ECSEL2017-2-783162

FitOptiVis

From the cloud to the edge - smart IntegraTion and OPtimisation Technologies for highly efficient Image and VIdeo processing Systems

Deliverable: D6.1 Demonstrators Framework Definition and Metrics

Due date of deliverable: (31-05-2020)
Actual submission date: (29-05-2020)

Start date of Project: 01 June 2018 Duration: 36 months

Responsible WP6: Thales Alenia Space in Spain (TASE)

Revision: Approved

Dissemination level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Service	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

0 DOCUMENT INFO

Author

Author	Company	E-mail
Marcos Martínez	TASE	marcos.martinezalejandro@thalesaleniaspace.com
Massimo Massa	AITEK	mmassa@aitek.it
Stefano Delucchi	AITEK	stefano.delucchi@aitek.it
Giuseppe Meloni	Absinsula	giuseppe.meloni@abinsula.com
Paolo Giugliano	IsaRail	
Luigi Longhi	SAT	luigi@acquedottitirreni.it
Carlo Sau	UNICA	carlo.sau@diee.unica.it
Francesca Palumbo	UNISS	fpalumbo@uniss.it
Simone Vuotto	UNISS	
Laura Pandolfo	UNISS	
Tiziana Fanni	UNISS	
Luca Pulina	UNISS	
Giacomo Valente	UNIVAQ	giacomo.valente@graduate.univaq.it
Vittoriano Mutillo	UNIVAQ	vittoriano.muttillo@graduate.univaq.it
Jukka Saarinen	NOKIA	jukka.saarinen@nokia.com
Francisco Barranco	UGR	fbarranco@ugr.es
Daniel Déniz	UGR	danideniz@ugr.es
Ricardo Ruiz	RGB	rruiznolasco@rgb-medical.com
Luis Medina	7SOL	luis.medina@sevensols.com
Raúl Santos	HIB	rsantos@hi-iberia.es
David Millán	ITI	dmillan@iti.es
Guillermo Amat	ITI	gamat@iti.es
Lukas Marsik	CAMEA	l.marsik@camea.cz
Rob de Jong	PHL	rob.de.jong@philips.com
Geran Peeren	PHL	geran.peeren@philips.com
Roman Čečil	UWB	rcecil@kky.zcu.cz
Martin Švejda	REX	svejda@rexcontrols.cz
Zdeněk Pohl	UTIA	xpohl@utia.cas.cz
Pablo Sánchez	UC	sanchez@teisa.unican.es
Pablo Chaves	SCHN	pablo.chaves@se.com

Documents history

Document version #	Date	Change
V0.1		Starting version, template
V0.2		Definition of ToC
V0.3		First complete draft
V0.4		Integrated version (send to WP members)
V0.5		Updated version (send PMT)
V0.6	24/04/2020	Updated version (send to project internal reviewers)
Sign off		Signed off version (for approval to PMT members)
V1.0		Approved Version to be submitted to ECSEL office

Document data

Keywords	Use Case, Final Demonstrator, Partial Demonstrator, Work Package 6
Editor Address data	Name: Marcos Martinez Partner: TASE Address: Einstein 7, 28760 Tres Cantos (SPAIN) Phone: +34 91 8070 7900 Ext. 9020

Distribution list

Date	Issue	E-mailer

Table of Contents

1 Executive summary

This deliverable reports on partial demonstration results. A detailed description of the full demonstration framework of each use case is included in this document. Each use case provider has also included a description of which technologies from the final demonstrator will be included in the partial demonstrators due at M24. As the demonstrators of FitOptiVis are used to provide real-world industrial implementations of the FitOptiVis framework an explicit relation between each demonstrator and the different FitOptiVis technologies that are being used in the demonstrators is also included in this deliverable.

The second goal of this document is to provide a list of common metrics for evaluating the whole FitOptiVis project. Each use case leader has elaborated a list for the metrics that could be measured in order to evaluate its demonstrator. From the common metrics found across the different use cases a table with a set of common metrics is also provided in this deliverable. Taking as input previous work done in WP3, a set of common tools per use case can be found in the document.

Finally, each use case provider has made explicit how to consider the feedback provided by the different end users that were present at the First FitOptiVis End User workshop held in Eindhoven in September 2019.

The document is divided in use cases. Each use case has the following sections:

- Full Demonstrator Description: In this section the use case providers fully describe what their demonstrator aims to achieve by the end of the project, It also includes the reference architecture of the proposed demonstrator that has been built with QRML, developed in WP2. In this architecture the components described in WP5 have been included. Finally, the demonstrator that will be presented in June (M24-demonstrator) is also described, as a partial version of the Full Demonstrator.
- Use Case in FitOptiVis: In this section a brief table relating the use case with the different parts that are built in FitOptiVis is presented. The goal of this section is to show the relations of every use case with the developed technologies.
- Use Case Metrics: The metrics for each use case have been defined in WP1 and can be found in D1.4. They are divided in User Needs and Use Case Requirements as means for validating and verifying the functionalities of each Use Case.
- End User Feedback and recommendations: In this section each use case provider refers on how they will address the comments made by the different end users at the First FitOptiVis End User Workshop.

The table below lists all the use cases and their owners within the project. Each use case will be analysed on validation and evaluation in this document.

Use case	Title	UC leader	Partner	Email
1	Water supply	Massimo Massa	Aitek	mmassa@aitek.it
2	Virtual reality	Jukka Saarinen	Nokia	jukka.saarinen@nokia.com
3	Habit Tracking	Ricardo Ruiz	RGB	rruiz@rgb-medical.com
4	3D industrial inspection	Santiago Cáceres and Ricard López	ITI	scaceres@iti.es rlopez@iti.es

Use case	Title	UC leader	Partner	Email
5	Road traffic surveillance	Lukáš Maršík	CAMEA	l.marsik@camea.cz
6	Multi source streaming composition	Rob de Jong	Philips	rob.de.jong@philips.com
7	Sustainable safe MRI	Geran Peeren	Philips	geran.peeren@philips.com
8	Robots calibration	Pavel Balda	REX	balda@rexcontrols.cz
9	Surveillance of smart-grid critical infrastructure	Luis Medina	Seven Solutions	luis.medina@sevensols.com
10	Autonomous exploration	Marcos Martinez	Thales Alenia Space	Marcos.martinezalejandro@thalesaleniaspace.com

2 Use Case 1 – Water Supply

The goal of the use case 1 is to study and implement an innovative and (semi)autonomous surveillance system for a water supply critical infrastructure. Heterogeneous SW and HW elements compose such system, including smart cameras, distributed video processing pipelines sensors and actuators and data processing units that trigger autonomous actuations. The system includes two high level goals that correspond to two different scenarios considered in FitOptiVis: *i)* unauthorized access detection and *ii)* damage/leakage detection.
This monitoring system will require components development during the project lifecycle to meet end-user needs and requirements, and to apply the technologies and the methodologies defined during the project. These components will be part of the final prototype to demonstrate the benefits provided by FitOptiVis and it will include the most relevant components that highlight project benefits, while the remaining will be implemented in order to complete the system, although they will not be part of the validation process.
This chapter describe a full system architecture, with the functions that will be implemented in the water supply monitoring system scenario. For each component there will be a dedicated comment related to the project scopes (e.g. a key element that demonstrate some aspects of FitOptiVis or a complementary element needed only to deploy the prototype). In particular, the focus will be on the components developed during the project. Such components will be further described and modelled using the DSL language defined in the WP2.

2.1 Full Demonstrator Description

The demonstration will consist of an integratedintegrated monitoring and control system that will be used to demonstrate its functionalities, in order to meet the user needs and requirements. The system adaptivity will be demonstrated at system level, as an adaptation of the system with respect to the external detected conditions. Moreover, some specific functionalities will be selected and shown to demonstrate reconfigurability at component level.
Finally, there will be a differentiation also regarding the application scenario. As defined since the beginning of the project, use case one includes two separated scenarios. The first one is about water supply security focusing on unauthorised access detection. It is more mature and its description is already available. The second scenario is about water leakage detection and its details will be defined during the last year of project.
Moreover, system adaptivity in scenario one will be implemented considering two different working modes:

- The monitoring system usually works in a **Normal mode**:
    - It consists of Video Content Analysis (VCA) based on traditional approaches. It provides **limited functionalities** (e.g. only moving target detection inside the monitored area).
    - It requires **less computational resources** (processing on the edge) and/or **less bandwidth** to transmit a video flow with limited quality and frame rate (for streaming and processing in cloud).
- When it is needed, the monitoring system can switch to **Performance mode**:
    - It consists of VCA based on Deep Learning approaches, able to provide **advanced functionalities** (e.g. intruder classification – person vs animal, face recognition and actions or behaviour detection).

- It requires **high computational resources** (processing on the edge) and/or **high bandwidth** to transmit a video flow with proper quality and frame rate (for streaming and processing in cloud).

We are also defining some policies that enable the system to switch from one modality to another one. Such mechanism can be briefly summarized as follows:

- Trigger from Normal to Performance: Detection of a safety/security threats (e.g. an intruder)
- Trigger from Performance to Normal: Problem solved/taken over by human operator

Further and more concrete details will be provided later during the third year of project. A similar approach is currently under definition for the second scenario, which includes also the drone. It will be based on different activities done by the drone itself that may consist of flying to a specific destination or collecting and streaming data and/or video at different frame rates or quality. In particular the drone will communicate with a Ground Station, transmitting a video stream and telemetry data using a radio frequency channel. Such streams will be provided to the monitoring system using a field gateway.
System level adaptivity will be defined in detail and implemented during the third of project.

2.1.1 Reference Architecture

Figure 2-1 shows the high-level architecture components of the management and control system for the water supply scenario. The orange modules describe components that will be part of the demonstrator, while the white ones are existing components that can be included in future applications, but are not currently part of the demonstration scenario.

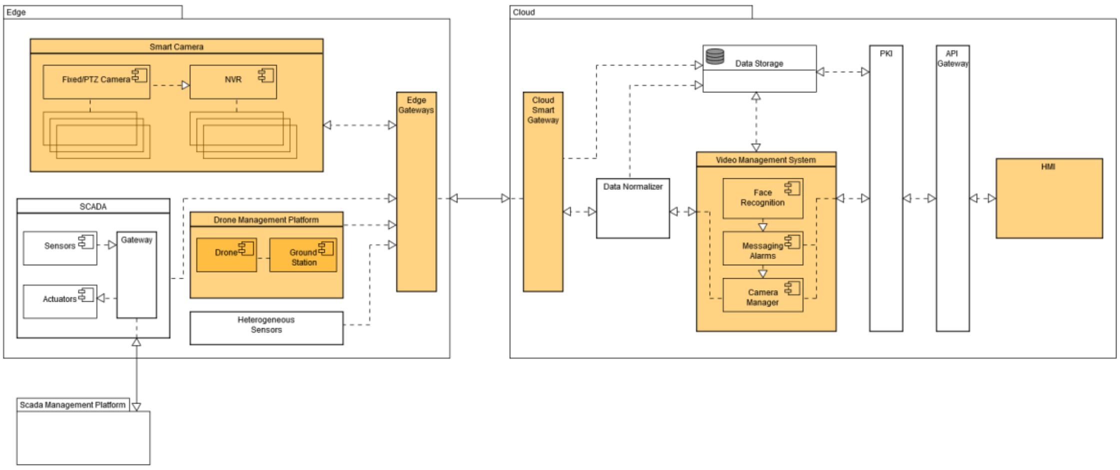


Figure 2-1: Use Case 1 Full System Architecture

Smart Cameras: The edge module dedicated to the management of cameras. The communication with the Edge Gateway is bi-directional, so that the cameras can also be controlled from the Cloud platform.

- **Field and Pan-Tilt-Zoom (PTZ) Camera**: The physical devices, fixed and PTZ cameras are cabled to communicate and store the video sessions in the NVR modules. The PTZ camera can be remotely controlled either manually or automatically based on the developed algorithms that run in the cloud within the Camera Manager module.

- **Network Video Recorder (NVR)**: This module stores the recordings from the camera field and applies the required video processing in order to send the computed information over the cloud to the Video Management System module.

Smart cameras are central elements in FitOptiVis demonstrator; therefore they will be integrated in the final prototype.

SCADA: Sensors and actuators that control the physical component of the water supply system (i.e. distribution tank, water pipelines) are currently monitored and manged by a SCADA system infrastructure.

- **SCADA sensors/actuators**: Sensors and actuators of the existent SCADA platform on-site.

- **SCADA Gateway**: the existent gateway on-site dedicated to bi-directional communication of the collected data from sensors.

The SCADA is a quite important element that is out of the scope of FitOptiVis but it is mentioned here to give a general description of the complete control system.

SCADA Management Platform: The existent SCADA management platform on-site that controls the water management system. In a possible future scenario, the data can be merged and computed on the Fitoptivis Cloud Framework and integrated in the HMI monitoring and actuation functionalities.
This component could be demonstrated only at simulation level during the project (as it is not possible to interrupt the water distribution to citizens).

Drone Management Platform: the platform dedicated to handle the drone deployment and the data collection.

- **Drone**: the drone itself is equipped with an HD camera and with a radio communication module that can send the gathered telemetry or video data on different radio frequencies based on the required bandwidth.

- **Ground Station**: acts as a ground gateway to control the drone during the flight. It also collects all the data gathered during the flight session and send them over the Edge Gateway.

So far, these components are not yet included in the first release of the demonstrator.

Heterogeneous sensors: the FitOptiVis framework is conceived to scale horizontally, so that other sensors can be used to gather useful information from the field scenario and then merged with data fusion techniques directly on cloud.
So far, these components are not yet included in the first release of the demonstrator.

Gateway to collect data: It consists of two different logical elements as described below:

- **Edge Gateways**: The gateway collects all the field data coming from the Field cameras and the Drone Management Platform and forwards them to the Smart Cloud Gateway, or from the Cloud Gateway to the Field cameras.

- **Cloud Smart Gateway**: The module is capable of dynamic data handling, so that the received messages can be tagged based on metadata content and processed based on a high or low priority queue bi-directionally.

In the final demonstrator they could be integrated in a unique software element

Data Normalizer: retrieves the raw data collected from the field or converts the requests generated by the user and normalize them so that the data could be processed by either the edge or cloud dedicated modules.
Video Monitoring System - VMS: The module processes the normalized data coming directly from the field cameras. The bi-directional flow allows to normalize a request sent from HMI module and forward it to the Field Camera module in case of manual override of the PTZ cameras.

- **Messaging and Alarms module**: events and messaging are generated based on the processed data coming from the field NVR or the Face Recognition module. The events are sent to the HMI in order to be handled by the operator.
- **Camera Manager**: This module is designed to handle automatically the PTZ camera functions based on the video processing received from the field. The cropped images are then stored and sent to HMI module.
- **Face Recognition**: This module will perform face recognition functions based on the inputs received from the other camera modules. The output is sent to the Messaging and Alarms module in order to handle the events.

It is important to say that this component will be included in the final demonstrator.

Data Storage: the raw and the normalized data will be stored in dedicated Database solutions in order to correctly handle events, drones GIS (Geographic Information System) data and historical information with a combined use of short-term and long-term storage solutions.
Public Key Infrastructure (PKI): The layer provides a set of rules, policies, software and procedures to manage digital certificates and public-key encryption.
API Gateway: provides the required abstraction layer to communicate with FE applications through HTTP and WebSocket protocols.
Data storage PKI and API Gateway are important components of a real system but they are currently out of the scope of the project. Therefore, they are only mentioned here but not included in the demonstrator.

HMI: A Graphical User Interface to manage the monitored processes of the whole water supply infrastructure. There will be sections dedicated to the management and field view of the cameras, the drone related data (i.e. runtime position within a GIS map, pictures shot and video), messaging and alarms and history related data (i.e. logs, stored sessions recordings). To enable bi-directional communication with the field, it will be possible to enforce some control actions via graphical interface on the controlled modules through the dedicated infrastructure gateway (i.e. manual control override of PTZ camera).
This component will be part of the demonstrator as it will enable to control the system, to view its behaviour and the data collected and processed.

