagerClient) dont change basetime while speed is 0

Known Bugs/Issues

- black/white screen while loading new shapes
- GStreamer empty buffer causes application crash
- GazeInput doesnt change on HMD scenes in SamsungGearVR
- Vertical scroll issues on Tablet scene
- RMTTP streams not working properly, crashes when no data arrives

Release v 0.18 2017.07-20

Functionalities

add: iTween plugin for movements and animations
 add: visual effects for interactions
 add: GUB test scenes for unityPlayer
 add: Loading and saving application preferences to json file immersiattvsettings.json
 add: (SessionManager) Interface definition for session manager with base implementation and tests
 add: (SessionManager) sending synchronization reports and TimelineSynchronization tests
 add: (SessionManager) video controller support
 add: (SessionManager) linux session manager
 add: (SessionManager) websocket wallclock
 add: (SessionManager) setting customInfo from client as user defined data (with USER prefix)
 add: (Qoe) Library for sending qoe feedback
 add: (Qoe) qoe integration in player

update: force unity binaries to text mode
 update: add enableGyroscope var to enable/disable it via script.
 update: remove gameobject and set enableGyroscope to false from tablet scene
 update: add BeginAction and EndAction events for interactions
 update: remove interactionMark from code
 update: add scale transform to visual feedback
 update: do not remove video shape of the interactive object while interacting
 update: default graphics api for win directx11

update: Remove JSONParser.cs, it is replaced by AppPreferencesJSON.cs, ContentMediaJSON.cs and JsonUtility

update: default values in constants and immersiatsvsettings.json file

update: (SessionManager) SessionManager and client implements ISessionState (with tests)

update: (SessionManager) controls in session-manager-server project

update: (SessionManager) adapt services names for the fixed in web-player

update: (SessionManager) move mongoDB configuration from session manager client to player

update: (Qoe) Add cameraLog to scenes

update: (Qoe) qoe disabled by default in Constants

update: (Qoe) move qoe start from GstUnityBridgeTexture to MediaObjectGUBController

fix: enable gyroscope on HMD scene

fix: Solve vertical scroll problem in Tablet

fix: remove the vr sdks on win64 standalone build in Buildmaker script

fix: add check to avoid infinite loop on DownloadManager

fix: (SessionManager) compilation issues on Ubuntu

fix: (SessionManager) SyncInfo for large time values

fix: (SessionManager) get user defined data

Known Bugs/Issues

- black/white screen while loading new shapes
- GStreamer empty buffer causes application crash
- GazeInput doesnt change on HMD scenes in SamsungGearVR
- RMTTP streams not working properly, crashes when no data arrives

Release v 0.19 2017.08-17

Functionalities

add: enabling multicast on android

add: use Anchor info (longitude and latitude) to re-orient the spheres

add: (SessionManager - Discovery) Multiple DiscoveryEngines (at the moment Broadcast & UPnP)

add: (SessionManager - Discovery) UPnP client/server working on all local interfaces

add: (SessionManager - Discovery) Broadcast client/server working on all local interfaces

add: (SessionManager - Discovery) ContentIdentificationService returns url's with proper server address if more if more local interfaces available.

add: (GUB) get framerate

add: (GUB) output dash segments address

add: (Qoe) setup qoe address in JSNO file

update: build_player_dependencies.py, build for submodules, gitignores

update: GUB reports synchronization time to session manager

update: (SessionManager) add,remove,rename device from discoveryClient

update: (SessionManager) save/load discovery results to file.

update: (SessionManager) example Client&Server non-blocking wait.

update: (SessionManager) UPnP binaries with modified Multicast binding (Android/Windows)

update: (Mono-Upnp) fixed multicast binding.

update: (Mono-UPnP) STA fix

update: (SessionManager - UPnP Discovery + Logging) Compliance with DVB-CSS - Proper XML with CSS-CII URL instead of simple URL passing.

update: (SessionManager - UPnP Discovery + Logging) log4net enabled for UnitTesting (ConsoleOutput)

update: (SessionManager - UPnP Discovery + Logging) Console.WriteLine replaced with log.Debug ...

update: (SessionManager-Discovery) code cleanup, comments on interfaces.

update: (SessionManager-Discovery) SessionManagerClientService is now able to start&stop multiple times during one lifetime.

update: (SessionManager-Discovery) DiscoveryServerService can start stop multiple times during lifetime.

update: (SessionManager-Discovery) removed dependency on ipv4Mask property, DiscoveryServerBroadcast separated for each interface.

update: (GUB) synchronization time getter

update: (GUB) pause and play keeping sync

update: (GUB) modify API to support speed and sync changes

update: (GUB) adjust sync for pipeline latency

update: (GUB) do not seek on small sync adjustments + code cleanup

update: (QoE) API changed to reflect actual usage scenario. Some additional parameters now reported.

API is tailored to report new parameters as they become available, not only after the vertical retrace was made.

New parameters can now be reported: frame resolution changes, roll parameter of the viewing device, actual stream fps.

update: (QoE) Stream of JSON messages now adhering to RFC 7464:

The stream of JSON messages is now delimited by a 'Record Separator' start flag and a 'Line Feed' stop flag.

update: (QoE) Reporting thread should now be closing the TCP socket correctly.

update: (QoE) API changed to reflect support of single session at once. Content ID added.

fix: remove the flick effect when removing the shape

fix: the fade in/out effect last the correct duration time

fix: the highest priority sphere or shape from the disabled pool shows up when the current main video is removed

fix: report sync in correct moment

fix: (SessionManager) resetting content on last client disconnected

fix: (SessionManager) change __server__ to r_server__ in Uri to work with Mono

fix: (JSONValidation) unable to create validator if run from unity.

fix: (QoE) setup fps, when initially unknown

Known Bugs/Issues

- black/white screen while loading new shapes
- GStreamer empty buffer causes application crash
- GazeInput doesnt change on HMD scenes in SamsungGearVR
- RMTP streams not working properly, crashes when no data arrives

2.2.4. Release notes of Web Player

Release web-0.21

Functionalities:

- * Use Three.js Library to compose the scenes
- * Use Dash.js Library for reproducing MPEG-DASH video streaming
- * Support for Chrome and Mozilla browsers
- * Works over HTTP and HTTPS
- * Available display modes:
 - * TV
 - * HMD
 - * Tablet
- * Media types supported:
 - * Video
 - * Images
- * Content types supported:
 - * Directive or traditional content
 - * Spherical projection
 - * Cube map projection
- * On HMD mode, the device orientation and touch controller is used to move the scenes
- * On Tablet mode, only touch controller is used to move the scenes
- * Synchronization with ImmersiaTV "session manager" using secure websockets
- * Interactions are supported
- * Transitions are supported
- * Ambisonics audio supported

Known bugs

- * Discovery service is not available, browsers are not compatible with UDP datagrams

2.2.5. Release notes of Off-line Production Tools

The table below contains change log for the Off-Line Production Tools (Premiere Pro plugin)

Release 0.2 2016-06-16

Functionalities

- * Product ready for first tests

Know bugs

- * the preview is not correct in size
- * the video format is not named correctly (equirectangular, not spherical)
- * The panel in this version has to be run in debug version, the installation notes contained special guide how to enable this feature

release 0.3 2016-07-4

Functionalities

- * Add interactive transitions effects and keyframes for Premiere Pro Panel
- * Visible: Whether this shape is visible in the scene or not. Hidden shapes can be activate through interaction.
- * Luma matte: Optional mask to use on the texture media. This mask is also a file name (see mediaFile), so it can either be a video or a still image. It uses a "secondary timeline", independent from the mediaFile, since it can be controlled by the application (it can be shorter and loop, or triggered by user actions). It must be a gray image or video. This allows, for example, to have circles with blurry edges shown in the scene, using only rectangular shapes, or transitions.
- * Overlay: A second texture media (with companion alpha mask in overlayMaskFile) which is rendered on top of the mediaFile. It uses the "secondary timeline" (see maskFile). It can be used to make a "beauty mask", embellishing transitions.
- * Overlay luma matte: Alpha mask for the overlayMediaFile.
- * OnActivate callback: Player method to call when the shape has been "activated". The exact action that triggers the activation depends on the device and the player (could be a "touch" action on a tablet, and "look for more than 3 seconds" on an HMD). The list of available methods is outside the scope of this metadata description.
- * OnDeactivate callback: Player method to call when the shape has been "deactivated". The exact action that triggers the deactivation depends on the device and the player (could be "stop looking" on an HMD). The list of available methods is outside the scope of this metadata description.
- * Separate switch:
- * Switch latitude: Degrees above or below the horizontal plane.
- * Switch longitude: Degrees on the horizontal plane.
- * Switch size: Relative size of the shape.
- * Switch appearance: Texture media (color) to display on this Shape.
- * Switch reference: Reference frame for the Anchor.
- * Modify visual interface for Premiere Pro Panel
- * Delete the output path box, now it appears when export button is clicked.
- * Export button is blocked if no render or metadata are checked.
- * Modify the refresh button and change his position over the panel.
- * Add information box into the top right of the panel