In addition to the traditional system-wide architecture, a portion of the final demonstrator architecture will be also complementary presented based on the QRML DSL language, which can be useful to represent the hierarchical structure of the system components. Figure 2-2 shows an example of the UC architecture exported using DSL methodologies, explaining all the interfaces dependencies and the functions provided by the deployed modules, i.e. Smart Camera and Local Station.

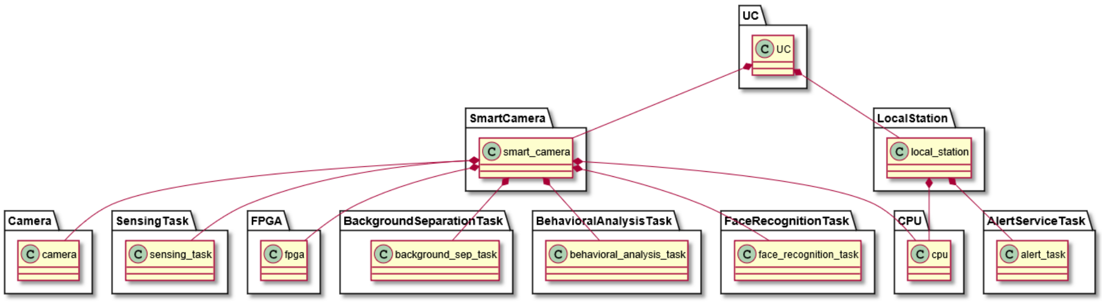


Figure 2-2:-Architecture of UC1 demonstrator [draft]

2.1.2 M24 Partial Demonstrator

Some components of the video monitoring system for unauthorized access will be demonstrated at M24.

- System interface, which enables human operators to control the whole system and to supervise its autonomous behaviour.
- Smart camera that consists of camera module to acquire images and an edge device in charge of process images in order to real time detect intruders in a restricted area.
- Adaptivity will be partially demonstrated at system level as the quality of the monitoring will be increased or decreased according to the dangerousness of the detected events.

2.1.2.1 M24 to M36 GAP

The demonstrator at M24 will be a collection of components that will be demonstrated as -stand-alone systems. Such components will be - fine-tuned and modified according to feedback provided by reviewers (i.e. during Y2 review), and end user board.
Moreover, such components will be integrated in a final prototype that will be demonstrated at the end of the project.
In more details, at M24 the focus will be only on the first scenario, which is about unauthorized access detection. At M36 also the second scenario, which is about the detection of damage and water leakage, will be demonstrated.

2.2 Use Case 1 in FitOptiVis.

Include in this section a table with a relation between your demonstrator and FitOptiVis technologies.

Demonstrated Technology	WP in FitOptiVis	Role in Demonstrator	Date for demonstration (M24/M36)
Reference architecture	WP2	Provide the architecture of the demonstrator	M24
Parallelization with OpenMP	WP4	Speed up the processing of images	M24
Design Space Exploration for reconfigurability	WP3	Provide suitable reconfiguration plans with respect to performance and energy/power trade-offs	M30
Customizable runtime monitoring of hardware accelerators	WP4	To perform the monitoring of some parameters to trigger reconfigurations and to runtime verify the system state.	M30
Assessment of the NEURAghe accelerators to implement UC relevant networks	WP3 & WP5	To access the potential advantages/disadvantages of custom HW accelerators with respect to the given scenario.	M30
Assessment of MDC-compliant accelerators to implement UC relevant networks	WP3 & WP5	To access, where present, the possibility of supporting different execution trade-offs executions.	M30
Video content analysis platform	WP5	Core video surveillance and processing units used to monitor the water supply	M24 - M36

2.3 Use Case 1 Metrics

2.3.1 User Needs

User Need	Validation method	Comment
Reliability, security and surveillance improvement	Detection of intruders in forbidden areas by means of an integrated system: visualisation of alarms for un-authorised accesses and of the acquired and processed images in the unified FitOptiVis HMI (remotely installed in the control room).	This is the goal of the final prototype. It will be validated at M36 and some preliminary evaluation could be done earlier.
Increase monitoring automation	1. Autonomous tracking of the people inside the forbidden/restricted accesses areas, using multiple fixed/PTZ cameras	This need will be validated at M36 when the complete prototype will be available. Some functions (as for example part of point 1 or

User Need	Validation method	Comment
	and the drones (in particular for remote zones). 2. Capability to autonomously distinguish between people and animals inside the area, to reduce false alarms. 3. Face recognition feature to autonomously verify people identity within the forbidden/restricted accesses areas to exclude authorized SAT employees or known guest from the control of their action inside the sensitive area. This will allow to focus on people who can really represent a risk for the infrastructure security	point 2) could be addressed already at M24
Reduction of human operators risks	As a positive consequence of the increased monitoring automation, less human interventions are expected.	The validation of these needs is pending as all of them are linked to the first one which is not yet validated. Increased system automation will pave the way to human intervention, recovery time and costs reduction.
Recovery time reduction (must be intended as reduction of overall time needed to successfully manage the event that caused alarm)	Reduction of intrusion detection time (real-time) and alarm verification (real-time).	
Cost reduction	Consequence of reduction of human operators checking images and managing interventions in case of danger/risk have a positive influence in costs reduction.	

2.3.2 Use Case Requirements

ID	Use Case Requirements	Verification Method	Comments
A1	Integrated system SHALL enable (on board and/or on centralized) processing of information from smart cameras, heterogeneous sensors and drones based on availability of wired or wireless connectivity	Comparison among different execution of video and other data processing on board and on centre. Assessment and evaluation of the results derived from the adoption (where possible) of different algorithms and architectures.	
A2	Integrated system SHALL enable remote control (autonomous or manual) of heterogeneous actuators, PTZ cameras and drones	Remote control (autonomous or manual) of actuators and verification of the behaviour wrt the end user expectations.	The system interface will be shown in a draft modality at M24. Regarding this requirement, it will be possible to do the following actions using such interface:

			1) Control PTZ cameras (automatically or manually) 2) Control the drone 3) Ongoing definition of which are the actuations that can be showcased Note: all these functions will be demonstrated between M24 and M36
A3	Integrated system SHALL be able to support computing heterogeneity, leveraging on FPGA platforms on the edge for developing accelerators and monitoring systems and SHOULD enable support of adaptivity by means of HW reconfiguration	Assessment and evaluation of the results derived from the adoption (where possible) of different accelerators and HW/SW partitioning	Currently different FPGA based accelerators are under development. MDC-based accelerators and AIPHS monitoring infrastructure are already integrated and assessed on a small video processing testcase. Stand-alone assessment on use case relevant algorithms are intended to be carried out by M30 to identify whether there will be any advantage in integrating them in the final demonstrator.
B1	Integrated system SHOULD make available a single unified HMI, easy to use and with a short and simple training	Verification whether all the installed sensors and actuators can be managed by the same HMI. End users feedback will be collected and analysed to confirm the completeness and the ease of use of the HMI.	This verification will start as soon as the interface is available. SAT employees will be involved to fill a questionnaire to collect relevant feedback about the usability of such interface (M24-M36). These feedbacks will be collected and evaluated periodically as they could improve the refinement of the interface. Final evaluation will be presented at M24
B2	When the operator visualizes an alarm, she/he SHOULD be enabled to implement appropriate countermeasures or supervise the actions executed automatically by the system.	Check if the correct corresponding countermeasures are automatically implemented (or taken) when the different suspicious events are detected.	To demonstrate this requirement UC1 partners with the supervision of SAT will define a specific sequence of countermeasures that corresponds to each detectable threat. They will produce a short description of such countermeasures at M24, while their correct implementation will be verified and demonstrated between M24 and M36.
F1	Each abnormal event detection SHOULD generate an alarm displayed on HMI, the alarm SHOULD remain in the "to be taken in charge" state until the operator visualizes it.	Check if the proper alarms are correctly and effectively displayed on the HMI in a visible and intuitive (for the operator) way when an event classified as “potentially dangerous” occurs. Check if the alarm remains in evidence on the HMI	Events will be simulated to verify if the system can detect them correctly and generate the correct alarms to be displayed on the interface. The detection of such events will trigger the countermeasures considered in the previous requirement (B2). Therefore, the demonstration of this requirement will be carried out together with the previous one.

		until the operator confirm the alarm notification	
F2	Each event detection SHALL be georeferenced and the coordinates SHALL be shared with the whole system	Simulations of different relevant events in known positions. Check if the detected locations are correct and properly displayed on the HMI	The HMI of the system contains different view modalities. One of them is a map of the monitored area where data collected by sensors and cameras as well as their status and position are shown in real time. This map will be shown at M24 with simulated data and sensors while between M24 and M36 it will be populated with data from real sensors and cameras.
F3	The drone SHOULD navigate through the area where the event has been detected and patrol it. PTZ cameras SHOULD be automatically oriented to monitor the same areas	Simulations of different events in known locations to check if these events are visible using cameras (including also the one installed on the drone).	Algorithms to automatically orient PTZ cameras are under development and some preliminary results could be already available at M24. Such requirement will be completely demonstrated at M36. Italian laws are quite restrictive concerning the use of drones. Therefore, due to such limitation there could be some issues related to the drone patrolling.
F4	HMI SHOULD show and store all the video streams acquired in the area where the event has been detected. It SHOULD also show the data acquired by sensors in the same area (including video and data from drone).	Simulations of different events in known locations to check if the right videos and data are available and easily retrievable on the HMI.	This requirement will be verified during between M24 and M36.
F5	The system SHALL detect intrusions in restricted areas, unauthorized accesses, leakages and damages	Simulations of these events and check if the system can really detect them.	Scenario 1 includes intrusion and unauthorized access detection while Scenario 2 includes leakage and/or damage detection. Scenario 1 will be partially demonstrated at M24. Both scenarios will be fully demonstrated at M36.
F6	Functional adaptivity SHOULD be enabled at the hardware accelerators level (accelerated functionality may change, e.g. video processing such as enhancement, filtering, coding)	Comparison among functional mode where possible.	MDC-based accelerators already proved in other contexts to be able to support functional adaptation. In the context of this use case, given the provided applications, it is more likely that non-functional oriented adaptation and trade-offs management will be implemented at the accelerator level.
NF1	The UI SHOULD be easy to use for a trained factory operator	End users feedback will be collected to evaluate the usability of the HMI	The verification of this requirement will be done in the same way with respect to requirement B1
NF2	Real time monitoring SHOULD not affect the system's performance	Comparison among monitoring-enabled and	This requirement is under revision, it could be updated, therefore it will be

		monitoring-disabled infrastructures.	reconsidered later, during the third year of project
NF3	The time between the detection and the alarm SHALL be few seconds	Simulations of events at known moment to measure the time between the event simulation and the alarm generation. Collect the feedback of the end users to understand which alarms should be spread immediately and which can be treated with less urgency	This assessment will be done during the last year of the project. Time log will be carefully considered.

2.4 End user feedback and recommendations

During the first workshop, end user board discussed and analysed each use case in detail providing relevant guidelines and suggestions.
In particular, regarding use case 1, the discussion covered both scenarios, the “Unauthorised access” and the “Leakage/Damage Inspection”. In particular, the first scenario was analysed in detail. The main outcomes of this discussion are reported here below, adding some explanation on how they have been addressed in the scenario development.
End user stressed the importance of finding an optimal compromise between resource consumption (in terms of energy consumption, computational power and bandwidth) and performance. The ideal (utopic) solution is to minimize the first while maximizing the latter, but obviously this is not always possible. As a consequence, in particular for a distributed monitoring system, this balancing is particularly relevant.
Partners involved in UC1 *we have taken into consideration this recommendation. In fact, the monitoring system under development has been designed to have adaptivity at system and component level, as previously described. Therefore, it will be possible to adapt performance and resource consumption at both levels.*
End user highlighted the importance of the automation for a monitoring system, and particularly for security applications like unauthorized access detection. Such importance increases if it is applied to monitor a critical infrastructure like a water supply system.
This recommendation is perfectly aligned with respect to ***UC1 partners’*** *vision. As a matter of fact, system automation is a relevant feature for use case 1 already identified during the requirement definition in WP1. As a matter of fact, it represents an important improvement with respect to the system currently adopted to monitor the water supply. Such system has a very limited automation as unauthorized accesses are detected by a human operator that has to be physically on the field. The final prototype will provide important benefits as it will automatically detect presence of introduces inside restricted areas, analysing images acquired by cameras. In this way a human operator will be able to control the situation remotely from a control room reducing reaction time, limiting the costs of the intervention and avoiding exposure to potentially dangerous situations.*

3 Use Case 2 – Virtual Reality

3.1 Full Demonstrator Description

VIDEO-BASED POINT CLOUD COMPRESSION IN VIRTUAL REALITY USE CASE

In the present deliverable, D6.1, we will give an overview of the current Video-based point cloud compression (V-PCC) demo implementation, its encoding and decoding characteristics. Furthermore, a description of the requirements and envisioned applications for V-PCC is also provided. It is expected that with the emergence of new, immersive 3D media applications, point clouds and compression technologies, such as V-PCC, will become an indispensable component for representing, delivering, and visualizing 3D representations and objects in VR / AR environments.

Real-time 3D scenery detection and ranging has become an important issue for the emerging field of Virtual Reality (VR) applications. These VR technologies require efficient 3D imaging representations in terms of complexity of rendering, ease of management, compression, and capture, among others. The point cloud representation is a good compromise that can address all of these market and application requirements. Therefore, Nokia has selected Video-based point cloud compression as the core technology enabler in the VR Use Case [1-2,7].

The main philosophy behind V-PCC is to leverage existing video codecs for compressing the geometry and texture information of a dynamic point cloud. This is essentially achieved by converting the point cloud into a set of different video sequences. In particular, three video sequences, one that captures the geometry information, another that captures the texture information, and one that captures the occupancy information of the projected point cloud data, are generated and compressed using existing video codecs, such as MPEG-4 AVC, HEVC, AV1 etc. Additional metadata, which is needed for interpreting the two video sequences, so called auxiliary patch information, is also generated and compressed separately. The video generated bitstreams and the metadata are then multiplexed together so as to generate the final point cloud V-PCC bitstream.

It should be noted that the metadata information represents a relatively small amount (i.e., 2-5%) of the overall bitstream. The bulk of the information is handled by the video codec. Figure 3-1 and Figure 3-2 provide an overview of the V-PCC compression and decompression processes, respectively. Next we provide an overview of the V-PCC process, its encoding and decoding characteristics as part of FitOptiVis solution [1-2,7].

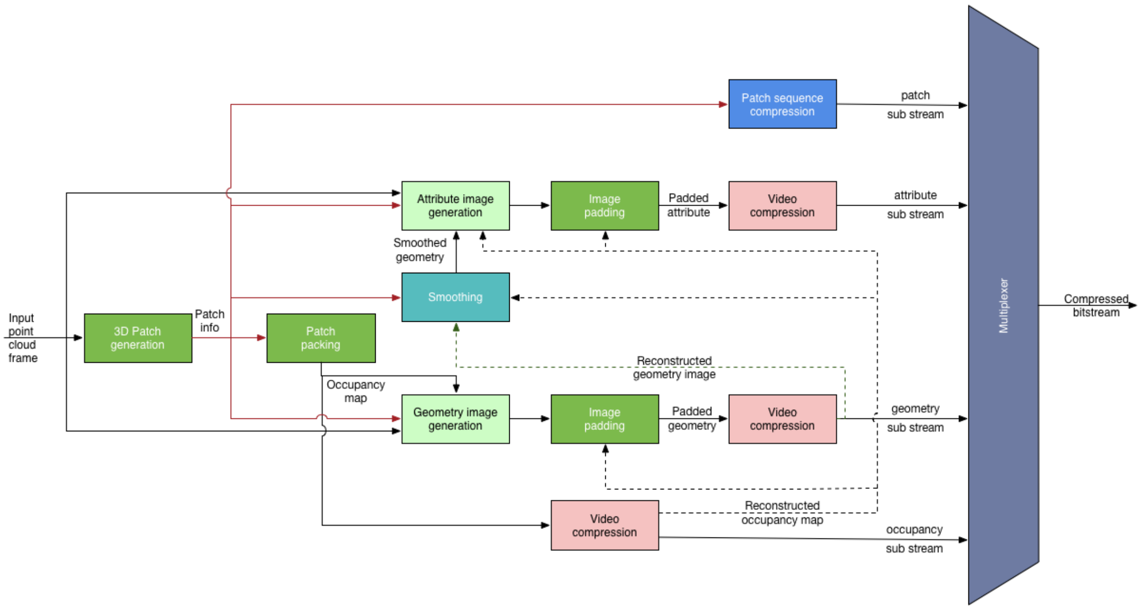


Figure 3-1 V-PCC encoding Process

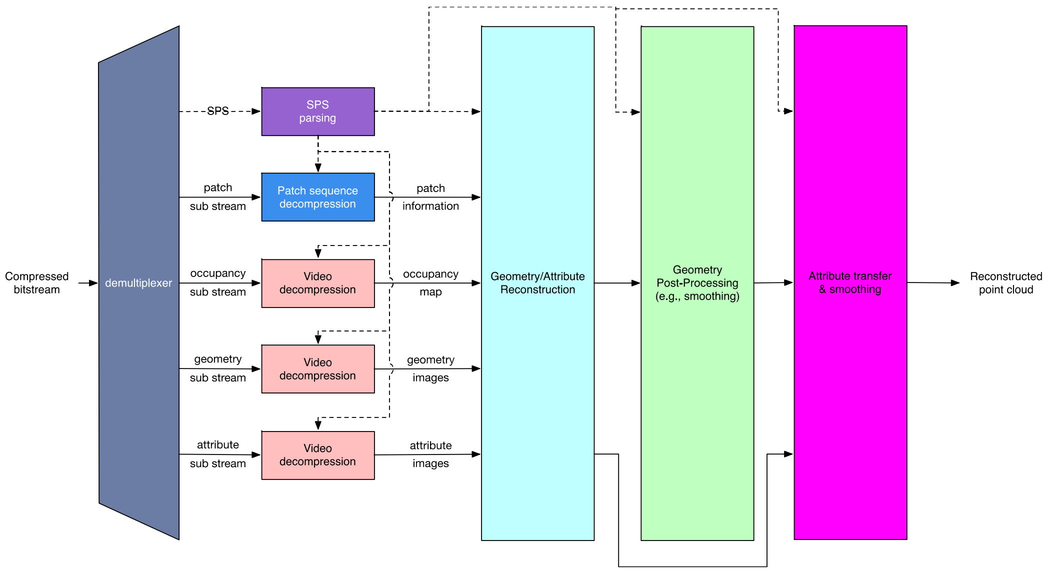


Figure 3-2 V-PCC Decompression Process

Encoding process

V-PCC exploits a patch-based approach to segment the point cloud into a set of clusters (or patches). These patches can be mapped to a predefined set of 2D planes through orthogonal projections, without self-occlusions and with limited distortion. The objective is to find a temporally coherent, low-distortion, injective mapping, which would assign each point of the 3D point cloud to a cell of the 2D grid. Maximizing temporal coherency and minimizing distance/angle distortions enable the video encoder to take full advantage of the spatiotemporal correlation of the point-cloud geometry and attribute signals.

A mapping between the point cloud and a regular 2D grid is then obtained by packing the projected patches in the patch-packing process. During the patch-packing process, it is possible to iteratively place patches in an image of size width × height. The patch-packing strategy may be different from encoder to encoder, and it can have a considerable impact on compression efficiency.

For example, an encoder may choose to exploit temporal correlation between patches. In such a scenario, patches with similar content could be placed in similar positions across time, generating a more temporally coherent video sequence.

Once patch packing is done, 2D images that represent the point-cloud geometry, its attributes, and occupancy (see details below) are generated, optionally scaled to a different resolution from their original (referred to as the *nominal*) resolution, and then compressed with any existing video codec.

The geometry image can be generated by inserting the depth values in the Luma channel of the image. However, a patch can have multiple points being projected to the same 2D pixel location. To handle such cases, the V-PCC standard allows the encoder to use up to 16 layers to store overlapping points.

Because the mapping between the point cloud and the 2D grid is not bijective, an extra binary image, referred to as the *occupancy map*, is needed to distinguish between the filled (i.e., associated with a point) and the empty (i.e., not associated with any point) cells of the grid. Similar to the geometry and attribute video sequences, the occupancy map video sequence can also be downscaled and then compressed using 2D-based image or video codecs.

All patch information that is required to reconstruct the 3D point cloud from the 2D geometry, attribute, and occupancy videos also needs to be compressed. Such information is encoded in the V-PCC patch sequence substream.

V-PCC introduces a new codec specifically optimized to handle this substream, which occupies a relatively small amount of the overall bitstream (i.e., lower than 5%). Additional information needed to synchronize and link the video and patch substreams is also signalled in the bitstream.

V-PCC can support multiple attributes per point. Supported attributes currently include texture, material ID, transparency, reflectance, normal, and user-defined attributes. Each attribute may have multiple instances (e.g., view-dependent texture). However, for most common applications, only texture/colour information is currently expected. Because the reconstructed geometry can be slightly different from the original one, the special transfers the colour from the original point cloud to the reconstructed point cloud when generating the attribute images.

The recolouring procedure considers the colour value of the nearest point from the original point cloud as well as a neighbourhood of points closer to the reconstructed point to determine the new colour value. Once the colour values are known, the special maps the colour from 3D to 2D using the same mapping applied to the geometry. A similar process could be used for other attribute types.

During the encoding process, the empty space between patches is filled by using a padding function, which generates a piecewise smooth image suited for video compression on geometry and attribute images. The V-PCC bitstream is then formed by concatenating the various encoded information (i.e., occupancy map, geometry, attribute, and patch sequence substreams) into a single stream. This is done by encapsulating these substreams into V-PCC data units, each consisting of a header and a payload.

The V-PCC unit header describes the V-PCC unit type. Currently, five different unit types are supported. The sequence parameter set (SPS) unit type describes the entire V-PCC bitstream and its subcomponents. The remaining unit types include the occupancy-video, geometry-video, attribute-video, and patch-sequence data units, which encapsulate the occupancy map, geometry, attribute, and patch sequence substreams, respectively.

Decoding process

The V-PCC decoding process is split into two phases: 1) the bitstream decoding process and 2) the reconstruction process. The bitstream decoding process takes as input the V-PCC compressed bitstream and outputs the decoded occupancy, geometry, and attribute 2D video frames, together with the patch information associated with every frame. The reconstruction process uses the patch information to convert the 2D video frames into a set of reconstructed 3D point-cloud frames.

Although the decoding process should be bit-exact and fully comply with the V-PCC specification, the same requirement does not apply to the reconstruction process. For instance, different implementations may choose to apply different up-sampling filters (i.e., filters for chroma up-sampling or for resampling to the nominal resolution) and include additional post-filtering and artefact-reduction methods after video decoding as well as advanced 3D filtering methods during the reconstruction process.

The reconstruction process requires the occupancy, geometry, and attribute video sequences to be resampled at the nominal 2D resolution specified in the SPS. The resampled videos are then used for the 3D reconstruction process, which consists of two main steps: 1) the geometry and attribute reconstruction and 2) the geometry and attribute smoothing.

The patch-packing process is constrained to guarantee no overlapping between patches. Furthermore, the bounding box of any patch, expressed in terms of $T \times T$ blocks, where T is the packing block size, should not overlap with any $T \times T$ block belonging to a previously encoded patch. Such constraints make it possible to determine, for each $T \times T$ block of the packing grid, the patch to which it belongs by analysing the 2D bounding boxes of all patches. The $T \times T$ blocks are then processed in parallel to generate the point-cloud geometry and attributes. For each cell of a $T \times T$ block, the corresponding pixel in the occupancy map is used to determine whether the cell is full or empty. If the cell is full, a 3D point is generated two different procedures, depending on the type of the patch.

V-PCC supports the concept of regular patches, which use the patch projection method described earlier. For regular patches, the 3D point Cartesian coordinates are computed by combining the depth information stored in the geometry image with the cell's 2D location, the patch's 3D offset, and the 2D projection plane. The attribute values associated with the reconstructed points are obtained by sampling the 2D attribute frames at the same grid location.

Because of the lossy nature of the video compression process, discontinuities may be introduced at patch boundaries in both the geometry and attribute signals. This may affect the visual quality of the reconstructed point cloud. Such artefacts could be alleviated by applying a smoothing process. Such a smoothing process consists of updating the position and attribute values of each boundary point with a weighted average of the position and attribute values of each point's nearest neighbors in 3D space.

Representations and Computational Algorithms in V-PCC

This document provides a description of the representations and computational algorithms and best practices for pre and post processing used in V-PCC. The Test model Category 2 v8 (TMC2v8) in MPEG describes the coding features that are under a coordinated test model study by 3DG as potential point cloud coding technology. The Test Model video-based point cloud compression (V-PCC) is a new project that was started after the Call for Proposals (CfP) for Point Cloud Coding [3]. The core encoding and decoding process for V-PCC were inherited from the solution that demonstrated the highest compression efficiency among all proponents as was agreed during the MPEG 119 meeting in Macau. The V-PCC solution is video codec agnostic, however, in the current testing procedure, HM [4] HEVC encoder implementation is used for video-based coding.

The ISO/IEC MPEG (JTC 1/SC 29/WG 11) group is studying the potential need for standardization of point cloud coding technology with a compression capability that significantly exceeds that of the current approaches and will target to create the standard. The group is working together on the exploration activity in a collaborative effort known as the 3 Dimensional Graphics Team (3DG) to evaluate compression technology designs proposed by their experts in this area. Nokia has been an active partner in this group.

Point Cloud Representation in V-PCC

Each point cloud frame represents a dataset of points within a 3D volumetric space that has unique coordinates and attributes. An example of a point cloud frame is shown in Figure 3-3.

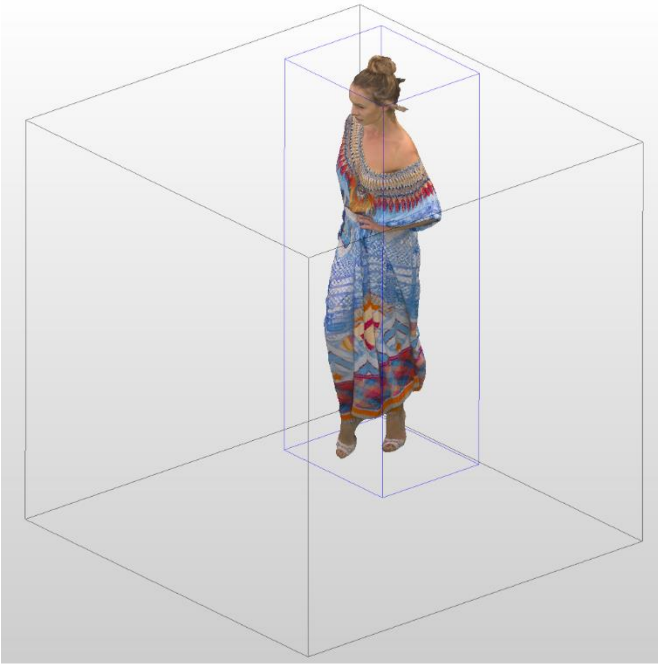


Figure 3-3. Point cloud sample image (1 frame).

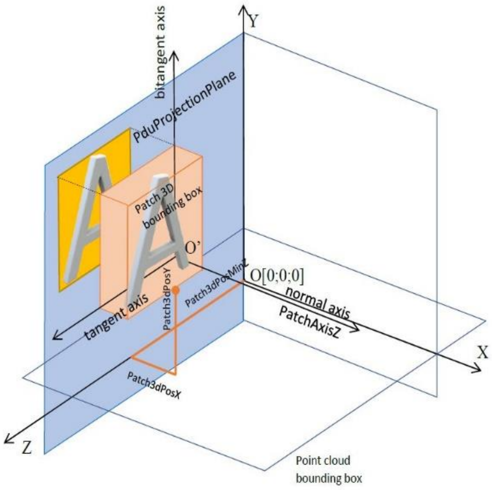
The reconstruction process for proposed pipeline elements is described on Figure 3-4 starting form the atlas information and adding the occupancy map, the geometry and the attributes information to the reconstruction process.

a) b) c) d)

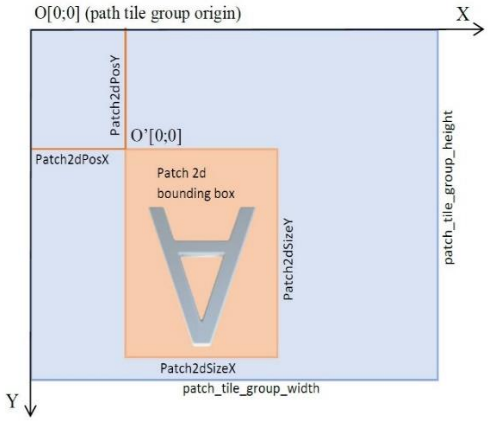
Figure 3-4. Point cloud reconstruction process: a – atlas; b – atlas and occupancy map; c – atlas, occupancy map and geometry, d – atlas, occupancy map, geometry and attribute.

Patch Description in V-PCC

The patch in the V-PCC notation is a collection of information that represents a 3d bounding box of the point cloud and associated geometry and attribute description along with the atlas information that is required to reconstruct the 3d point positions and their corresponding attributes from the 2d projections. The graphic representation of the patch is provided on Figure 3-5.



a) 3d point cloud domain



b) 2d projection domain

Figure 3-5. Patch description and associated patch information for atlas data.

Patch axis orientation (tangent, bitangent, normal axis) depends on the projection plane index (PduProjectionPlane), and patch projection mode. It should be noted that any side of the bounding box and additional 45 degree diagonal projections may be a projection plane. The origin of the patch bounding box is the nearest vertex to the point cloud coordinates origin point *O* (see Figure 3-5 a). The projection image is divided into tile groups. The origin point of the patch projection is the nearest point to the patch tile group origin point *O* (see Figure 3-5 b).

The patch information is generated per each point cloud frame unless the information is considered static, in this case the atlas information shall be generated only for the key (IRAP) pcc frames.

The remainder of this section is organized as follows: First subsection describes the patch generation and packing processes, which aim at determining how to best decompose the input point cloud into patches and how to most efficiently fit those patches into a rectangular 2D grid. Following subsection details the image generation and padding processes, which transform the point cloud geometry and texture information into temporally correlated, piecewise smooth, 2D images suited for coding using traditional video codecs. The processes of generating the auxiliary patch information and occupancy map are described in next coming subsections. The last subsection describes the smoothing module and the geometry and texture reconstruction processes.

Patch Generation and Packing

Leveraging traditional video codecs to encode point clouds requires mapping the input point cloud to a regular 2D grid. The objective is to find a temporally coherent low-distortion injective mapping that would assign each point of the 3D point cloud to a cell of the 2D grid.

Maximizing the temporal coherency and minimizing the distance/angle distortions enables the video encoder to take full advantage of the temporal and spatial correlations of the point cloud geometry and attributes signals. An injective mapping guarantees that all the input points are captured by the geometry and attributes images and could be reconstructed without loss. Simply projecting the point cloud on the faces of a cube or on the sphere does not guarantee lossless reconstruction due to auto-occlusions (i.e., auto-occluded points are not captured), and generates in practice significant distortions.

In order to avoid such limitations, V-PCC decomposes the input point cloud into a set of patches, which could be independently mapped, through a simple orthogonal projection, to a 2D grid without suffering from auto-occlusions nor requiring re-sampling of the point cloud geometry. Furthermore, the patch generation process aims at generating patches with smooth boundaries, while minimizing their number and the mapping distortions. In order to resolve this NP-hard optimization problem, V-PCC applies a heuristic segmentation approach that is described in Figure 3-6.