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.

release 0.5 2016-08-17

Functionalities

- * Reorder the effect panel.
- * Delete the "luma matte" checkbox from Premiere Pro Panel.
- * Renamed portal effect controls:
- * OnStart Callback --> OnStart
- * OnDeactivate Callback --> OnDeactivate

- * OnActivate Callback --> OnActivate
- * Separate Switch --> Extra Portal
- * Switch Vertical position --> E.P. Vertical degrees
- * Switch Horizontal position --> E.P. Horizontal degrees
- * Switch Size --> E.P. Size
- * Switch Appearance --> E.P. Appearance
- * Switch reference --> E.P. Reference
- * Rendering with new transformation function
- * Portal effect size in range 0 to 100%
- * Visual size control changed on portal effect panel

Bugs fixed

- * In extra portals, the MediaFile name don't show the number of track.
- * The distance of spherical shapes is between 0.8 to 1.
- * Ommnidirectional videos will not exported.
- * Export different tracks in Premiere Pro when there's no interactivity
- * Now don't generate xml file if no folder is selected.

Known bugs

- * If a project is closed and another is opend, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.

release 0.6 2016-08-25

Functionalities

- * Add new preset for 16:9 videos.
- * Fix support over 10 tracks for type selector
- * Fix problem with transitions that it will export black in 360 videos.
- * Fix bug that duplicate the distance parameter in XML file.
- * (0.6.1) Changed the name of tracks exported in XML file, now it will have and extension "sp_" for spereical tracks, "re_" for rectangular tracks and "tv_" extension for TV tracks.
- * (0.6.1) The number of tracks in XML displayed on premiere pro panel
- * (0.6.6) Add new presets for Portals and TV

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content

release 0.8 2016-01-12

Functionalities

- * Premiere pro panel with after effects integration, interactive transitions, sync
- * Modify presets with new outputs resolutions

- * Add new transition list
- * FadeIn/FadeOut
- * IrisOpen
- * IrisClose
- * SlideLeft
- * SlideRight
- * WipeLeft
- * WipeRight
- * Modify Portal effect to RectangularPortal and SpkericalPortal
- * Modify onActivate to onActionStart
- * Modify onActivate to onActionStop

Known bugs

- * If a project is closed and another is open, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.

release 0.9

Functionalities

- * Add new option to select at what spherical resolution can be exported media files:
 - * 1024x512
 - * 2048x1024
 - * 4096x2048
- * Add new option to select at what directive resolution can be exported media files:
 - * 1920x1080
 - * 2560x1440
 - * 3840x2160

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.

release 0.9.1

Functionalities

- * Fixed bug: The portals were displayed in red in Premiere Pro Panel

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.

* All video inputs must have the same resolution to render the content.

release 0.10.0

Functionalities

- * Modify: select device type button in Premiere Pro Panel
- * Add: export bitrate selectable option in Premiere Pro Panel
- * low
- * high
- * Add: new presets
- * Fixed bug: error when one clip ends at same time that another starts in the same track

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.
- * Lumetri effects are incompatible with ImmersiaTV presets.

release 0.11.0

Functionalities

- * Fixed bug: don't export correctly all 360 tracks in XML.
- * Fixed bug: id of track selected in onActionStart.

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.
- * Lumetri effects are incompatible with ImmersiaTV presets.

release 0.12.0

Functionalities

- * Modify: XML file name, now only accept alphanumeric characters.
- * Fixed bug: The names of 360 tracks are incorrect.

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.
- * Lumetri effects are incompatible with ImmersiaTV presets.

release 0.13.0

Functionalities

- * Modify: default portal size to 20%
- * Modify: rename transitions effects
- * Modify: now export all tracks individually (no fusion tracks)
- * Add: preset to export PNG images
- * Add: new transitions, wipe open and wipe close
- * Delete: fusion tracks function
- * Delete: lowest export quality of spherical tracks

TODO

- * Add support to put multiple images into a same track.

Known bugs

- * Preview player is rotated 90 degrees to Premiere preview.
- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.
- * Lumetri effects are incompatible with ImmersiaTV presets.

release 0.14.0

Functionalities

- * Modify: Changes in ppro panel preview mode
- * Fixed bug: in playerWeb (orientation and portal position)
- * Fixed bug: in Ppro panel preview module with user reference portals
- * Modify: shader names.
- * Modify: now can export multiple images in a same track
- * Add: New quality option (default), export video at same quality that original
- * Fixed bug: warnings for MacOs compilation

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.
- * Lumetri effects are incompatible with ImmersiaTV presets.

release 0.15.0

Functionalities

- * Add: Presets to PPRO plugin to determinate the portals position
- * upperleft

- * uppright
- * lowerleft
- * lowerright

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.
- * Lumetri effects are incompatible with ImmersiaTV presets.

release 0.16.0

Functionalities

- * Modify: Refactor of preview player
- * Add: button to auto select all tracks to a specific device.
- * Fixed bug: preview mode in mac don't run correctly

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.
- * Lumetri effects are incompatible with ImmersiaTV presets.

release 0.17.0

Functionalities

- * Modify: Little corrections with filenames generation
- * Modify: Export current preview as Thumbnail.png on media export
- * Fixed bug: when network has not internet connection
- * Fixed bug: uncaught exception when run app.encoder.encodeFile()

Known bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.
- * All video inputs must have the same resolution to render the content.
- * Lumetri effects are incompatible with ImmersiaTV presets.

release 0.19.0

Functionalities

- * Modify: RectangularPortal -> Portal
- * Modify: UserReference ref. portals % to the corner.

- * Add: 360 projection Mapping effect for omnidirectional videos.
- * Add: Anchor in sphericalCap shapes.
- * Add: Support to cubemap projections.

Known bugs

- * All video inputs must have the same resolution to render the content.
- * Lumetri effects are incompatible with ImmersiaTV presets.

Fixed bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.

release 0.20.0

Functionalities

- * Modify: Preview position of user reference portals
- * Add: version to XML file.

Known bugs

- * All video inputs must have the same resolution to render the content.
- * Lumetri effects are incompatible with ImmersiaTV presets.

Fixed bugs

- * If a project is closed and another is opened, Premiere does not update the QE and preserve the old tracks.
- * When using the QE to get the number of tracks (getActiveSequence()), it stops working if a track is removed from the list.

2.2.6. Release notes of Live Production Tools

Live Production Tools 1.0

Functionalities

- Initial release with embedded modules for handling stream processing related tasks (sync, transform, etc);
- Basic UI according to the design discussed with ImmersiaTV partners.

Live Production Tools 1.1

Functionalities

- Cinegy Transport is introduced as separate component to allow distribution of stream transcoding tasks.

Live Production Tools 1.2

Functionalities

<ul style="list-style-type: none"> - Demo integration with players, image/streams processing improvements; - Bugfixing.
Live Production Tools 2.5.0.53
<p>Functionalities</p> <ul style="list-style-type: none"> - Re-worked UI with advanced accessibility; - Compatibility with latest ImmersiaTV scene description XML format; - Re-worked Cinegy Transport pipeline; - Cinegy Transport WEB API; - Bugfixing.
Live Production Tools 2.6.14.59
<p>Functionalities</p> <ul style="list-style-type: none"> - Advanced Windows OS installer GUI; - Improved stream sync processing; - Fixed preview of the live stream in Live VR interface; - Configuring Cinegy Transport live stream processing pipeline from Live VR; - Bugfixing.
Live Production Tools 2.6.16.62
<p>Functionalities</p> <ul style="list-style-type: none"> - Scaling preview stream on Cinegy Transport side to reduce CPU requirements on Cinegy Live VR for streams decoding; - Bugfixing.
Live Production Tools 2.6.19.65
<p>Functionalities</p> <ul style="list-style-type: none"> - Improved support for VideoStitch Vahana VR generated streams with variable frame rate; - Improved global streams delay logic in order for the player to have time detecting the scene changes; - Bugfixing.
Live Production Tools 2.6.21.69
<p>Functionalities</p> <ul style="list-style-type: none"> - Enabled ImmersiaTV scene XML generation for several device types simultaneously; - Extended logging on Cinegy Transport side for stream processing progress; - Bugfixing
Live Production Tools 2.6.31.80
<p>Functionalities</p> <ul style="list-style-type: none"> - Extended controls for stream processing to workaround network performance issues (input/output network adapters, etc); - Improved active source change by operator command; - Improved portals source change by operator command logic; - Bugfixing
Live Production Tools 2.6.32.81
<p>Functionalities</p> <ul style="list-style-type: none"> - Extended logging of the live stream processing;

- Improved broken/unstable streams processing;
- Streams transcoding optimization;
- Prepared for the stream processing pipeline optimization to reduce network stability dependency when processing live streams;
- Bugfixing

2.2.7. Release notes for Server

Release v 0.1 2016-03-07

Functionalities

- Generate Dash content according the Premiere.xml file

Known Bugs/Issues

- The name of the Premiere.xml only accept alphanumeric characters

Release 12 2016-03-17

Functionalities

- Generate Dash content according the Premiere.xml file
- Generation of multiple portal and spherical video
- Add and Remove representations

Known Bugs/Issues

- The name of the Premiere.xml only accept alphanumeric characters
- Incorrect indexation of the content in the main.json

Release 13 2016-03-31

Functionalities

- Generate Dash content according the Premiere.xml file
- Add and Remove representations
- Add quality property for every representation

Todo

- Add image handler
- Refactor
- Implement logger
- Add Control points in the generation content

Known Bugs/Issues

- The name of the Premiere.xml only accept alphanumeric characters
- Incorrect indexation of the content in the main.json

Release 14 2016-04-12

Functionalities

- Generate Dash content according the Premiere.xml file
- Add and Remove representations
- Add quality property for every representation
- Support for images in Premiere XML file
- Progress information in the ffmpeg process

- Check correct generation of the resolutions

Todo

- Add directly dash premier content
- Correct configuration of the Ffmpeg conversion in h264
- Refactor (Class refactor)
- Doxygen documentation (complete documentation)
- Premiere Xml validator (extract and check the info of the XML only for debug)
- Implement a Debug mode
- Direct conversion of the premier output media content without reenconding
- Refactor

Known Bugs/Issues

- The name of the Premiere.xml only accept alphanumeric characters
- Incorrect indexation of the content in the main.json
- Dynamic generation of the video codec type value in the xml

Release 15 2016-04-28

Functionalities

- Generate Dash content according the Premiere.xml file
- Add and Remove representations
- Add quality property for every representation
- Support for images in Premiere XML file
- Progress information in the ffmpeg process
- Check correct generation of the resolutions

Todo

- Add directly dash premier content
- Correct configuration of the Ffmpeg conversion in h264
- Refactor (Class refactor)
- Doxygen documentation (complete documentation)
- Premiere Xml validator (extract and check the info of the XML only for debug)
- Implement a Debug mode
- Direct conversion of the premier output media content without reenconding
- Refactor
- Jenkins debug options

Known Bugs/Issues

- The name of the Premiere.xml only accept alphanumeric characters
- Incorrect indexation of the content in the main.json
- Dynamic generation of the video codec type value in the xml

Release 16 2016-05-17

Functionalities

- Generate Dash content according the Premiere.xml file
- Add and Remove representations
- Add quality property for every representation
- Support for images in Premiere XML file
- Progress information in the ffmpeg process
- Check correct generation of the resolutions
- Autodetect profile configuration for the Ffmpeg conversion in h264

Todo

- Implement an error manager
- Unify Json names
- Normalize name pats and folders
- Refactor (Class refactor)
- Doxygen documentation (complete documentation)
- Premiere Xml validator (extract and check the info of the XML only for debug)
- Implement a Debug mode
- Jenkins debug options

Known Bugs/Issues

- The name of the Premiere.xml only accept alphanumeric characters
- Incorrect indexation of the content in the main.json
- Dynamic generation of the video codec type value in the xml

Release 17 2017-07-07

Functionalities

- Generate Dash content according the Premiere.xml file
- Add and Remove representations
- Add quality property for every representation
- Support for images in Premiere XML file
- Progress information in the ffmpeg process
- Check correct generation of the resolutions
- Autodetect profile configuration for the Ffmpeg conversion in h264

Todo

- Implement an error manager
- Unify Json names
- Normalize name pats and folders
- Refactor (Class refactor)
- Doxygen documentation (complete documentation)

Known Bugs/Issues

- The name of the Premiere.xml only accept alphanumeric characters
- Incorrect indexation of the content in the main.json
- Dynamic generation of the video codec type value in the xml

Release 18 2017-07-21

Functionalities

- Generate Dash content according the Premiere.xml file
- Add and Remove representations
- Add quality property for every representation
- Support for images in Premiere XML file
- Progress information in the ffmpeg process
- Check correct generation of the resolutions
- Autodetect profile configuration for the Ffmpeg conversion in h264
- Autodetect profile configuration for the Ffmpeg conversion in h265
- MultiBitrate generation for the same resolution

Todo

- Transformation Equirectangular -> Cubemap projection
- Tiler for Mpeg-Dash
- Implement an error manager
- Unify Json names
- Refactor (Class refactor)
- Doxygen documentation (complete documentation)

Known Bugs/Issues

- The name of the Premiere.xml only accept alphanumeric characters
- Incorrect indexation of the content in the main.json
- Dynamic generation of the video codec type value in the xml

Release 19 2017-08-17

Functionalities

- Generate Dash content according the Premiere.xml file
- Add and Remove representations
- Add quality property for every representation
- Support for images in Premiere XML file
- Progress information in the ffmpeg process
- Check correct generation of the resolutions
- Autodetect profile configuration for the Ffmpeg conversion in h264
- Autodetect profile configuration for the Ffmpeg conversion in h265
- MultiBitrate generation for the same resolution

Todo

- Transformation Equirectangular -> Cubemap projection
- Tiler for Mpeg-Dash
- Implement an error manager
- Unify Json names
- Refactor (Class refactor)
- Doxygen documentation (complete documentation)

Known Bugs/Issues

- The name of the Premiere.xml only accept alphanumeric characters
- Incorrect indexation of the content in the main.json
- Dynamic generation of the video codec type value in the xml

2.2.8. Release notes of VideoStitch Studio

As VideoStitch Studio is a commercial software following its own release cycles, we provide here the full release notes, highlighting **in bold face** the changes relative to ImmersiaTV.

VideoStitch Studio v2.2.0 – 2016-11-21

VideoStitch Studio 2.2.0 introduces **rig presets for an easier and better calibration**, GTX-10 series support, h265 input codec and Prores codec (both input and output).