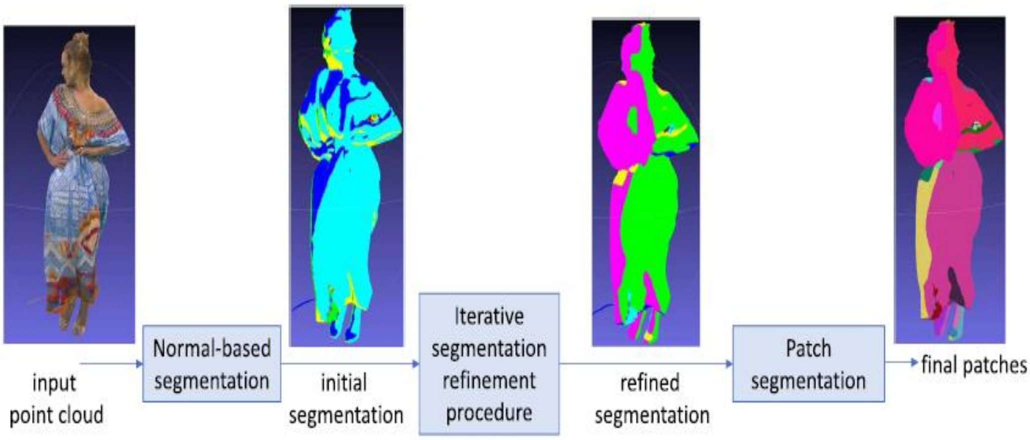


Figure 3-6: Overview of the V-PCC Patch Generation Process

First, the normal at every point is estimated as described in [5]. An initial clustering of the point cloud is then obtained by associating each point with one of the six-unit cube-oriented planes. More precisely, each point is associated with the plane that has the closest normal (i.e., maximizes the dot product of the point normal and the plane normal). The initial clustering is then refined by iteratively updating the cluster index associated with each point based on its normal and the cluster indexes of its nearest neighbours. The final step consists of extracting patches by applying a connected component extraction procedure.

The packing process aims at mapping the extracted patches onto a 2D grid, while trying to minimize the unused space and to guarantee that every $T \times T$ block (e.g., 16×16 block) of the grid is associated with a unique patch. V-PCC uses a simple packing strategy that iteratively tries to insert patches into a $W \times H$ grid. W and H are user defined parameters, which correspond to the resolution of the geometry/texture images that will be encoded. The patch location is determined through an exhaustive search that is performed in raster scan order. The first location that can guarantee an overlapping-free insertion of the patch is selected and the grid cells covered by the patch are marked as used. If no empty space in the current resolution image can fit a patch then the height H of the grid is temporarily doubled and the search is performed again. At the end of the process, *H* is reduced so as to account only for the used grid cells.

Image Generation & Padding

The image generation process exploits the 3D to 2D mapping computed during the packing process to store the geometry and texture of the point cloud as images. Figure 3-7 shows an example of generated geometry and texture images.

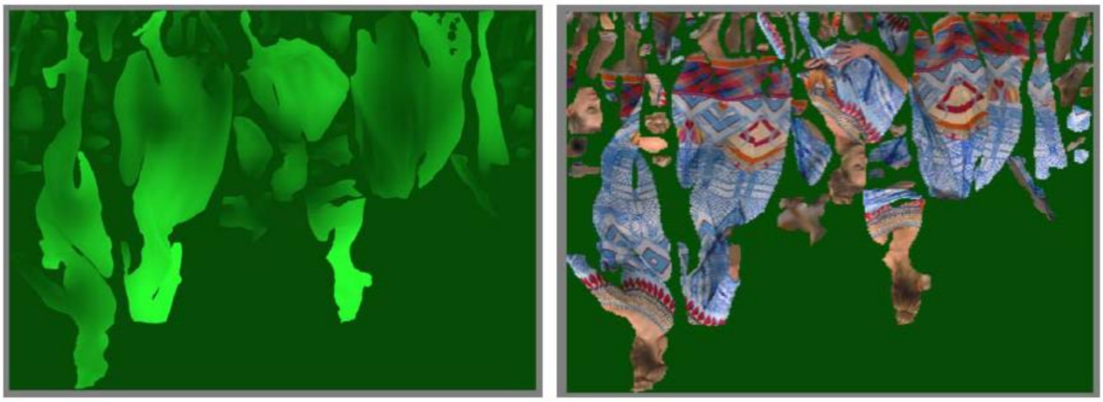


Figure 3-7: Example of geometry (left) and texture (right) images

In order to better handle the case of multiple points being projected to the same pixel, each patch is projected onto two images, referred to as layers. More precisely, let $H(u, v)$ be the set of points of the current patch that get projected to the same pixel (u, v). The first layer, also called the near layer, stores the point of $H(u, v)$ with the lowest depth $D0$. The second layer, referred to as the far layer, captures the point of $H(u, v)$ with the highest depth within the interval $[D0, D0 + \tau]$, where τ is a user-defined parameter that describes the surface thickness.

The padding process aims at filling the empty space between patches in an attempt to generate a piecewise smooth image that may be better suited for video coding. V-PCC uses a simple padding strategy, which processes each block of $T \times T$ pixels independently. If the block is empty (i.e., all its pixels belong to the empty space), then the pixels of the block are filled by copying either the last row or column of the previous $T \times T$ block in raster order. If the block is full (i.e., does not contain any empty pixels), nothing is done. If the block has both empty and filled pixels, then the empty pixels are iteratively filled with the average value of their non-empty neighbours.

Auxiliary Patch and Block Information Coding

In order for the decoder to be able to reconstruct the 3D point cloud from the geometry and texture images, the following patch/block metadata information is encoded in the bitstream:

- For each patch, the index of its projection plane, its 3D location, and its 2D bounding box.
- For each $T \times T$ block, the index of the patch to which it belongs.

The patch metadata is predicted and arithmetically encoded. The block to patch mapping information, is encoded as follows:

Let L be the ordered list of the indexes of the patches such that their 2D bounding box contains that block. The order in the list is the same as the order used to encode the 2D bounding boxes. L is called the list of candidate patches. The empty space between patches is considered as a patch and is assigned the special index 0. This patch is also added to the candidate patches list of all the blocks. Let I be the index of the patch to which the current $T \times T$ block belongs to and let J be the position of I in L. Instead of explicitly encoding the index I, its position J is arithmetically encoded. This can lead to better coding efficiency.

Smoothing and Geometry/Texture Reconstruction

The smoothing procedure aims at alleviating potential discontinuities that may arise at the patch boundaries due to compression artefacts. The implemented approach moves boundary points to the centroid of their nearest neighbours. The point cloud geometry reconstruction process exploits the occupancy map information in order to detect the non-empty pixels in the geometry/texture images/layers.

The 3D positions of the points associated with those pixels are computed by leveraging the auxiliary block/patch information and the geometry images. More precisely, let P be the point associated with the pixel (u, v), let $(d0, s0, r0)$ be the 3D location of the patch to which it belongs, and let $(u0, v0, u1, v1)$ be its 2D bounding box.

P could be expressed in terms of depth $d(u, v)$, tangential shift $s(u, v)$, and bi-tangential shift $r(u, v)$ as follows:

- $d(u, v) = d0 + g(u, v)$
- $s(u, v) = s0 - u0 + u$
- $r(u, v) = r0 - v0 + v$

where $g(u, v)$ is the luma component of the geometry image.

V-PCC Performance Evaluation

Figure 3-8 a-c shows the results of the subjective evaluation for three dynamic sequences compressed with V-PCC, against the anchor at different bitrate points. During the subjective evaluation, uncompressed point clouds were shown as hidden reference, thus the bitrates shown for “uncompressed” in Figure 3-8 do not represent actual bitrates, but the respective target bitrate point of the test point.

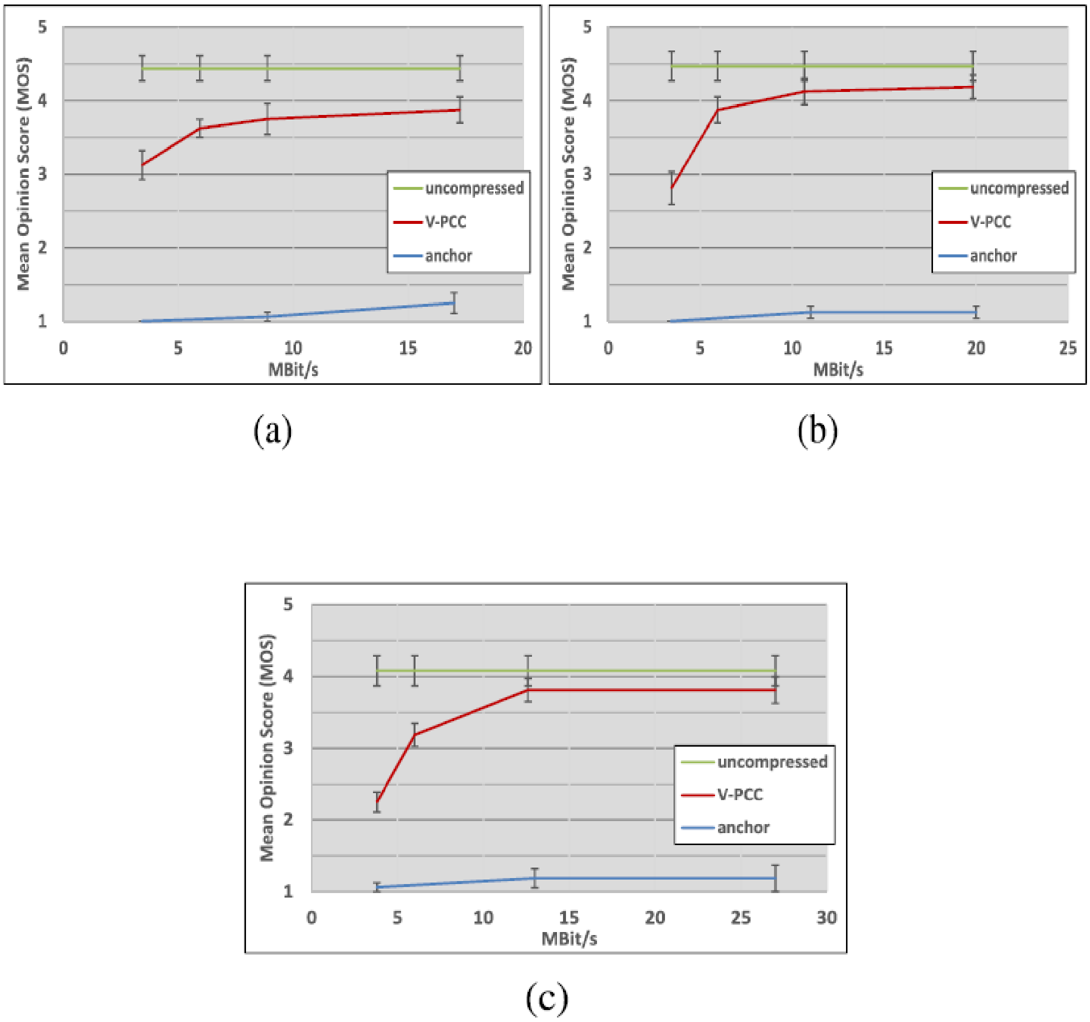


Figure 3-8: Subjective evaluation V-PCC results for RedandBlack (a), Soldier (b), and Longdress (c).

The benefits of V-PCC over the anchor in terms of visual quality are clearly visible and in line with the objective evaluation results, e.g. as shown in Figure 3-9: Even at the lowest target point, reasonable quality was achieved, and already at the third target point the achieved quality was close to the uncompressed data. Depending on the sequence, this means compression factors between 1:100 to 1:500 are feasible. Thus, this approach was selected as the basis for the test model for this category.

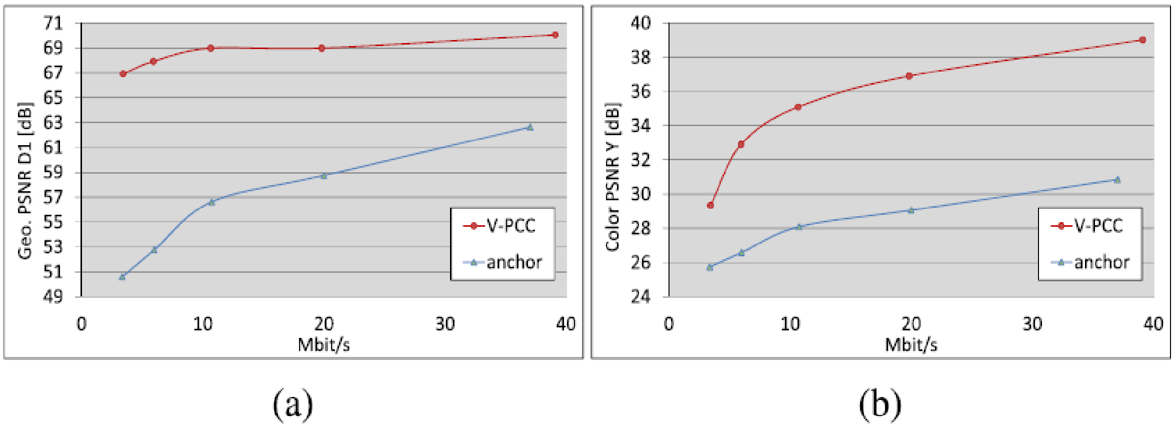


Figure 3-9: Objective metric RD-curves for sequence Soldier.

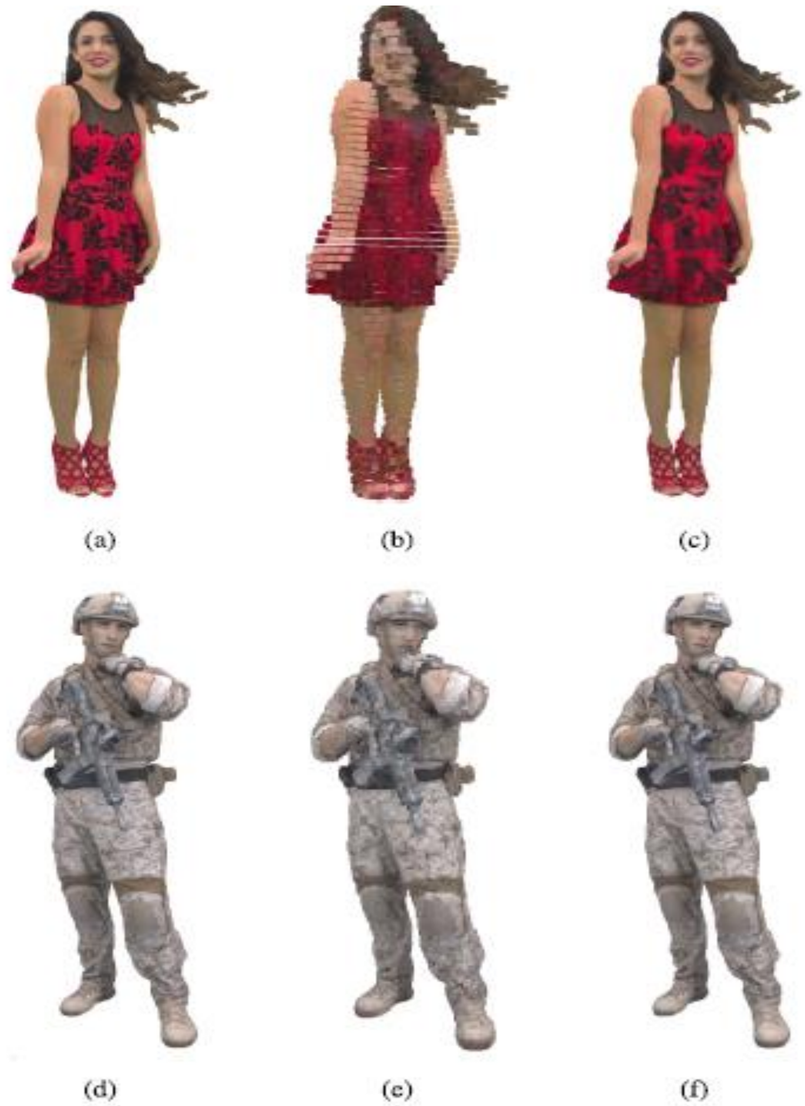


Figure 3-10: Original (uncompressed) and reconstructed point clouds for sequences RedAndBlack at 3.5 MBit/s (top), and Soldier at 11 MBit/s (bottom) (a) Original (b) Anchor (c) V-PCC (d) Original (e) Anchor (f) V-PCC.