Features

- **Rig presets to get more accurate calibration results**
- Support GTX-10 series graphic cards
- h265 inputs and Prores (input/output) support
- **Audio and motion synchronization redesign (more robust and deterministic)**
- Feathering (= better merging): blend_radius parameter control, masks support and overall improved algorithm
- Output audio resampler (sample depth and rate)
- **I can update calibration while preserving orientation**

Bug fixes

- "Seek frame"/scrubbing Studio feature is really slow / hangs
- Crash when low on GPU memory
- "Lock ratio" and output size accept odd values
- Extract source images that do not hold the mask
- Black seems on Multiband merger
- Video playback breaks at t=1s on GoPro metadatas
- Automatic and external calibration crash GPU drivers (2GPU)
- Masks used only on GPU 0 (2GPU)
- Black stitch preview screen (2GPU)
- Crash when importing an unsupported output
- Output bitrate is ignored
- Timeline gets huge values for still images
- Unable to export audio with Studio
- Video track is cut when exporting with audio
- Exporting project with audio fails, only part of the video is playing in VLC
- NTSC input framerate (like 47.952) cannot be processed with audio
- Output video length is not the one expected and contains black frames
- Launching twice batchstitcher creates zombie
- If I drag and drop a video with NTSC framerate to Studio icon, it crashes or does nothing
- Input does not appear in the output, leaving empty black zones
- Huge OS X performance regression
- Activation error pop up is not displayed for wrong license
- When launching audio syncs on videos without audio, Studio proceeds... and succeed?
- Crash when you change the output size (out of memory)
- Studio crashes when opening project with big pano (out of memory)
- Crash on motion synchronization (out of memory)
- Using inputs of unusual image dimensions crashes (out of memory)
- Studio unusable after project initialization fails (out of memory)
- The warning pop up doesn't appear when the machine is in lack of memory (out of memory)
- Output image changes size when window dragged to monitor with different dpi, video widget render size (Mac)
- Import twice a 8K image will make studio freeze (out of memory)
- Crash of studio when I import a template with a project that has 10 jpg inputs (out of memory)
- Algorithm curves ignored when exporting sequence not starting at 0
- Create a new project after the video is played make studio crash

- Studio output projection is forced to equirectangular when using interactive tab
- Studio crashes when there is no up-to-date CUDA on Mac
- Crash when opening small images
- If I try to open a still image inside a video project, Studio crashes
- Calibration crash with variable framerate input videos
- Studio crashes when removing or replacing input if copy audio is set
- Photometric calibration removes geometric calibration
- Studio crashes when deactivating "Show input numbers" while playing
- Stitching aborts with error after libav reader error
- Calibration crashes when FOV set to 0
- Edit orientation not working correctly for Hi-DPI displays
- A missing GPU is detected
- Drag & drop does nothing on OS X
- Disable all inputs crashes
- Backward compatibility issue with PTGui viewpoint correction
- If the audio codec is not supported, the video cannot be opened
- Launching calibration immediately after a cancelled calibration crashes
- No error Message when the user launch a Flash synchronization with one video
- Can't find the maximum display size and display 4 x 2
- Photometry calibration on images doesn't work/crashes
- Timeline can be undocked... and not recovered
- A video file is created when you cancel the process before the end.
- Studio can't be used after an input failure
- Remove the video parameters from the process tab when I have only images as inputs
- The user can use the stabilization widget when he use image in studio
- Studio crash when the user launch a synchronization with images
- Can't export a panorama with image on the inputs
- Studio crashes when the user open a project when a project is already opened.
- Studio crashes when you drag and drop a project or a media file while a process is running
- The stitched size is updated only after you reopen your project
- Studio memory leak
- Studio loading bar stuck at 50% if ptv is invalid
- We set a maximum bitrate for the audio instead of the bitrate
- Using images and videos in Studio: display/hide features and algos
- List of the audio sources is not updated when an input is removed
- Studio crashes when the user import an unsupported file after a supported file

Known bugs

- License file can be saved in the wrong directory => fixed in 2.2.1
- If the ptv contains a special character, the process operation crashes => fixed in 2.2.1
- Crash when opening an old pttv with special characters in path => fixed in 2.2.1
- Frame offset by 1 after seeking, go to previous frame not working => fixed in 2.2.1
- Jump frame does nothing when using Enter shortcut => fixed in 2.2.1
- PTGui template geometry ruined when imported in studio after a calibration => fixed in 2.2.1
- * Resetting a task in the batch stitcher does not reset the progress bar to 0 => fixed in 2.2.1

VideoStitch Studio v2.2.1 – 2016-12-22

Features

- Sign Studio v2.2 as a Mac application
- Link error messages with custom forms for a better support

Bug fixes

- If the ptv contains a special character, the process operation crashes
- Crash when opening an old ptvb with special characters in path
- Frame offset by 1 after seeking, go to previous frame not working
- Jump frame does nothing when using Enter shortcut
- PTgui template geometry ruined when imported in studio after a calibration
- The calibration of a calibrated camera rig is not preserved when you replace "circular fisheye" by "rectilinear"
- Project state "unsaved changes" after initial project save
- Resetting a task in the batch stitcher does not reset the progress bar to 0
- License file is not generated in the correct directory

VideoStitch Studio v2.3.0 – 2017-05-17

VideoStitch Studio 2.3 is now compatible with not only NVidia but also AMD and Intel graphics cards. For Mac users, it means most Apple computers released in 2013 and later are supported. Intel NUC is also among the supported devices.

Features

- **AMD and Intel graphics support** (unified executable)

Improvements

- **Enhanced calibration workflow** (crop and presets in the same view)
- Need a mention "Untitled project" when a media file is imported and the project is not saved
- Warning missing when audio codec is not supported
- **Open MPEG4 files on Mac**

Bug fixes

- Crash when trying to open a corrupted .jpg file
- Nothing happens when I open an image that is of different size than what "width" and "height" in the input configuration specifies
- Repairing input path fails when path contains forward and backward slashes
- Activation fails due to missing administrator rights
- Orientation button is missing because of an invalid license
- Get the pop-up activation successful with an outdated license
- Batch stitcher crashes when HDD runs out of space on OS X
- Process an output into another drive (not main) does not check size availability.
- Wrong values when importing a calibration from PTGui (depending on the locale, comma vs dot confusion)
- Studio allows project changes while in orientation adjustment mode
- OpenCL build failure (rBObzLAW program) on Intel HD Graphics 530 & 550
- Studio hangs in the middle of an export or stays in the task manager and can't be deleted (Intel HD Graphics 4600)
- "Show input numbers" button overlaps Blending settings on MacOS Sierra
- Jump frame dialog adds 0

2.2.9. Release notes of Vahana VR

As Vahana VR is a commercial software following its own release cycles, we provide here the full release notes, highlighting **in bold face** the changes relative to ImmersiaTV.

Vahana VR v1.1.0 – 2016-08-01

In this new Vahana VR release, the external audio pipeline and RTMP output features have been re-written from scratch. Moreover, you will have several new ways of capturing input videos (Magewell Pro acquisition cards and network streams) as well as more output options. After YouTube live streaming announcement, Vahana VR now provides a simplified workflow.

Features:

- I can stream to YouTube using Vahana VR UI
- I can use an external microphone with Vahana VR
- Improved RTMP output with authentication and server reconnection
- Support Magewell Pro acquisition cards
- Support HTC Vive as a preview device
- Support network stream inputs
- Support Decklink interlaced inputs
- Support Decklink card input audio (in SDI and with XLR adapter)
- Support Akamai CDN
- Use time-stamps to synchronize audio and video
- ProRes video compression format support in the HDD output (available with mov container)

Improvements:

- Fit the preview to the screen to improve performances
- I can use an external microphone named with Unicode characters
- I want to choose which GPU Vahana VR is using
- Set accurate default output values (resolution, fps, bitrate mode, key frame, audio configuration...)
- Displayed time, target fps and stitching fps now based on the same clock (the stitcher's target fps)
- "Save to HDD" and "RTMP" Uis redesign
- Acquisition cards in/out UI redesign
- Add the RTMP connection status to the log
- Vahana VR stream user agent is now recognizable
- Vahana VR stability on project open/close
- Crash reporter comments are 41ancelli
- Support Oculus CV1

Bug Fixes:

- Crash when clicking on Snapshot in output tab when audio is enabled and RTMP activated