Standardization Status of V-PCC

As ISO/IEC 23090-5 is closing in on to its publication as an international standard, this first V-PCC implementation is an important asset to prove its relevance to the public. The V-PCC AR playback application source code has been made available to the public and will be further developed as the V-PCC standard has reaches codec stability.

In the meantime, activities on efficient storage and streaming of V-PCC data have started. Again, these are benefiting from the underlying 2D video coding structure and utilising existing 2D video coding infrastructure solutions. For example, it is feasible to store each V-PCC video track as a separate track in the ISO base media file format (ISOBMFF). The auxiliary patch information is stored as timed metadata track and combined with the video tracks as a V-PCC GroupListBox. This box provides the list of all tracks of the V-PCC content. Thus, a flexible V-PCC content configuration supporting a variety of client capabilities, e.g. multiple versions of encoded data components, can be stored in a file. Such content configurations can also be stored in an MPEG-DASH manifest (MPD) for dynamic adaptive streaming of V-PCC data over current video delivery infrastructure.

3.1.1 Reference Architecture

The reference architecture for the final demonstrator using DSL is under planning and design. Nokia is participating in MPEG standardisation work using the learnings of the current version of the demonstrator. The standardisation work status and the maturity of the DSL tool-set are impacting the time-schedule of DSL reference architecture design and studies.

3.1.2 M24 Partial Demonstrator

Current Nokia's V-PCC player can decode and render MPEG V-PCC coded bit streams in real-time. An early version of Nokia's V-PCC application was presented as VIP internal Nokia demo at the Mobile World Congress 2019 in Barcelona. The current version has been showed at the International Broadcasting Conference (IBC2019, September 13-17, 2019) in Amsterdam. The V-PCC playback application source code has been made available to the public as reference: Nokia Technologies, "Video Point Cloud Coding (V-PCC) AR Demo," https://github.com/nokiatech/vpcc. This source code includes architecture and all features of M24 partial demonstrator.

3.1.2.1 M24 to M36 GAP

V-PCC is expected to be delivered as an international standard ISO/IEC 23090-5 in mid 2020, with the accompanying technologies for storage and streaming (ISO/IEC 23090-10) shortly after. This real-time V-PCC system intends to promote the standard and highlight its importance for next-generation immersive media applications. We expect a follow-up application demonstrating real-time V-PCC AR streaming before the finalisation of the respective standards.

3.2 Use Case 2 in FitOptiVis

Demonstrated Technology	WP in FitOptiVis	Role in Demonstrator	Date for demonstration (M24/M36)
Reference architecture planning and designs	WP2	Provide the reference architecture of the demonstrator	M36
Parallelization of selected V-PPC algorithms	WP3	Speed up the compression methods	M36
Real-Time monitoring of the whole V-PCC system	WP4	System level optimizations	M36

3.3 Use Case 2 Metrics

3.3.1 User Needs

User Need	Validation method	Status
Efficient 3D scene description	Demonstration. Compare performance parameters with previous generation. Evaluation by image quality experts and end users	Partial: End users have evaluated initial results and given the feedback for development persons.
Efficient compression of point cloud of the 3D object	Demonstration. Compare performance parameters with previous generation.	Pending
Efficient delivery/streaming of point cloud data	Demonstration. Compare performance parameters with previous generation.	Pending
Efficient rendering of static and dynamic point clouds on power-limited mobile device	Demonstration. Ask the user for a score (1-5) how well rendering is controlled. 3 or higher is considered Passed.	Pending
User interaction	Demonstration. Review by User Interaction elements. Ask the user for a score (1-5) how well user interactions are controlled. 3 or higher is considered Passed.	Partial: End users have evaluated initial results and given the feedback for development persons.

3.3.2 Use Case Requirements

ID	Use Case Requirement	Verification method	Status
A1	Mobile phone with wireless network connection with at least 8 MBit/s bandwidth.	Check that predicted bandwidth is available to the user	Partial: Initial test done.
A2	Mobile phone with storage capacity of at least 1 GB.	Check that predicted storage capacity level is available to the user	Partial: tested via trials.
A3	Mobile phone supporting rendering of at least 1 Million points per frame at 30 fps	Check that the accepted rendering processing level is available to the user	Pending
B1	Pinch-to-zoom interaction	Check that the predicted zoom interaction can be executed in sufficient level for the user	Pending
B2	Object rotation and translation interaction	Check that the predicted rotation and and translation interactions can be executed in sufficient level for the user	Partial: tested via trials.

ID	Use Case Requirement	Verification method	Status
B3	Collaboration feedback	Ask end user feedbacks for selected cases	Pending
F1	Encode and encapsulate volumetric video object	Check that identified encoding systems and parameters are provided for the application	Pending
F2	Receive and decode encapsulated volumetric video object	Check that identified decoding system parameters can be provided for end-user application	Pending
F3	Visualize decoded volumetric video object	Check that system will adapt to user's needs in visualization phase	Pending
F4	Record user interactions and transmit to collaborator	Measure difference between predicted and actual for selected interaction case	Pending
NF1	System should be capable of switching between inputs	Measure difference between predicted and actual for selected test cases	Pending
P1	Real-time playback of ~1M points at 30fps	Measure difference between predicted and actual for selected playbacks	Pending
P2	Latency of max 1s for collaboration feedback	Measure difference between predicted and actual latencies for selected playbacks	Pending

3.4 End user feedback and recommendations

During the Y1 end-user board meeting held in Eindhoven on September 10th, 2019 the following topics were discussed. We include our comments as of the writing of this document.

End User Board's Comments	Comments/Actions/Views/Reactions
Latency requirement should be better defined, since it is expected to be a stringent constraint.	Nokia is working the latency requirements at a very detailed level in this Use Case.
Is there any reconfigurability among edge and cloud? What does it need? How is it exploited? Is it affordable (see latency constraints above)?	Nokia is studying the edge and cloud solutions, but they are not focus areas in this Use Case.
Are we missing something? I think Data fusion approaches should be improved.	Data fusion techniques can be used in the selected areas and use cases.

End User Board's Comments	Comments/Actions/Views/Reactions
Which should be relevant for the FitOptiVis VR case? Use of holographic technology for immersive experiencing	Nokia has not studied the holographic technology for immersive use cases.
Buzz potentials? Telesurgery in medicine	Yes. This is very potential area but needs a lot of investments and new competencies.
New business models areas inside VR use case? Which models are relevant for the FitOptiVis VR case? Pay per use	Pay per use is one potential buzz model.
One of the needs we are missing it is a "real-time" response just for those situations related to virtual assistance character, for instance, where a real-time response is. Of course, in the needs related to "efficiency" and improved UX is implicitly a quick response of the system, but does that imply RT?	Nokia will consider different aspects of real-time responses since we are doing the standardization activities in this domain.
New buzz potentials? Education. The education is going increasingly towards an online approach and remote assistance. VR could be a key for developing technical and social skills thought virtual scenarios. One of the important points in the classroom education is the social factor and the need of interacting with the classmates or teachers. In the educational online approaches, that is missing, but it could be used a strong virtual environment for recreating it. Also, this could be used as therapy for certain social or mental disorders in a studied way depending of the needs or degree of the patient.	Nokia has identified the education domain as potential business area.
Probably part of the efficient delivery, but I'm missing the latency here. For example, when using it for training in a situation where hand-eye coordination is important. Mobile devices are not only power limited, but also speed (various generations of mobile devices).	The latency challenges will be studied in this Use Case in the detail mode.

4 Use Case 3 - Habit Tracking

4.1 Full Demonstrator Description

4.1.1 Introduction

The demonstrator of the UC3 will show how the system is able to identify the activities carried out at home. There will be a camera in each room. As a result, the information will be coded and processed accordingly, so that the caregiver can have a clear view of the status/performance of the monitored person.

This information will serve to take measures accordingly to change the life style and thus improve the quality of life. Expected TRL = 5/6.

This is done with cameras, telemedicine modules and other devices connected to the network through TSN.

The images collected by the cameras will be analysed in real time using neural networks capable of:
1. Inferring actions carried out by the monitored person.
2. Facial recognition.

Telemedicine modules and other possible devices will share the same network to send data. Under no circumstance will images be sent outside the user's local network in order to preserve the privacy of sensitive data.

Figure 4.1 shows the architecture for the UC3 - Habit Tracking with its different modules:

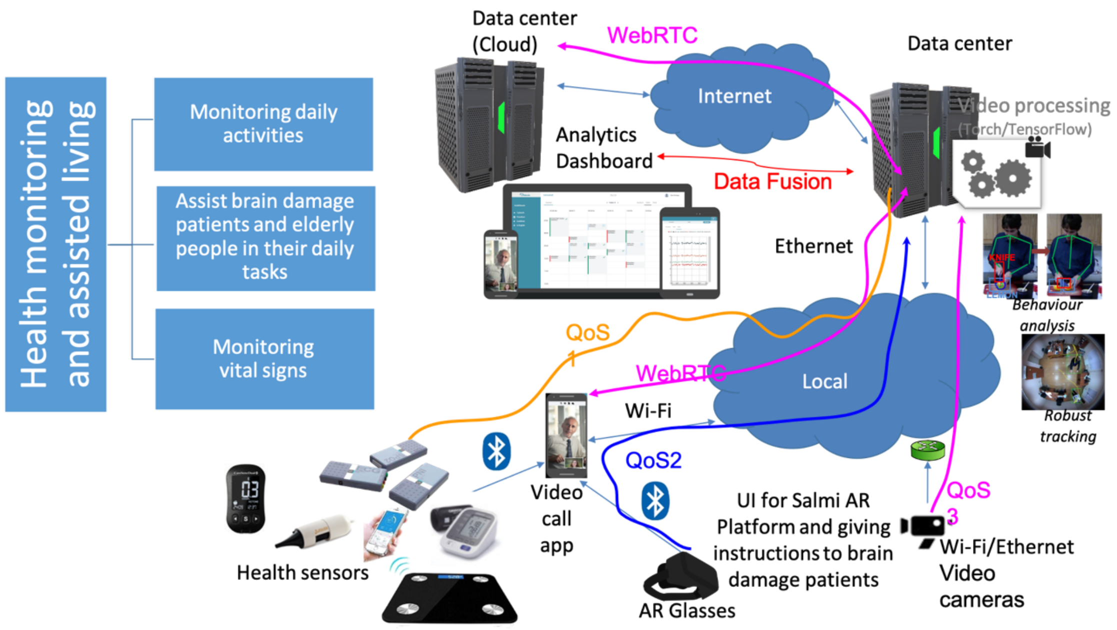


Figure 4-1 UC3 - Architecture

Regarding the action recognizer, this module will monitor the actions performed by the person using as input the video stream from a camera, in an indoor scenario. The camera sends images to our edge processing node, the NVidia Jetson Xavier. This node performs the analysis of the video feed to infer the action, and it outputs a label for the action with a likelihood that represents the confidence of the action recognition. This platform is connected through the TSN network to a PC that will be our centralized cloud processing node. It will perform tasks such as receiving and aggregating alarms, integrating and analysing information to make decisions (making use of the FIVIS platform), or sending reconfiguration commands to the edge processing nodes through the pocl-remote framework.

Time Sensitive Networking (TSN) provides Ethernet connectivity between the different devices participating on the demonstrator. Furthermore, TSN can protect sensitive streaming data (i.e. alarms, control, reconfiguration commands, and biomedical signals) from congestion loss by isolating them with respect from best-effort traffic, such as video streaming or non-critical monitoring. Time synchronization is provided to enable accurate and coherent monitoring of distributed processing applications, as well as the coordination between the heterogeneous systems participating in the demonstrator.

4.1.2 Infrastructure

Habit Tracking - Action Recognizer:

Name	Description
Jetson AGX Xavier	System on Module. Camera control. Edge processing. Human action recognition. Reconfigurable node for pocl-remote integration.
Manta G-125C	CCTV IP camera.
PC Ubuntu 18.04	Cloud processing. Aggregation of local data from edge nodes. Decision making. Communication and integration with the other nodes in the UC. Visualization, communication with FIVIS, and data logging.

Habit Tracking - Facial Recognizer:

Name	Description
nVidia Jetson Nano	Single board computer, extraction of faces, extraction of facial features, classification of facial features, and management of energy profiles.
FOSCAM FI9800P	CCTV IP camera.
Raspberry Pi camera	Test camera connected to the board with MIPI interface
Geekworm NVidia Jetson Nano 18650 UPS Hat + 4 18650 LiOn 3,7V batteries	External powering battery for NVidia Jetson Nano. Communication to the board by means of power wire and i2c communications line (battery monitoring).

Habit Tracking – Telemedicine Modules:

Name	Description
ECG, SpO2 and NIBP Module	Three independent modules that can be used to monitor the user's vital signs.
Android Mobile	Mobile that connects to the modules with Bluetooth BLE and sends the data to a web platform

Habit Tracking - Time Sensitive Networking

Name	Description
Custom Board	Zynq-7030 based platform Four TSN interfaces Standard conformance: IEEE 802.3, IEEE 802.1Q, IEEE 802.1Qbv
TL-WR802N	IEEE 802.3 - IEEE 802.11n gateway

4.1.3 Deployment

The final demonstrator shows the integration of the different parts: action recognition, facial identification, vital signs, and Time Sensitive Network communications.
The demonstrator includes: at least 2 edge nodes for the vision-based subsystems (Jetson TX2 nano and Jetson Xavier platforms) attached to at least 2 cameras to perform the facial identification and the action recognition tasks, the time sensitive communication network that includes a topology with at least 2-3 bridges (2 for the vision nodes and an additional one for the vital sign traffic) plus an additional connection to the server that centralized all the information and decision making, and every other useful information from other modules connected into the system, for example, ECG, SpO2 and/or NIBP signals from RGB telemedicine modules.
This Vital Signs monitoring solution will be used for the measurement of four parameters (Heart Rate, ECG, SpO2 and NIBP) and will research on a new concept for a non-invasive and ergonomic new approach to measure the cardiac output values and trends. It will provide a highly portable yet transparent solution for the patient.

The monitoring system allows recognize actions in a continuous way. It sends data to the Jetson Xavier platform to detect interesting or potentially risky actions, and logs the information sending it to the server, to update the information about the person lifestyle. A signal from the cloud centralized server is used to start the communication and it runs until a stop signal is received. Also, the centralized cloud server will send monitoring and commands to do the reconfiguration using the pocl-remote framework provided by TAU. Also, we are already monitoring qualities such as performance, energy consumption, working frequency of our edge processing nodes, and the action labels sending the information to the FIVIS monitoring platform provided by CUNI.

4.1.4 Test plan (scenarios)

d.1) A person is sitting on a chair / sofa, then this person stands up and starts walking. While he is walking, he falls down and keeps lying on the floor. In this scenario, the action recognizer module must analyze the video feed and in a successful scenario shows probabilities with relatively high confidence values for the action classes (*stand up, walk, fall down, lying on the floor*). When critical actions such these are detected, an alarm is triggered and sent through TSN to a centralized cloud server.

Test Name	**UC3-001**
Description	Detection of falls in an indoor scenario with the action recognition module
Input Data	Video feed
Expected Output	An alarm is triggered when the person falls down.

Pass/fail criteria	
FAIL	Alarm not triggered when the fall occurred.
PASS	Actions performed on video are successfully recognized (at least in the top 3 actions with higher confidence). The alarm is properly received by the cloud server..

d.2) A person is walking around the room, and subsequently the subject starts swiping the floor (Figure 4-2). In this scenario, we are also testing reconfiguration. Thus, while this happens, the average power consumption of the GPU for the last minute is higher than a threshold in Watts. The information about power consumption is being monitored and consequently, reconfiguration is triggered the neural network model for action recognition changes to a simpler one that consumes less energy, and we reduce the operating frequency of the GPU and CPU of our edge processing node.

Test Name	**UC3-002**
Description	Reconfiguration when high power consumption is detected.
Input Data	Video feed.
Expected Output	When no significant change happens in the action, reconfiguration to save energy is triggered. Operating frequency of the GPU and CPU of the edge processing nodes are reduced. Neural network model for action recognition is changed to reduce power consumption and resource demand. Action is still monitored but energy consumption is reduced.

Pass/fail criteria	
FAIL	Average power consumption is higher than X Watts (threshold) and reconfiguration is not triggered in less than a minute.
PASS	Average power consumption is higher than X Watts (threshold), and reconfiguration is triggered in less than a minute.

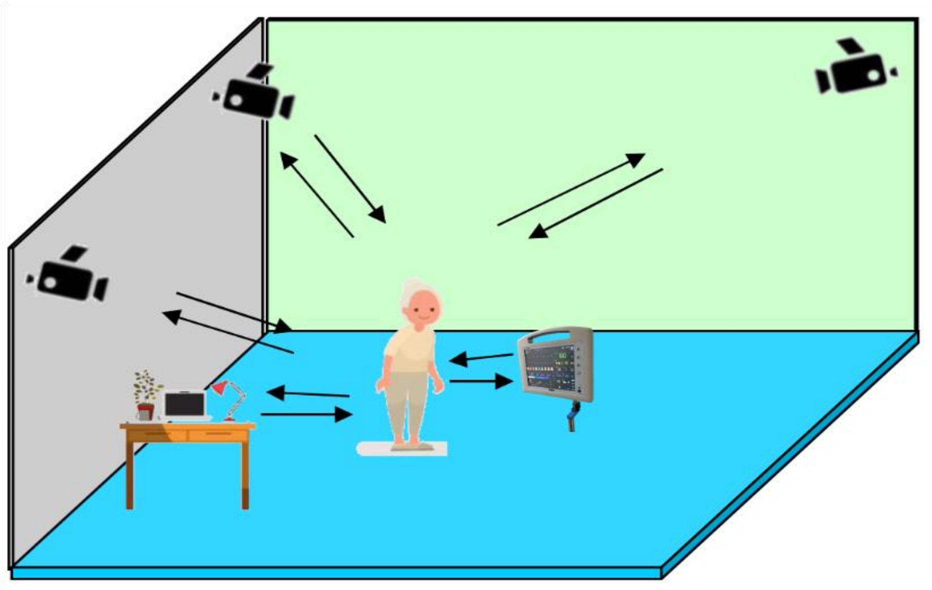


Figure 4-2: Person walking around the room being monitored

4.2 Reference architecture

The next diagram (Figure 4-3) shows the reference architecture for the System for Surveillance of a Smart Grid Critical Infrastructure. The complete QRML DSL description can be found in the appendix.

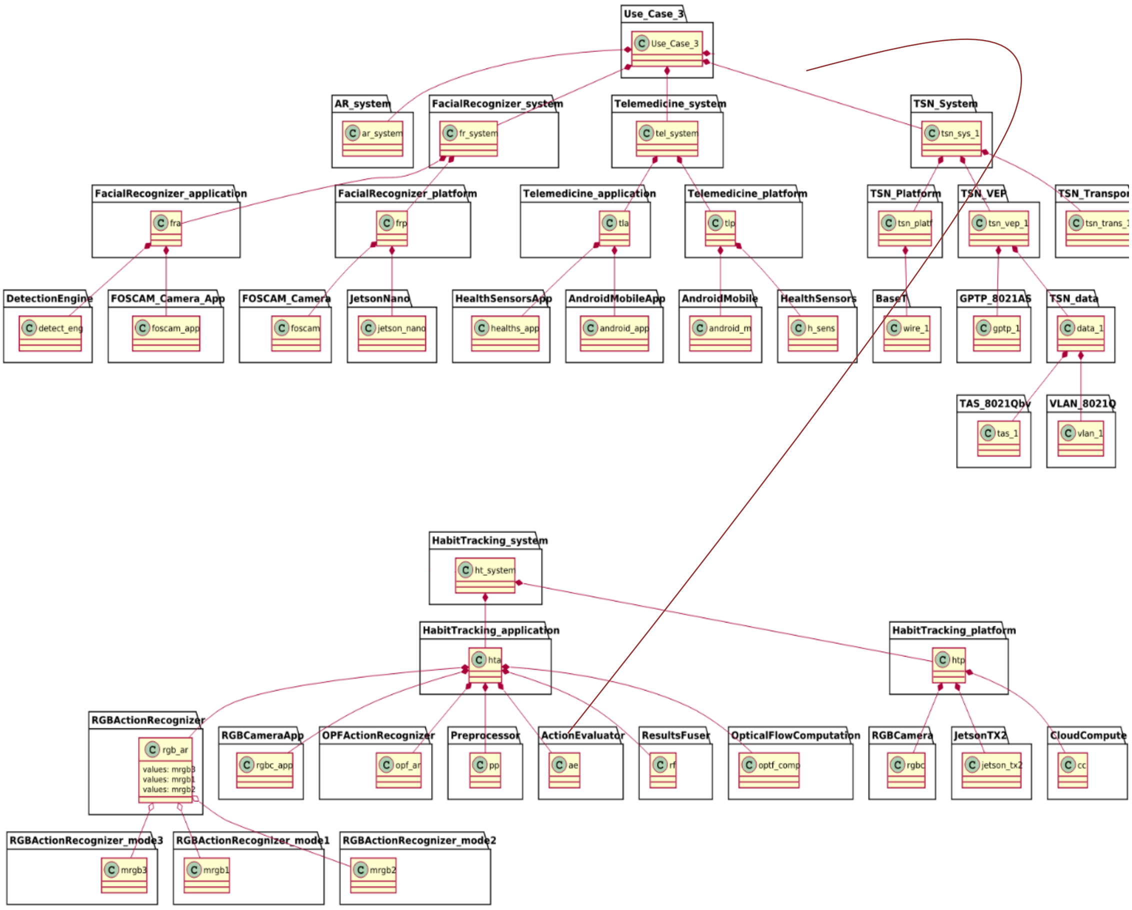


Figure 4-3: reference architecture for the System for Surveillance of a Smart Grid Critical Infrastructure

4.3 M24 Partial Demonstrator

In the M24 demonstration, both cameras and devices will run through the network, recollecting data for face recognition, inferring actions carried out by the monitored person, vital signs, etc. All will work on the same network.

Figure 4-4 Shows a person walking around the room being monitored. Example of M24 demonstration.

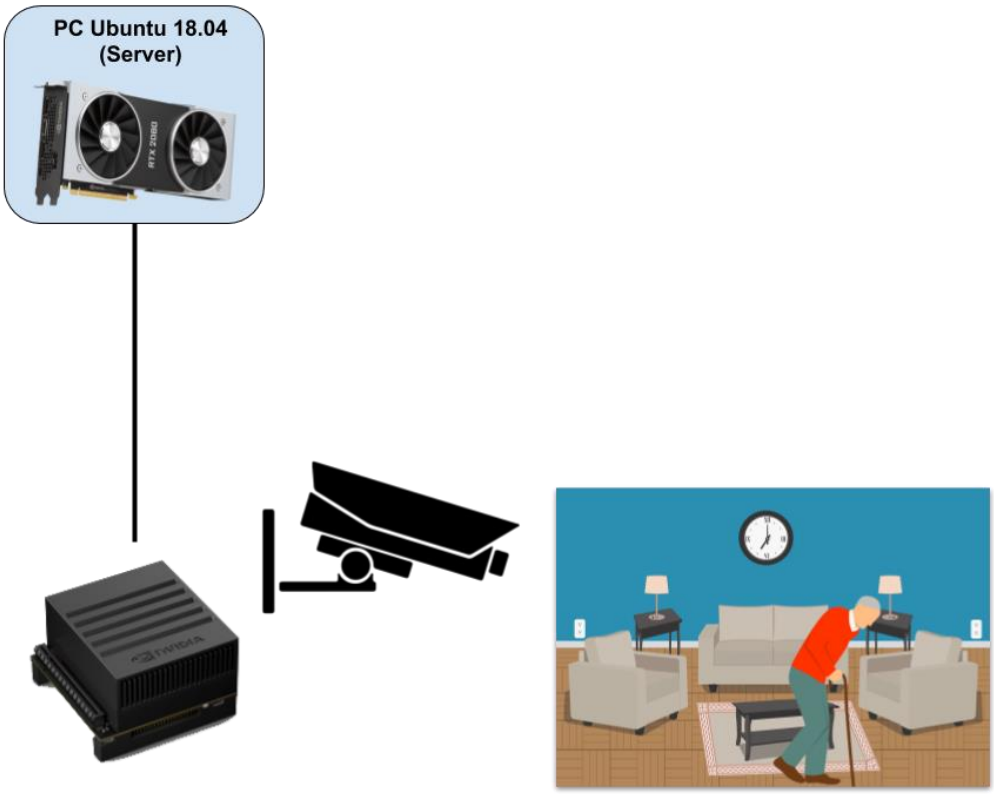


Figure 4-4: Example of M24 demonstration

A part of the partial demonstrator is used to show that the facial recognition system (and potentially other computer vision processes) can be adaptive to the power situation of the system. The facial recognition board is connected to a powering system with batteries and with monitoring of the battery level. By integrating this with the runtime and using Jetson command line functions to throttle the clock speed and maximum power draw from the batteries, we can fine tune the energy consumption of the system. This is fully explained in D4.2.

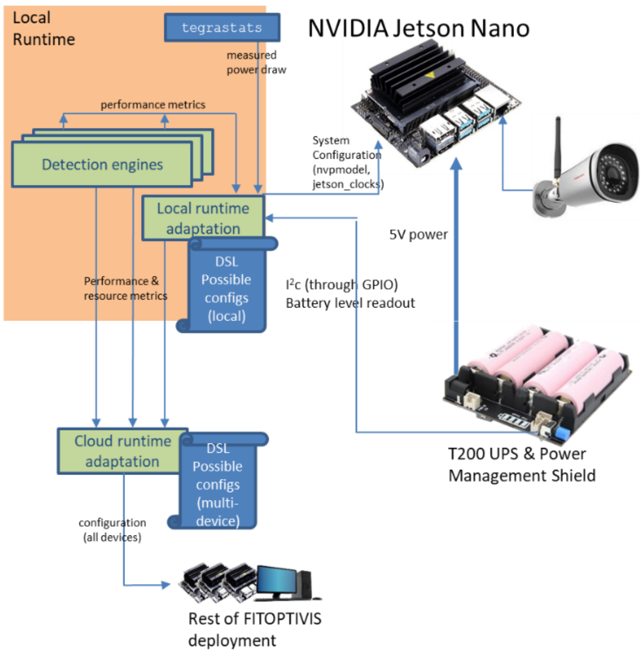


Figure 4-5. Overview of nVidia Nano edge device setup for the partial demonstrator for M24

The action recognition module for the demonstrator will use the Jetson Xavier edge processing node to perform inference. It will demonstrate its operation and integration with the monitoring platform FIVIS and the use of reconfiguration (although not the pocl-remote configuration already mentioned). The edge processing node will be connected to the server and show its integration and how the deployment for this part works: signals to start monitoring, to stop the system, and commands to do the reconfiguration of the edge processing nodes.

4.4 M24 to M36 GAP

In M24 demonstrator we are not showing integration between the use case partners. In the next figure, we can see a diagram of the final demonstrator where some of the components are connected by the TSN network.
Also, for the final demonstrator, the action recognition part will show reconfiguration using the pocl-remote framework provided by TAU.

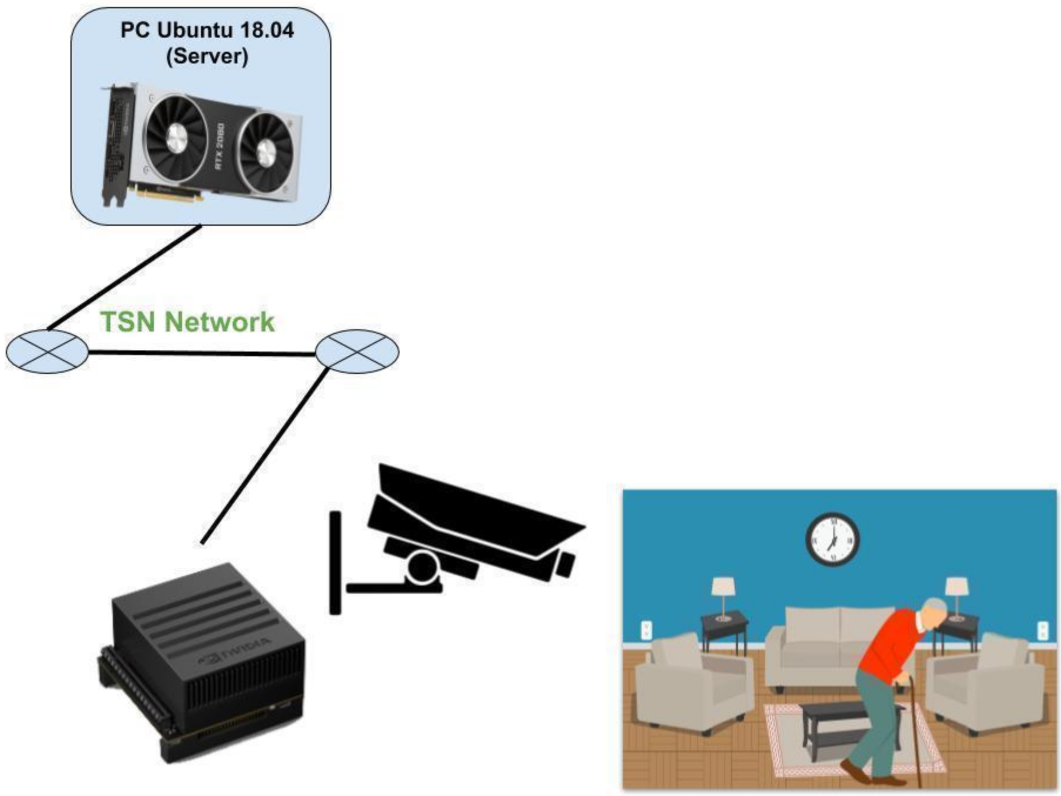


Figure 4-6. Integration of TSN Network in the use case demonstrator, showing Nvidia Xavier edge local node

We must decide the final tool to show all the information collected from the cameras and modules and be able to show alarms and events through that tool. We will have an output similar to Figure 4-7:

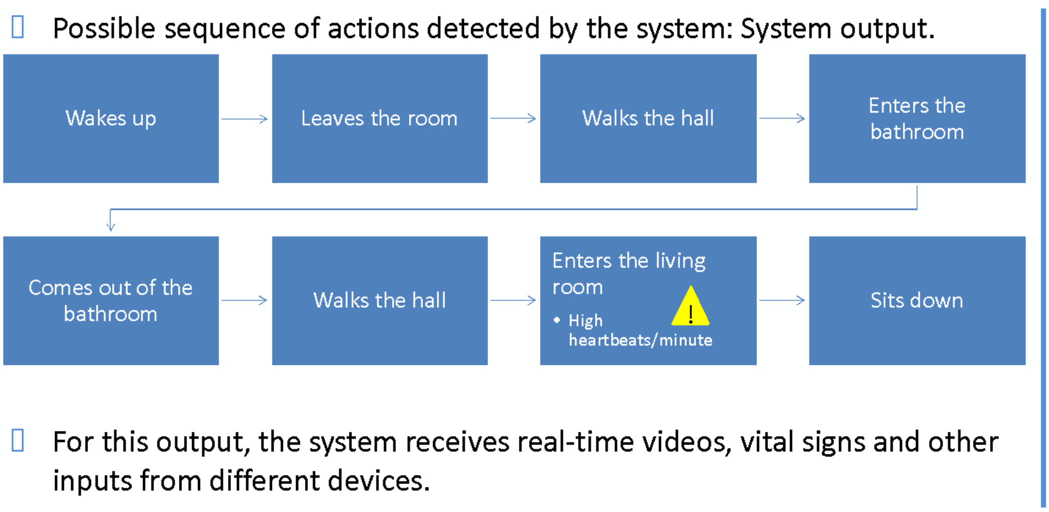


Figure 4-7: Final tool possible output

4.5 Use Case 3 in FitOptiVis

Demonstrated Technology	WP in FitOptiVis	Role in Demonstrator	Date for demonstration (M24/M36)
Runtime monitoring	WP4	The performance of the system is monitored to adequate the performance to the current energy level. Also, using the FIVIS monitoring platform	M24
Resources management		Energy usage in the facial recognition is throttled by monitoring and adapting performance.	M24
Runtime reconfiguration	WP4	Reconfiguration through the pocl-remote framework	M36
Real-time processing using accelerators	WP5	Image-video processing in real time	M24
Use of components: Behaviour Classifier (UGR) Pose Estimation Engine (Hilberia) Time Sensitive Networking (TSN)	WP5	Demonstrator building blocks	M36