- Decklink capture doesn't work when the same card model is used twice
- Edit the panorama during a live streaming will break the interface and stop the stream
- The user can change the parameters of an output that is currently used
- Output settings are not updated when you change the input settings
- Using inputs of unusual image dimensions leads to a crash
- Can't reconnect to the RTMP output after a disconnection
- Vahana VR crash when I close a project with an activated output
- Interface is broken when you restart one of your two screens
- Vahana VR crash when you import twice the same template
- When saving to HDD the output is cut at the end
- The user can't record 2 videos consecutively with a different video format
- Downsample/Upsample the RTMP streams breaks it
- The user can delete an output that is currently used
- Vahana VR crash when the user choose the pixel format "BGRU" on an output Decklink
- Crash when you change panorama size
- Unsupported pixel format (like RGB for Magewell Quad inputs) make Vahana VR crash
- Time display is cut when project is open for more than 10h
- Don't create by default RTMP log file as it can fill up the disk (1GB after 16h)
- Changing RTMP output configuration leads to crash
- Changing the output file then saving to disk without reloading the project leads to a crash
- Setting a custom vah.fps results in erroneous framerate
- Changing display mode in Magewell does not update the stitcher's FPS
- Masked pixels shouldn't be used
- Import a PTGui project with crop produces an erroneous FOV
- Opening videos from HDD as inputs doesn't work just after installing Vahana VR
- Inputs used in calibration contains the watermark when no license is activated
- Crash after SDI was disabled or when changing SDI settings
- Vahana VR will crash when you don't update the Oculus Runtime
- If I remove my external microphone, I cannot open my configuration
- Lens type resets to fullframe fisheye after calibration
- Vahana VR crashes if the input SDI signal doesn't match the one specified in the vah file
- The Field of View value set in the crop window is not used
- "Control points" feature is misleading and doesn't bring value
- GOP parameter is not working as expected

Known issues:

- Nvidia GTX1080 is not supported -> fixed in v1.1.1
- Decklink Duo2 and Quad2 are not supported -> fixed in v1.1.2
- Memory leak when streaming in CBR mode if the bandwidth is not wide enough -> fixed in v1.1.3
- HDD output files have compression issues when using SDI inputs
- Oculus Rift output instability on disable/close
- Continuous exposure compensation accumulates ev correction errors
- Calibration crashes when the Field of View value is set to 0
- Stitching time negative on stream inputs
- Crash when you close Vahana VR during a calibration or the loading of the inputs

- Cancel update leads to a crash
- If you unplug the screen on which Vahana VR is running, the app layout is broken

Vahana VR v1.1.1 – 2016-08-22

Following the release of the v1.1, this patch release adds support to GeForce GTX 10-Series GPUs.

Features:

- GeForce GTX 10-Series GPUs support

Known issues:

- Decklink Duo2 and Quad2 are not supported -> fixed in v1.1.2
- Memory leak when streaming in CBR mode if the bandwidth is not wide enough -> fixed in v1.1.3
- HDD output files have compression issues when using SDI inputs
- Oculus Rift output instability on disable/close
- Continuous exposure compensation accumulates ev correction errors
- Calibration crashes when the Field of View value is set to 0
- Stitching time negative on stream inputs
- Crash when you close Vahana VR during a calibration or the loading of the inputs
- Cancel update leads to a crash
- If you unplug the screen on which Vahana VR is running, the app layout is broken

Vahana VR v1.1.2 – 2016-08-30

Following the release of the v1.1.1, this patch release adds support to Decklink Duo 2 and Quad 2, and fixes some Decklink naming issues.

Features:

- DeckLink Quad 2 capture cards support

Bug Fixes:

- Decklink Quad 2 and Duo 2 cannot be used at the same time
- Distinguish two Blackmagic Pro cards

Known issues:

- Memory leak when streaming in CBR mode if the bandwidth is not wide enough -> fixed in v1.1.3
- HDD output files have compression issues when using SDI inputs
- Oculus Rift output instability on disable/close
- Continuous exposure compensation accumulates ev correction errors
- Calibration crashes when the Field of View value is set to 0
- Stitching time negative on stream inputs
- Crash when you close Vahana VR during a calibration or the loading of the inputs
- Cancel update leads to a crash
- If you unplug the screen on which Vahana VR is running, the app layout is broken

Vahana VR v1.1.3 – 2016-11-14

Following the release of the v1.1.2, this patch release fixes a memory leak, occurring when streaming in RTMP on a network with a too small bandwidth (network congestion).

Bug Fixes:

- Memory leak when streaming in RTMP if the bandwidth is not wide enough

Known issues:

- HDD output files have compression issues when using SDI inputs
- Oculus Rift output instability on disable/close
- Continuous exposure compensation accumulates ev correction errors
- Calibration crashes when the Field of View value is set to 0
- Stitching time negative on stream inputs
- Crash when you close Vahana VR during a calibration or the loading of the inputs
- Cancel update leads to a crash
- If you unplug the screen on which Vahana VR is running, the app layout is broken

Vahana VR v1.1.4 – 2017-02-14

Following the release of the v1.1.3, this patch release fixes a crash when 44ancelling an update. YouTube feature has been disabled, but you can still output to YouTube using the RTMP feature.

Bug Fixes:

- Cancel update leads to a crash

Known issues:

- HDD output files have compression issues when using SDI inputs
- Oculus Rift output instability on disabling/closing
- Continuous exposure compensation accumulates ev correction errors
- Calibration crashes when the Field of View value is set to 0
- Stitching time negative on stream inputs
- Crash when you close Vahana VR during a calibration or the loading of the inputs
- If you unplug the screen on which Vahana VR is running, the app layout is broken

Vahana VR v1.1.5 – 2017-03-30

Following the release of the v1.1.4, this patch release fixes instabilities due to outdated Nvidia drivers.

Improvement:

- Detect outdated drivers

Known issues:

- HDD output files have compression issues when using SDI inputs
- Oculus Rift output instability on disabling/closing
- Continuous exposure compensation accumulates ev correction errors
- Calibration crashes when the Field of View value is set to 0
- Stitching time negative on stream inputs
- Crash when you close Vahana VR during a calibration or the loading of the inputs
- If you unplug the screen on which Vahana VR is running, the app layout is broken

Vahana VR ImmersiaTV Unofficial Release – 2017-08-18

This version is an unofficial release made specifically for Pilot 2, to provide advanced functionalities for the project, on top of V1.1.5.

Features:

- **Enhanced calibration workflow** (crop and presets in the same view)
- **Ingestion of Equirectangular 4K uncompressed inputs through Decklink cards**
- **Supporting cubemap projections**
- **Supporting HEVC RTMP streaming**

Known issues:

- **Need to define input crop areas, as uncompressed 4K does not have a 2:1 aspect ratio**
- **HEVC RTMP is not a standard implementation, as RTMP does not specify HEVC**

3. INTEGRATION

3.1. Workflow integration

For the Phase 1, only two modules were released and imposed integration issues: **Production tools** and **Player**. Production tools should create content that is possible to be played by the player. However, there were more components and functionalities to be integrated for Phase 1:

- **Capture** - Recording omnidirectional and associated directional content using GoPro and Elmo Qbic 360 rigs and traditional cameras. Various testing clips as well as “Dragon Force” movie were shot and edited by Lightbox. The Capture system used in Phase 1 was described in D3.1 Section 4.2.2.1 (omnidirectional cameras) and 4.2.2.3 (directional cameras), and the evaluation of the capture system is available in D4.4 in section 3.1.
- **Stitching** – Combining video images from several cameras was done by Lightbox in Autopano and VideoStitch Studio with assistance of VideoStitch. The off-line stitching system was described in D3.1 Section 4.2.2.2, and the evaluation of the stitching component is available in D4.4 in section 3.2.
- **Encoding** – for Phase 2, partners of the consortium decided to use standard H.264 coding built-in production tools (Adobe Premiere) and defined common parameters to be adapted and used by all the other components related to encoding (stitching, production tools, distribution and reception). The encoding parameters were described in D3.1 Section 4.4.2.1, and the evaluation of the compression functionality is available in D4.4 in section 3.4.
- **Off-line Production Tools** – as mentioned, this component was crucial for Phase 1. It was used for production and synchronization of omnidirectional and directional clips and for preparation the metadata. The architecture and specific workflow of the Phase 1 production tools were described in D3.1 Section 4.3, and the evaluation of the production tools is available in D4.4 in section 3.3.
- **Distribution** – the web server distributing MPEG DASH streams was implemented in order to stream content prepared with using production tools to the client application (Player). The architecture and workflow of distribution server was described in D3.1

Section 4.5, and the evaluation of the distribution server is available in D4.4 in section 3.5.