4.6 Use Case 3 metrics

4.6.1 User Needs

User Need	Validation method	Comment
Run-time support: on-line multi-objective optimization a) The active vision mechanisms ensure the robustness of the tracking system and the reduction of data bandwidth. b) Deep Learning vision systems that can use online reconfiguration c) Development of techniques to efficiently map heterogeneous application descriptions into various heterogeneous platforms,	Development environment living lab: Sensorized room (Partner lab) where the developers test the progress in terms of Quantitative and Qualitative characteristics.	Partial. Independent subsystems working separately
Energy-efficient, high-performance, smart devices and components. a) Development of software libraries for the person tracking and the analysis of the user behavior.	Development environment living lab: Sensorized room (Partner lab) where the developers test the progress in terms of Quantitative and Qualitative characteristics.	Partial. Energy monitoring for the video-processing systems.

b) Development and selection of SW/HW components and configurations suitable for optimal energy and performance use		
a) Proving the validity/performance and assessing the applicability of its Map4Me demonstrator into real-world industrial needs by integrating results from earlier WPs into MVP-level demonstrator for Habit Tracking and AR/VR use cases bearing in mind safety, security, and b) Exchanging information/knowledge inside/outside the project consortium and investigating/supporting the exploitation of project results	Development environment living lab: Sensorized room (Partner lab) where the developers test the progress in terms of Quantitative and Qualitative characteristics.	Pending.

4.6.2 Use Case Requirements

ID	Use Case Requirements	Verification Method	Comment
A1	The system must be distributed with 4 principal elements: LAN Cameras, Processing unit, Smart phone and Other devices.	Check that all elements are available.	Partial. Different independent components available
A2	Other devices with information that may be important depending on the user (detection of vital signs, AR Glasses, etc.).	Check if there are other devices available.	Pending
A3	The system must connect all devices in the same LAN (WiFi/Ethernet).	Check that all devices are in the same network	Pending.
A4	Low power execution board(s) must be installed to perform image processing.	Measure Power consumption. Compare power consumption according to the architecture profiles	Verified. Power consumption monitored from the components based on Nvidia Xavier and Nvidia Nano
B1	Notify any alert situation. The doctor/relative must be able to see what happened	Check behaviour of the system when an alarm is triggered: check	Verified. Use case scenario to be demonstrated. Also, accuracy of the detection to be

	and if it is a real alert situation and not a false alert.	that video of the potentially risky action is sent	included (using dataset of actions such as people falls)
B2	Store all the data collected by the devices and the output of the recognition of actions and video analysis in the cloud to show the user's status	Verify the correct storage of all the data shown in logging files	Pending
F1	Detection of user actions.	Action recognition performed and validated using accuracy values	Verified. Action classification accuracy measures
F2	Minimization of used resources (i.e., energy).	Monitor energy consumption	Verified. Energy monitoring consumption for different component alternatives and through the monitoring platform FIVIS
F3	Minimization of data transmission. Only metadata about status information/Alert situation/Medical information must be sent. Actual data (images, etc.) must be minimized.	TB Decided	Pending
F4	Collection of alerts in the house.	Show logging of the actions (monitored labels along time)	Partial. Real-time demonstration of detection of actions with confidence levels. Also, log of all this actions over time
F5	Track users in their own home.	Validate via accuracy of tracker	Pending.
P1	Real-time image processing ~ 20 fps.	Calculation and tracking of the figure, defined as number of analysed frames divided by seconds elapsed.	Pending.
P2	Real-time event recognition (e.g., human-real-time, less than 500ms).	Measure performance of action recognition (inference)	Verified. Performance measurement of action inference in frames per second

NF1	Protection of privacy (no transmission of images).	Check that only data (not images) are sent to the server (encrypted if necessary)	Pending.
NF2	Alerts sent with priority.	Coherence between user provided configuration and provided QoS (i.e. no latency violation). Measure synchronization accuracy between time aware stations in the network Alerts transported over prioritized VLAN frames with bounded latency. Coordination between talker and listener nodes: Synchronization and/or timestamps	Pending.
NF3	Video processing data and Vital signs data transmitted by the same network.	Communication capabilities between the different subsystems and the sensors/network components	Pending.

4.7 End User feedback and recommendations

During the Y1 end-user board meeting held in Eindhoven on September 10th 2019, the following topics were discussed. We include our comments as of the writing of this document:

Comment by the board	**Discussion by UC3 participants**
Is there any video recording? That may be particularly relevant to understand the cause of certain events like falls. You should consider storage of partial frames to allow predictions.	A window of time surrounding events of interest is defined and information before and after the event is recorded for human inspection and determination of causality. The value of this window (how many seconds around the event should be stored) is still to be determined. Automatic estimation of causality is a research topic and could be implemented in commercial offerings based on this, but is out of scope in FITOPTIVIS.
Privacy matters → allow users to request for temporary disconnections.	Privacy is not within focus in FITOPTIVIS, but for a commercial system such guarantees of privacy requests would be implemented.
WRT this scenario: from the descriptions reconfigurability is not clear at all.	Hardware reconfigurability is not part of the intended design for this use case. In response to this greater emphasis has been put during Y2 on the software adaptability of the modules (e.g., UGR trigger different detection networks based on context, HIB use different power consumption profiles in the target hardware).

Our expectation and possible questions for the Y2 meeting are as follows:

- Devices in the patient's home: How important would energy consumption be? What's the expected cost for the monitoring of one patient?
- Performance: How accurate should the synchronization between video feeds and sensor signals be?
- Usability: How intrusive would be the use of AR devices for patients in different age groups? What would be the expectations for a system dashboard for the caregivers (this question should be done along with maybe a hands-on study on a mockup of said dashboard)? Would end-users be willing to perform training to detect their facial features?

5 Use Case 4 - 3D industrial inspection (ITI)

5.1 Full Demonstrator Description

Many manufacturing industries still have open problems regarding quality control (QC) in terms of costs and productivity. Sometimes manufacturers are bound to manual inspect the production. This is highly time consuming and increases the production costs. Current 3D QC scanners are too slow and therefore not suitable due to the complexity and variety of shapes. Automation takes too much time and makes almost impossible to automate the process at an affordable cost.

Zero Gravity 3D (ZG3D) has the objective to improve the QC process applying innovative 3D computer vision techniques to capture, reconstruct and compare each produced part against a CAD model to inspect 100% of the production. The device captures, reconstructs and compares each produced part against a CAD model. To analyse an object, the system performs multiple complex operations such as image acquisition, pre-processing, segmentation, building of the 3D model and the analysis of the constructed model.

In this project, the goal is to improve ZG3D by means of saving network usage which improves the overall performance of the solution. To do so, a new prototype (MiniI3D) is being assembled implementing an edge architecture that will reduce the network bandwidth usage. In addition to that, new algorithms are being developed making possible to manage the edge boards dynamically during the runtime and featuring the change of the pipeline of each board depending on its state and capabilities.

Figure 5-1 MiniI3DMiniI3d

Regarding the information to be gathered, MiniI3D will collect the following data for monitoring its state:

- Network bandwidth.

- Throughput, measured as parts per minute.
- CPU load.
- Memory usage.

In addition to that, the system will be configured during the runtime in the following scenarios:

1. Configuration during the system start up.

All the edge boards will report about their capabilities. Depending on the provided information, the system will be configured selecting a segmentation type and the set of operations to be performed on the boards.

2. Optional segmentation executed at the edge boards.

Depending on latency CPU load, memory usage or any other changing condition the segmentation step will be skipped by the edge boards and performed by the workers.

3. Raw image sending.

When the image ROI is bigger than the 50% of the original size, then the image will be sent in raw format.

5.2 Reference Architecture

This ZG3D prototype has been adapted to include an edge computing component which is called Edge Capturer. Because of this, the system can be divided into two main parts, one being the distributed system and the other the edge component.

The architecture diagram below shows a 'Master' node. This machine communicates with the 'Capturer' to request a new image from the cameras. Later on, when the edge boards send the images taken by the cameras, the 'Capturer' stores them in a queue where the 'Master' is able to read them. The 'Master' also includes a workload 'Dispatcher' that distributes the captured images among worker agents ('Workers') where the reconstruction of the object is finished.

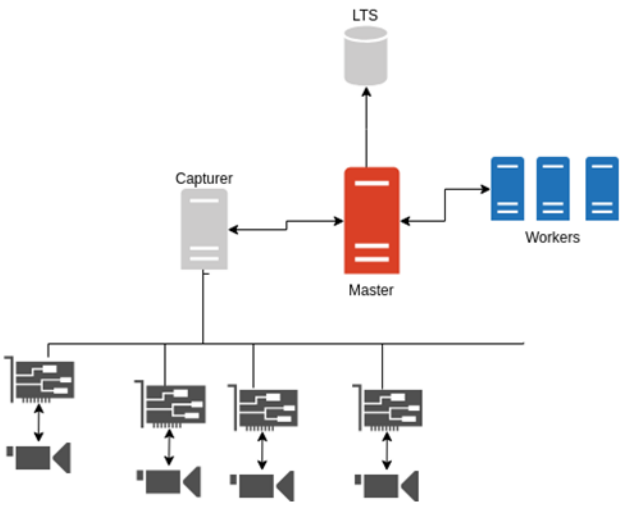


Figure 5-2 ZG3D architecture

The Edge Capturer is the main development carried out during this project. Due to this, the component has been modelled in QRML. The model defines an application and an execution layer. Hardware is composed of cameras and edge computing boards which are abstracted, making possible to give alternatives in terms of the board type to use. As an example, Nvidia Jetson Nano or Raspberry Pi boards can be installed to perform edge computing operations. The software layer encapsulates two subcomponents, the image transferring for managing image data transmission and so, the bandwidth usage, and the image segmentation which selects different types of segmentation methods depending on the system load.

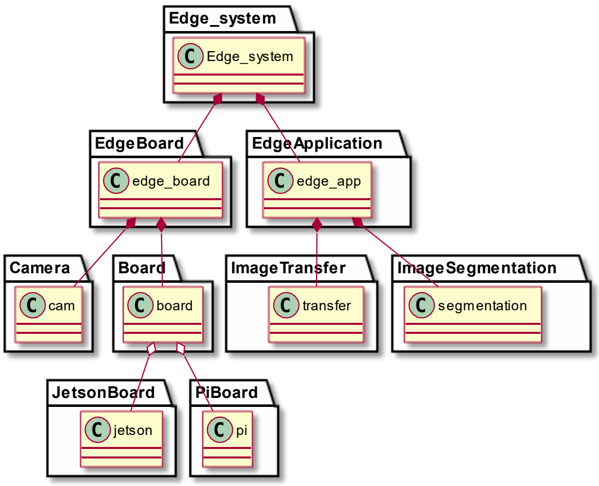


Figure 5-3 CamEdge diagram

5.3 M24 Partial Demonstrator

First version of the demonstrator includes the following features:

- A fully assembled prototype, having all the sixteen cameras and edge computing boards installed. This prototype follows the reference architecture represented in QRML.
- CamEdge software component is adapted to this new edge architecture. The device is capable of capturing images and generating the 3D representation of the object.
- Another feature available regarding monitoring is the capability to send data to FIVIS.

5.3.1 M24 to M36 GAP

The final version will add the following developments:

- Complete version of the prototype will add the runtime configuration when the device is started.
- Additional algorithms for resources reconfiguration in runtime will be available.
- The monitoring systems will be completed and the data will be shown using graphical dashboards.

5.4 Use Case 4 in FitOptiVis.

The following table shows the relation between the MiniI3D demonstrator and FitOptiVis technologies.

Demonstrated Technology	WP in FitOptiVis	Role in Demonstrator	Date for demonstration (M24/M36)
Reference architecture	WP2	Provides the architecture of the demonstrator	M24
System configuration from QRML generated configuration file	WP2	Provides de configuration when the prototype is started	M36
Integration with FIVIS monitoring system.	WP4	Data stored in FIVIS	M24
Monitoring data dashboard	WP4	Dashboard available from FIVIS	M36
Algorithms for resources reconfiguration	WP4	System reconfiguration	M36
CamEdge software adapted to Edge boards and new architecture (Edge Capturer)	WP5	Software component captures images and 3D objects are generated	M24

5.5 Use Case 4 Metrics

User Need	Validation method	Status
Easy to integrate with existing quality inspection process	Edge Capturer is completely described following the FitOptiVis abstraction model, and integration requirements of Edge Capturer are derived	Validated
Resources monitoring	The monitoring system is in place, and it is possible to check the resources consumed (CPU and RAM) at run time	Partial
Useful for different shapes and type of parts	Check that only one system (FitOptiVis 3D industrial inspection system) is capable to detect quality defects in heterogeneous parts, with different shapes. Same pieces would need more than one single system previously	Partial
User experience	Ask the user for a score (1-5) how easy handling the system is. A score of 3 or higher is considered Passed	Pending
Cost savings	Higher throughput and lower resources are used for quality inspection compared with previous methods	Pending

ID	Use Case Requirement	Verification method	Status
A1	The Edge Capturer must be composed of capturer (camera plus low execution boards) and dispatcher	Edge Capturer implemented in the pilot following the architecture	Partial
A2	Low power execution boards must be installed with dedicated connections to the cameras to perform image capture and processing	Check that there is an efficient connection between cameras and their corresponding execution boards	Verified
A3	Low power execution board shall consume less energy to perform segmentation than the computing cluster	Check that the energy consumption of the system is less than the previous one	Pending
A4	Cameras must capture high resolution images (at least 5Mpix) so parts can be measured with a precision of 52µm	Cameras used in the pilot have a resolution of 5Mpix or higher	Partial
A5	The system must be designed to scale up and down based on production volume	Simulate several production situations with varying production volumes, and check that increasing the number of workers, the production volume scales up accordingly	Verified
B1	Operator's training must be clear, short and simple, and cover all features and functionalities	Guidelines are available for operators to understand the functionalities of the system	Pending
B2	Operator should monitor regularly the performance of the system	User interface is available, and performance of the system can be monitored by the operator	Partial
F1	Low power execution boards must be able to deal with capturing problems (e.g., incomplete captures or missing objects) without assistance of the main computing cluster	Perform a set of tests under unfavourable conditions	Pending
F2	The system shall be fault tolerant and adapt when one of its worker agents fails, being able to diverge the computing load to another worker agent	Check if the system still works when a worker is disconnected while the system is processing objects	Pending
F3	The dispatcher shall distribute the computing load among the workers in a way that optimises the time needed to provide a result for each analysed part	Test the system with varying workload situations to confirm that dispatcher distributes computing load between the workers	Pending
F4	The system must understand Geometric Dimensioning and Tolerancing (GD&T) definitions so parts can be analysed to check if they	Several tolerances for different metrics can be analysed by the system	Partial

ID	Use Case Requirement	Verification method	Status
	comply with the requirements established by the engineers		
F5	Low power execution boards perform part of the image pre-processing before those images are sent to the main computing cluster	Check that images received by the workers are already pre-processed and segmented	Verified
NF1	The UI must be easy to use for a trained factory operator	Check if somebody not in the development team can be trained to use the system in a short period of time	Pending
NF2	Real time monitoring must not affect the system's performance	The average processing time for a capture when real time monitoring is on increases less than 20% when the monitoring system is off	Partial
NF3	The system must be able to differentiate several types of parts and sort them	Test the system sorting at least 5 different parts	Partial
NF4	The system must be versatile so it can analyse parts with a wide variety of shapes	Test the system inspecting at least 5 different parts	Partial
NF5	The time elapsed since a capture is taken until the system response is below 5 seconds	The time taken to fully analyse a part is less than 5 seconds under production conditions.	Partial
NF6	The size of an image after the edge capture process should be less than 1 MByte	A capture processed and segmented images weights less than 1 MByte	Pending
NF7	Throughput should be appropriate for production environment	Measure throughput of the system, 10 parts or less per minute	Pending

5.6 End user feedback and recommendations

Most companies apply QC processes only to a small part of their production. Sampling methodologies in their quality control processes need human interaction and are slow procedures involving gauges, profile projectors (analog or digital), 2D and 3D computer vision and others. When the inspection of the whole production is required, the production costs increase highly.