- **Player** – the end-user application for Windows and mobile phones and VR googles (both on Android) for presentation the streaming content to the user. The evaluation of the player as part of reception and interaction component is available in D4.4 in section 3.6.

The simplified workflow for Pilot 1 is depicted on **Figure 2** and fully described in D3.1 Section 4.1.

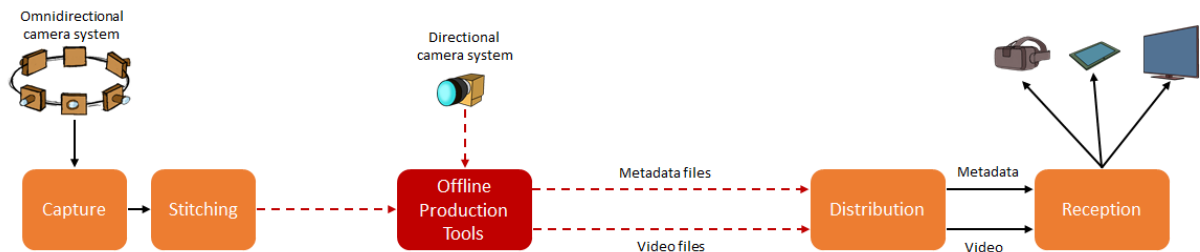


Figure 2: Architecture for ImmersiaTV system for Pilot 1

For the Phase 2, the main integration effort comes with introducing new software - **Live Production Tools**. However all other integration points needed to be reviewed in terms of specifics of live scenario:

- **Capture and Stitching** – Live output from camera systems as described in D3.1 Section 5.2. A specific version of Vahana VR was also created for this purpose, functionalities are described in D3.2 Section 2.3.
- **Encoding** – encoding parameters were described in D3.1 Section 5.4.
- **Live Production Tools** – component crucial for Phase 2, in real time providing and synchronizing omnidirectional and directional output from acquisition components and preparing the metadata. The architecture and workflow were described in D3.1 Section 5.3. Integration with Vahana VR, Orah 4i cameras and the Studio.One camera was also performed.
- **Distribution and Player** – no major architectural changes comparing to pilot 1, however implementations needed to be checked for compatibility with live streams.
- **QoE** – module responsible for logging information from player described in D3.1 Section 5.7.

The simplified workflow for Pilot 2 is depicted on **Figure 3** and fully described in D3.1 Section 5.1.

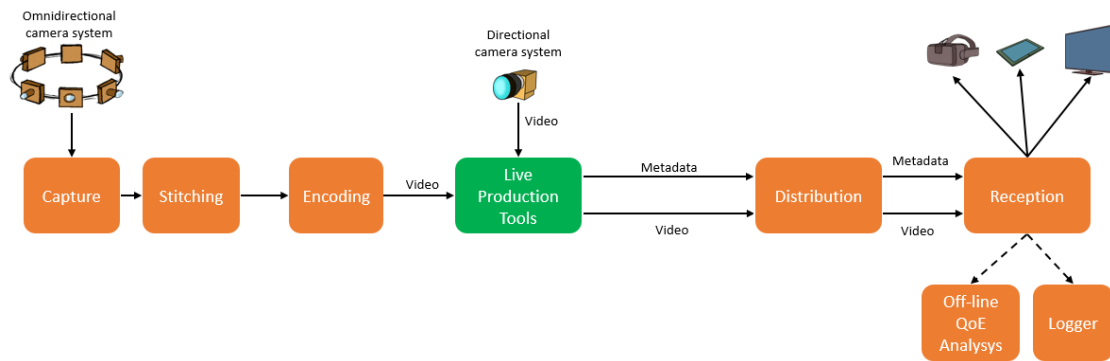


Figure 3: Architecture for ImmersiaTV system for Pilot 2

In all the releases, the workflow and integration of components was tested by functional testing in the software involved and performing qualitative and quantitative tests with media with different resolutions and bitrates.

Modules in the same release were to be compatible, although due to rapid development of new features, the backward or forward compatibility was not guaranteed. On the other hand, the integration of submodules within single module is a constant issue. New features are only merged into component if they don't break existing functionality of a module. This process is depicted on **Figure 4**

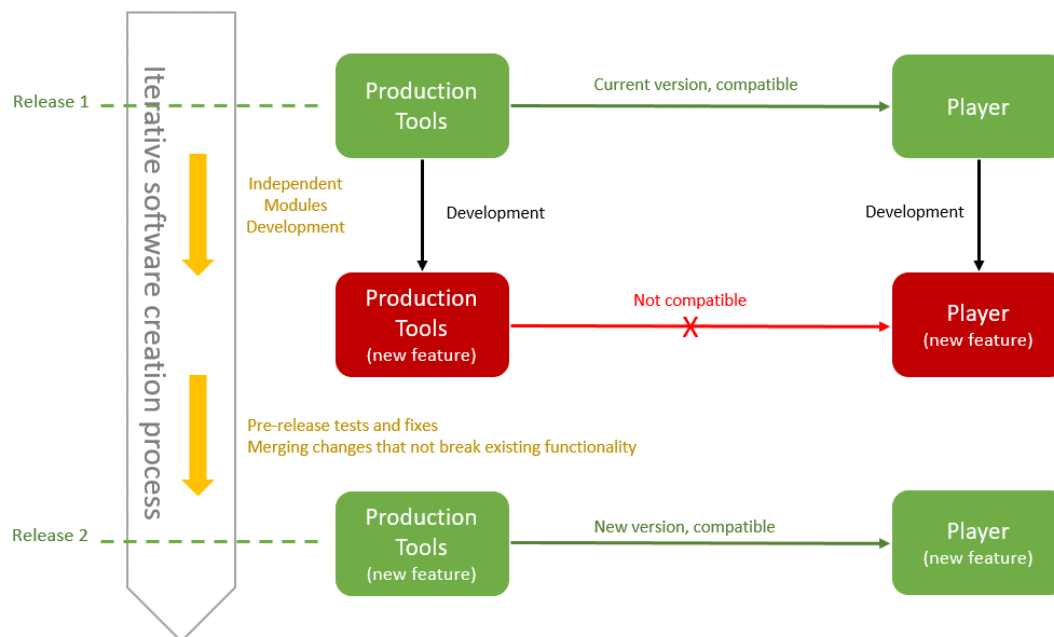


Figure 4: Process of software integration

3.2. Methodology

Development is conducted with **agile approach** in **2 week** sprints. At the beginning of each sprint new tasks are defined – either adding a new feature or modifying existing one. At the end, if the internal tests show the feature to be ready, it is included in the new release.

As the software development process was split by several partners, we organized periodically videoconferences in order to define new functionalities and discuss existing problems, we use slack team collaboration tool to resolve doubts and share documentation.

Internal tests are conducted in development environment. Specifics of application makes automatic tests difficult. Developers follow basic usage scenarios defined by requirements, profile applications, monitor available logs and observe parameters, especially quality and synchronization.

Releases are not publicly available and are intended for project partners to use software in real environment and provide feedback. Reported opinions and minor bugs are then addressed in next sprint. Important bugs are fixed as soon as possible.

Management tools: **Slack** team collaboration tool to discuss new functionalities and doubts with the technical team and describing bugs with other partners, **Kanboard** (Kanban project management tool) for fixing bugs, adding new functionalities and following development process.

The packages for release were prepared in a biweekly basis, developing the different functionalities and fixing bugs the first week, testing the solution the second week each piece of software at a time (post-production tool, transcoding and content server and player) and finally integrating all the components to pack the release.

In order to produce a next release (a tag in git version control solution nomenclature) the software has to compile, this code that has to be in a branch is committed to develop version and when it has been extensively tested a new tag is produced, and that code is frozen and assigned a release version number.

The release version number is based in GNU naming structure X.Y.Z where X is the major version and breaks compatibility, Y is the minor version where the code is compatible with the previous minor release version and adds new functionalities and Z is incremented when a bug is resolved for that release.

After committing the tested release the components are packed individually with automatic building tools, with an in-house building solution for post-production software (plugin for Adobe Premiere Pro) and tested in MacOS and Windows, with a docker container for content and transcoder server and Unity3D building environment for unity player.

Live Production Tools were developed by CINEGY, following similar agile approach. The Planned functionality and bugfixes are grouped into tagged branches in the internal source control system. The preparation of the release itself follows CINEGY's internal development cycles and includes automatic version compilation followed by manual version testing. Once core tests are completed the version is uploaded to the shared repository to make available for ImmersiaTV partners in order to perform additional tests on specific hardware. Software is provided in a form of standard Windows OS installers with all required components included.

The version number naming structure X.Y.Z.W, where X is the major version and breaks compatibility, Y is the minor version where the code is compatible with the previous minor release and adds new functionality and/or bug fixes, Z and W are automatically generated by the build server for internal tracking purposes.

VideoStitch Studio, Vahana VR and Orah 4i camera were developed by VideoStitch, also using an agile approach based on 2-weeks sprints with internal testing and bug tracking system, following VideoStitch's own development cycles.

The integration of Live production tools, Vahana VR and Orah 4i were done on the level of common interfaces and protocols between tools and Immersia TV modules.

3.3. End-to-end tests

3.3.1. Internal system tests

Internal system testing was done by development teams during the iterative work on software, then releases were tested by other partners.

For the **off-line scenario** testing team, composed of video producers and editors from Lightbox dealt with two types of testing: the Immersia TV plugin for Premiere Pro and the Dash Server Converter. Both these features were tested in parallel with the pre-production (while we ran tests), production and post-production of Pilot 1. The area of focus was the plugin itself which is an integral part of the production process. The main issues in first releases were related to slowing down the process due to lack of optimization in the plugin. There were also problems with constant preview which made the editing process difficult. In first releases there were also issues with Metadata exportation – not always it was exported in proper way. The Immersia export panel extension also had a couple of problems with refresh and track selection for exportation. Transitions were also troubling to work with at the beginning - using luma matte clips to transition between two omnidirectional images had a complicated logic or organizing and creating a track hierarchy that was counter intuitive to the normal editing thought process inside premiere. These issues arose mainly between pre-production and production and were reported to implementation teams.

Post-production was slightly different at this phase, these initial issues had been somewhat worked out and corrected. An important problem was also having a big project of very high resolutions in demanding and sometimes unstable plugin. At the same time, there were also problems with other the features of the tool - typology of the embedded portals and the process of creating them. All these errors or problems were reported.

Since then, major improvements have been done - the interface of the plugin and export panel have changed a lot. Luma mattes are no longer needed and everything runs much smoother than previously it did. The process of testing was done iteratively across consequent versions of the production tools, all problems were reported and fixed by development teams.

All major issues related to the player, distribution and production tools were described in Chapter 3 of D4.3 (Technical evaluation).

After production of test clips and "Dragon Force" omnidirectional movie was done improvements to production tools and players were still tested, preparing for phase 3. In parallel tests for **live scenario** started.

The main difference in this case is that all elements of the system work at the same time, requiring tighter collaboration in testing between partners. The part that merges capture and encoding with distribution and client side (elements already used in off-line scenario) are live productions tools, making them the main focus point of live demo integration and testing.

The main test setup was prepared in VRT as the preparation for the final deployment of pilot 2.

3.3.2. Testing workflow in demo environment

Besides continuous internal testing there were several occasions to present the demonstration of the system. Not only it required special testing during preparation, but also presentation in real environment itself gave an important feedback for development.

IBC 2016

The Release 0.6 was prepared in order to make Immersia TV system demonstration on IBC fairs in Amsterdam and present first results for Pilot 1. IBC is one of the most known annual event for professionals of audio-video creation, management and delivery content. In 2016 the event was visited by more than 55,000 attendees from 170 countries around the world and was place where 1,600 companies presented their products and services. The Immersia TV demonstration was hosted by European Broadcasters Union (EBU) booth. This demo integrated presentation of two key components developed so far: the android player and production tools and were supported by third component, the web server working in the background and not shown directly during the demo. The Figure 5 depicts the demonstration environment.

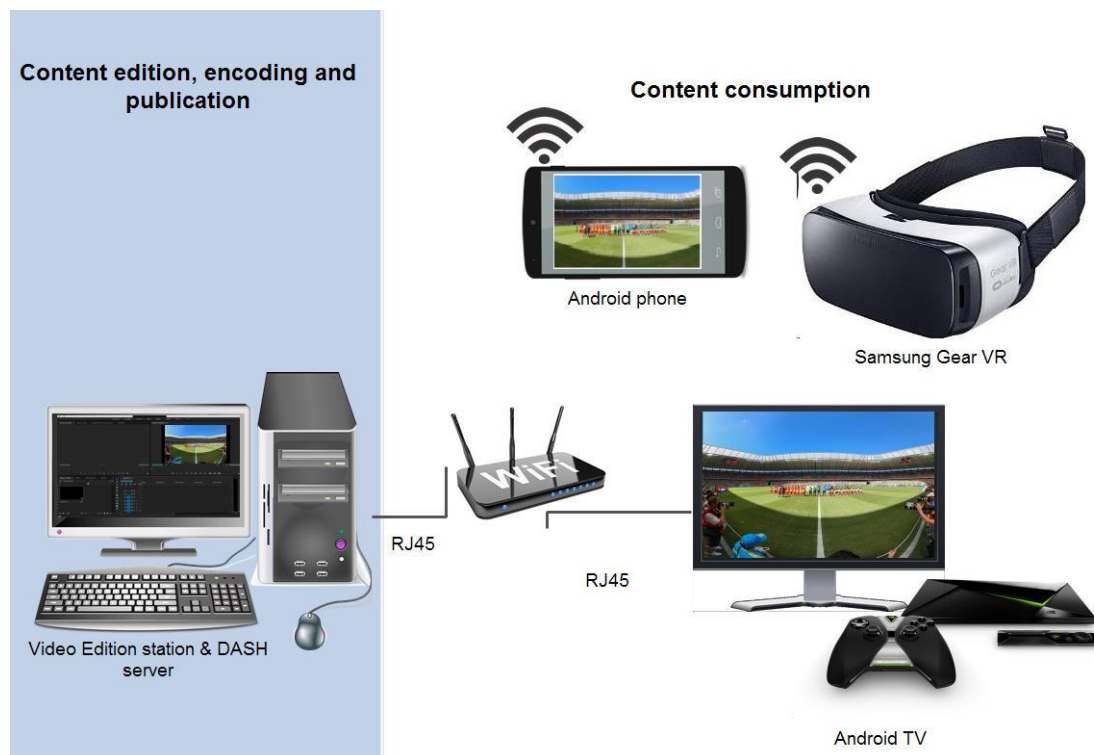


Figure 5: through the immersiaTV services, the content creator can convert his content to DASH and publish the content for synchronized playout during content consumption

The key features demonstrated during IBC were:

- Video-based content delivered synchronously on TV, second screens (tablets) and third screens (virtual reality goggles)
- Portals and video inserts allow defining interactive experiences based on omnidirectional video
- Integration in Premiere Pro for easy content creation

In order to make the demonstration all the components were to be prepared and tested. The “Dragon Force” video clip was prepared and omnidirectional streams were enriched by portals

containing directional videos. The result of using production tools were several video clips and metadata file describing relationship between them. They were streamed by local network (WI-FI router) to the end-user devices: TV (standard TV directional clip) and android devices – tablet and Samsung Gear VR (omnidirectional clips). The synchronization between clips was server by Session Manager.

The main technical problems observed during demo were an entry to the discussion and bug fixing process:

- Mobile devices consumes much power during displaying video clips all the day, and this problem must be resolved by constant connection to power suppliers or power banks in order to not run out of battery during presentation.
- On exhibition area there were plenty of wireless networks operating on the same channels (about 30-50 wireless networks with different session id). They interfered to each other decreasing the bandwidth and reliability of the network. The solution is to look for less occupied channel or switch from 2.4Mhz into 5Mhz frequency.
- Several software bugs were observed, especially related to synchronization and previously mentioned problem. They were a subject to further development and bug fixing process.
- Quality, especially this obtained on VR glasses, was not enough for spectators due to problems with network bandwidth in congested and crowded wireless area. Developers need to improve quality in order to present better demonstration of Pilot 1.
- The low-cost devices like google cardboard and Samsung Gear VR have limitations for the human eye to perceive high resolution videos because of the hardware, more high-end devices like HTC Vive and Oculus Rift will give better results.
- We had to check different synchronization mechanisms because first proposed solution (DVBCSS) gave us some problems and took a time to detect how it affected with the different player technologies been used (Gstreamer and Unity3D).

Having a trade-off between synchronization accurate system and high resolution we have to choose frame accurate sync because is what makes ImmersiaTV unique, the software had to adapt to new requirements, processing CPU and GPU limitations on the devices and software updates.

We learned various lessons, mainly that synchronization and different streams does not just depend on the architecture of the solution but has to be extensively tested in different devices. The network was also involved in the tests and the different network negotiations involved in the solution forced us to have a better knowledge of the infrastructure and try to get the best configuration for the testbeds to achieve best results.

NEM 2016

The next opportunity to present the system was on NEM Summit in Porto. In terms of architecture it was the same demonstration as on IBC, however many problems, especially in synchronization, were solved, leading to much better experience. Stability was also improved. The work of the consortium partners were highlighted, and the ImmersiaTV project won an award for the best NEM demonstration.



TNC 2017

The TNC Networking Conference is one of the largest and most prestigious European research networking conferences. In 2017 the conference took place in Linz, Austria, gathering more than 650 participants. The idea of the demonstration was to present not only off-line use case, but also to introduce elements of live scenario, which was under development at that time. The Figure 6 depicts the demonstration environment.

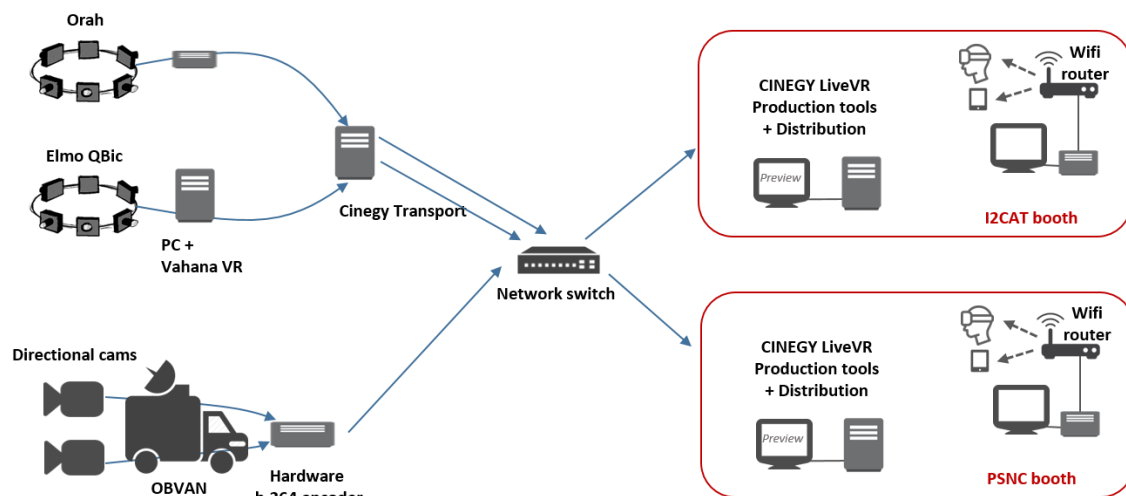


Figure 6: TNC demo

Streams from own directional cameras located in plenary room and two additional omnidirectional cameras (Elmo Qbic rig and Orah 4i camera) were set up and were sending omnidirectional video streams to the CINEGY Live VR tools. It was possible to demonstrate Vahana VR stitching, CINEGY Transport application and interface of Live VR production with live preview.

Due to unstable streams generated by the live tools it was not possible to present full output on the client side. Instead off-line demo with enhanced interactive features (switching camera views) was presented. We also showed integration of the Pilot 2 components: Cinegy Transport tools, and Vahana VR with Elmo QBic rig.



Figure 7: TNC2017 demo on photographs. On the left there is CINEGY transport tools integrated with the Elmo Rig camera. On the right photograph, Immersia TV team is presenting live components

IBC 2017

Another try to perform full live demonstration occurred on IBC 2017. Scenario with streams from 2 omnidirectional and 2 directional cameras, managed by live production tools and delivered to client devices was prepared. However system was not stable enough and not ready to perform in non-lab environment due to large dependency on stable network conditions that were affected by large number of concurrent WiFi networks using the same channels in the demo area. The presentation fall back to offline demo of Pilot 1 with demonstration of Cinegy Live VR interface, operator workflow in Live VR, live streams produced by the camera and limited live streams display via ImmersiaTV WEB based players.

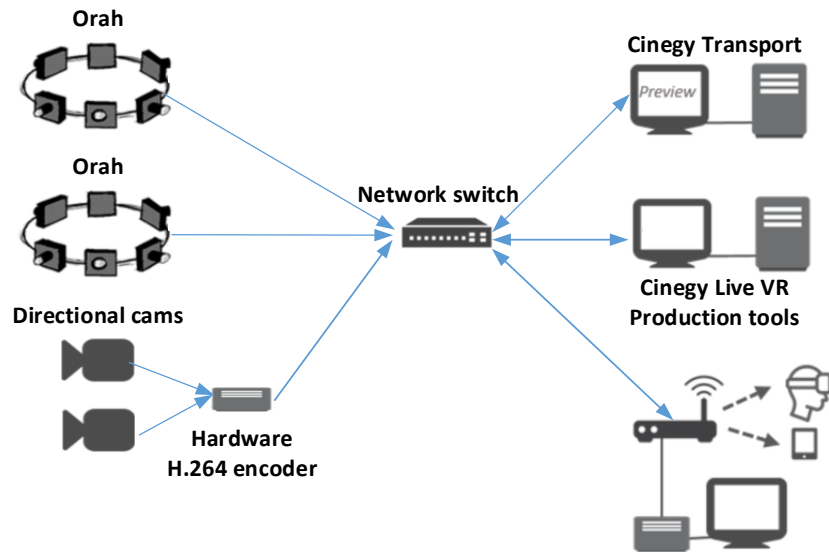


Figure 8: IBC demo



Figure 9: Immersia TV demo booth at IBC Future Zone in Sep 2017

4. CONCLUSIONS

The integration of the Immersia TV system which is developed by several partners of the project, require good coordination and communication inside the consortium. Both, the requirements

for the system (D2.2 prepared by WP2) and software architecture description (D3.1 prepared by WP3) need to take into consideration the iterative process of the software development. Requirements on common codec and interfaces between modules and must be perfectly defined in order to avoid integration problems.

Although, both internal testing (done by software developers and testers within teams) and external testing (by other partners making use of the software) are crucial for the development process and bug fixing, good opportunity for finding unexpected problems are demonstrations in production environment. That's why taking part in open pilots, exhibitions and conferences is so important, because it gives a lot of feedback for developers.

A common test and develop environment between partners responsible for components being part of the releases, gives lower response time to bugs and to adapt new functionalities as does a team with version control experience and agile techniques to adopt better strategies when we needed to boost the performance. The process of cooperation and remote collaboration worked pretty well during first releases.

Using external software, such as GStreamer requires specific knowledge and experience. This streaming framework keeps changing all the time, what causes integration problems. That's why I2CAT and PSNC keeps contact to GStreamer community (including taking part in GStreamer workshops) in order to track changes and have an assistance in resolving streaming and synchronization issues.

Last conclusion is that integration of Pilot 2 Live Tools with existing components takes more time than expected and is more complex, so additional efforts were put into this activity. The stability of the streams and the network deployment affected dramatically the progress in this area, so the whole chain of testing was delayed.