Industry is demanding a system capable of working in real time meaning that there will be no significant delay between production and availability of each order caused by the QC process. Ideally, they are asking for an almost perfect quality control device which could increase the speed and accuracy of the existent processes, minimizing errors, providing traceability to track defects and contributing to reduce costs. Taking this into account, the solution should accomplish the following user needs:

- The solution should be easy to integrate with the existing production line.
- The user experience should be improved. The UI must be easy to use for a trained factory operator.
- The device should provide a resources monitoring component.
- The quality control of parts must not affect the system’s performance.
- The system must be versatile so it can analyze parts with a wide variety of shapes.
- The system must be able to differentiate (classify) several types of parts and sort them.
- The time elapsed since a capture is taken until the system response should be minimal to allow the inspection of the whole production.
- The system should reduce costs.

6 Use Case 5 – Road Traffic Surveillance

6.1 Full Demonstrator Description

Intended final demonstrator is a camera system for traffic monitoring. This camera prototype will automatically process captured images, searching for passing vehicles based on their LPs (license plates), and sending pre-selected images to the cloud. For reduction of transmission bandwidth towards the cloud, the detected LPs could be processed via recognition (reading the text). Camera system should be very robust and reliable and thus it shall handle tricky lightning conditions such as strong frontal illumination or sharp shadows that could significantly increase dynamic range of the scene. Therefore, HDR mode will be embedded in the system. It will use multi-exposure capture of the scene with subsequent image composition and tone mapping.

Demonstrator is being developed in cooperation with academic partner BUT who is within FitOptiVis project focused on image processing and computer vision and its optimization for HW platforms. For the purpose of the demonstrator, BUT will train LP detectors and prepare HW core for LP detection. CAMEA will then integrate these cores to the demonstrator. This will be demonstrated using a demo camera system based on a computational platform with Xilinx Zynq Z-7020 chip. For the final demonstration, a similar system based on the same platform is being developed. However, there is focus on industrial parameters and optimization for price reduction.

Figure 6-1: Camera prototype with Xilinx Zynq Z-7020

.

For the purpose of LP detection, a novel detection algorithm was selected. It is a classification cascade that is using decision trees over Aggregated Channel Features (ACF). It provides very good performance, high accuracy, good capability of various objects detection under changing lightning conditions, and last but not least it enables very effective implementation in HW. BUT partner has prepared an open source tool for training of such classifiers (see Deliverable D3.2). At the same time, a powerful HW implementation of the detector in the form of an IP core for FPGA has been prepared.

The classifier for this platform has been trained based on LPs dataset collected by CAMEA. The classifier gives good results in terms of accuracy of detection and it is comparable with a current solution that is used in real applications. However, these applications are running on a PC. Switching to a computational platform based on FPGA resulted in lower power consumption, price reduction and better reliability. Parameters of the synthesized IP core for LP detection are:

- Aligned LPs (images captured by static cameras) with a range of LPs size in the image is 50 to 500 pixels.
- ACF detector with 4 channels of HOGwith subsampling using max pooling method and depth of tree equal 2.
- Optimization via static planning of memory access using two memory banks.
- Size of input image is FullHD times 2 exposures at 30 FPS (60 FPS total)

Figure 6-2: Example of evaluation of the detector using collected testing dataset.

For the processing of HDR images, multi-exposure principle has been employed. A sequence of two or three images with different exposure is captured in as short time as possible. In consideration of sufficient processing power, LP detection is performed on each of the individual exposed images. This approach shall bring the best precision. For the purpose of surveillance camera application, multi-exposure images are composed to HDR range on FPGA and subsequently converted to displayable range using tone mapping. Two IP cores have been developed. First one for composition of multi-exposure images providing 18-bit HDR image, and the second one tone mapping these 18-bit images to 10-bit standard range using a logarithmic operator. Both these IP cores are able to process multi-exposure FullHD images at 60 FPS.

6.2 Reference Architecture

The camera utilizes both of our IP cores provided to the consortium - IPs of Object detector and HDR acquisition. The SmartCamera requires an image input which can be provided by directly connected CMOS sensor or external third-party camera. This block is then specified as a platform component. The stream of images in stored into another platform component Storage, which is represented by DDR memory.

The Preprocessing block is acquiring buffered image data from Storage. Depending on configuration, the HDRAcquisition IP is utilized or not. If so, the HDR merge and ToneMapping IPs are included into the pipeline, otherwise only the debayerization block is inserted. HDR block merges three images into one and for that case there is specified a requirement for higher Storage throughput. The output of ImagePreprocessing is the remaining LDR due to Tonemapping block. In both alternatives, the output is RGB and grayscale, RGB for possible visualization/encoding of image, the grayscale is present for Object detection. This last part of pipeline is expressed as platform computation component ObjDetectorCompute and it has two alternative implementations, one in FPGA and second as software running on Zynq’s ARM. The detector performance, according to implementation and input resolution, is expressed in model. The detector output in forms of metadata is directed to the FIVIS component, which analyzes and uploads the detection statistics to the cloud.

The components provided in the form of IP core are marked in the model scheme.

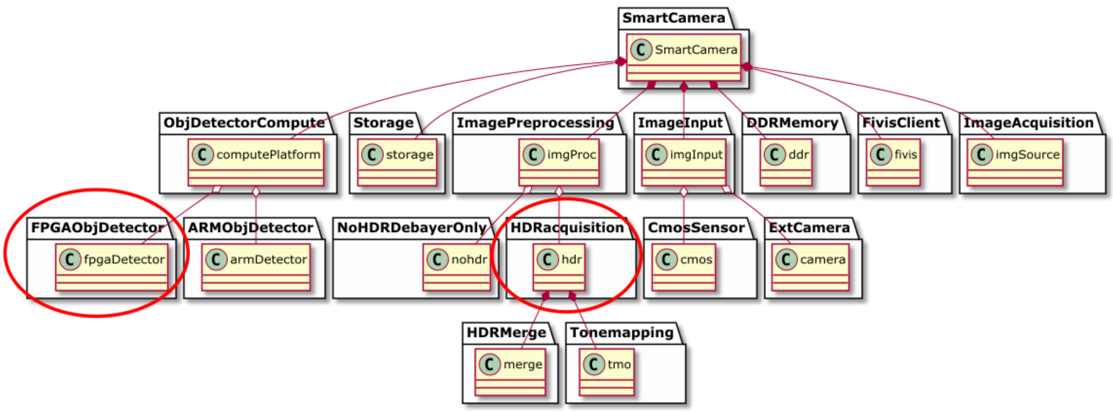


Figure 6-3: Reference Architecture

6.3 M24 Partial Demonstrator

The whole system including detector IP core and pipeline for HDR processing has been integrated in Xilinx Vivado IDE. The design has been synthesized for demonstrational platform Xilinx Zynq Z-7020. Service libraries for individual components have been coded and an application for an ARM processor controlling the whole system has been created. Currently, first testing and tuning steps of the custom code using profiling tools developed within the project so far were carried out.

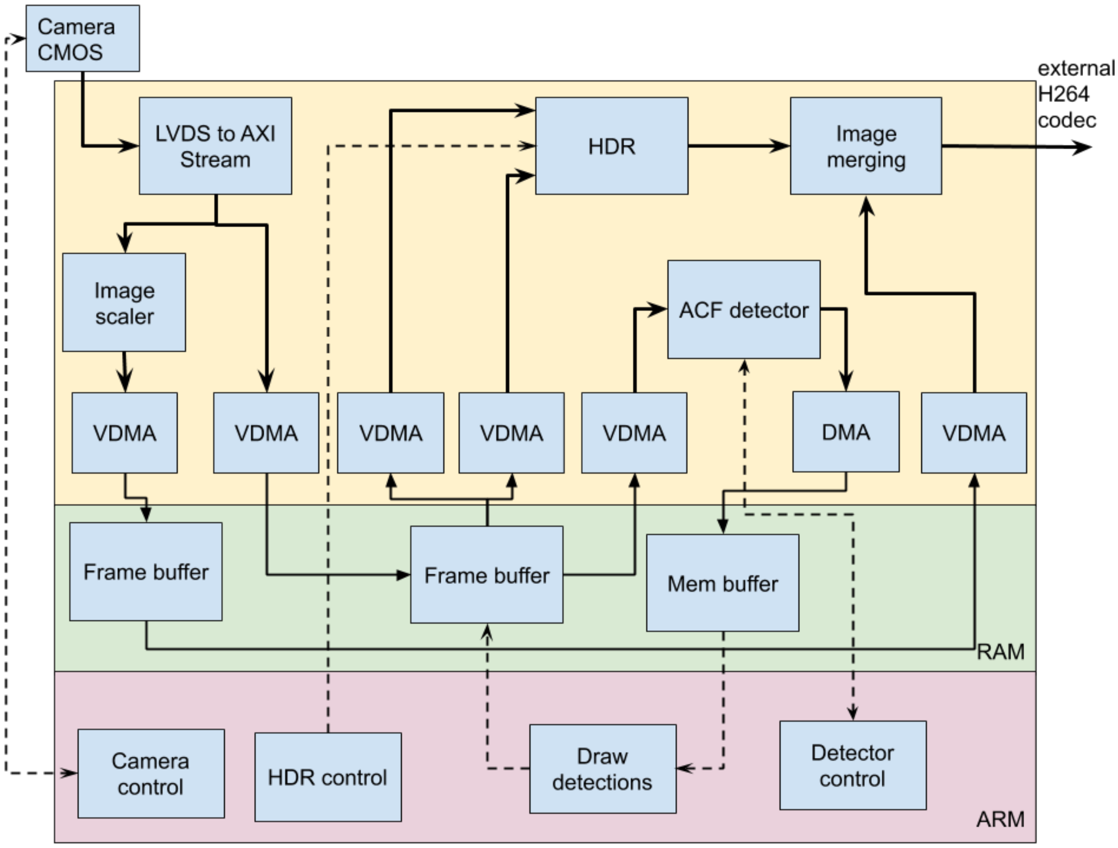


Figure 6-4: Block scheme of the system for detection of LPs including HDR pipeline for display.

	LUT	REG	BRAM
ACF Detector IP	5418	8015	48
HDR processing	1372	2850	3

Table 6-1: FPGA resources consumed for the demonstrator design.

6.4 M24 to M36 GAP

In the next period, we will continue with testing and tuning of the custom code using improved profiling tools developed within the project is in progress. We will also focus on evaluation of results of detection performed directly in camera. Currently, all processing is done in a connected PC or remotely in the cloud. This brings significant demand on network infrastructure and necessary bandwidth towards the cloud. First option is single image selection (per individual LP - vehicle's pass). This can be done based on LP detection when tracking of the detected objects is done and the best image is picked. Next option is direct LP recognition in the camera and transfer of result strings to the cloud with subsequent image transfer on demand. LP recognition is a quite computationally intensive task and the ARM processor embedded in the Xilinx Zynq platform has only a small portion of power. Nevertheless, after first experiments it can be estimated that LP recognition taking a couple of seconds is doable and sufficient for some applications.

For HDR processing we are planning to involve a more advanced deghosting algorithm into the HDR merging block and for tone mapping we are planning to use Durand operator and to involve it in the temporal coefficient filter, which will enable the smoother brightness adaptation.

6.5 Use Case 5 in FitOptiVis

Demonstrated Technology	WP in FitOptiVis	Role in Demonstrator	Date for demonstration (M24/M36)
ACF-CORE	WP3	IP for license plate detection	M24
HDR-CORE	WP3	Image preprocessing from CMOS	M24
FIVIS-CONNECTOR	WP4	Runtime device monitoring	M24
WB-PACKAGE	WP3	Design time - training of IP detection models	M24
IP-PROFILER	WP3	Design time optimization of execution of tasks on FPGA/CPU	M36
Use of components: License Plate Detector	WP5	Demonstrator building blocks	M36

6.6 Use Case 5 Metrics

6.6.1 User Needs

User Need	Validation method	Status
Dense information about traffic flow that can be used for traffic optimization for safer road and cleaner air (e.g. rerouting traffic, speed reduction).	Demonstration/evaluation by member of end user board	Shall be satisfied thanks to heterogeneous infrastructure and low-power accelerated processing.
Reasonable costs and complexity of the system that can be installed on the site of any city (possibly added to existing infrastructure) and then easily maintained (mostly remotely).	Cost calculation/analysis based on HW components used in the demonstrator	Shall be satisfied thanks to heterogeneous infrastructure and low-power accelerated processing.
Reliability of the system under all conditions – weather and 24/7 operation.	Demonstration/evaluation by member of end user board	Shall be satisfied thanks to heterogeneous infrastructure and low-power accelerated processing.

6.6.2 Use Case Requirements

ID	Use Case Requirement	Verification method	Status
A1	CMOS sensor with variable exposure sequence, high resolution and framerate	Select suitable sensor (based on datasheets) and carry experiments that verifies declared parameters	Done by design, tested
A2	FPGA of appropriate size	Select suitable FPGA based on estimated demand of sources of testing firmware	Done by design (Xilinx Zynq Z-7020)
A3	ARM CPU (with Linux OS)	Select platform with ARM cores available	Done by design (Xilinx Zynq Z-7020)

ID	Use Case Requirement	Verification method	Status
A4	HW element for digital signature	Schematic review Performance calculation	Not yet available (security won't be probably solved in this project)
A5	Wireless network interface	Measurement of reaction times	Reaction time sufficient for image transfer (not strictly real-time critical)
A6	Computational server	Design review	Computation done in local PC
B1	FPGA will control CMOS sensor	Use PC's to generate video streams that correspond to the various source types	Tested and verified
B2	ARM CPU will control detector	Inspection of the generated output images	Tested and verified
B3	ARM CPU will send image signed image to processing server	Use actual 4K monitor for display	Tested sending to connected PC
F1	Cutting edge detector	Use PC's to generate video streams that correspond to the various source types, PC's adapt behaviour based on communication with the demonstrator	Tested, benchmarked and verified
F2	HDR image composition	Test against golden model.	Tested and verified
F3	Detailed HDR-based LP detection	Test against golden model.	TBD
F4	Offloading to cloud	More powerful remote processing availability?	LP recognition done remotely
F5	Digital signature of image	Signed images checked at reception side.	Not yet available (security won't be probably solved in this project)
F6	Low heat dissipation	Measure heat in the box under load	TBD (final implementation on target platform)
F7	Outdoor operation (sealed in box)	Measure heat in the box under load	TBD (final implementation on target platform)
F8	Battery operation	Measure power consumption	TBD (final implementation on target platform)
F9	Wireless connection	Can be data transferred wireless?	Done with wireless adapter
F10	Easy installation and serviceability	TBD	TBD
NF1	At least 3 individual exposures (CMOS)	Collect individual exposures with varying time. Check images and compare them.	Done - satisfied
NF2	5Mpx+ resolution (CMOS)	Check in datasheet, check output image.	Done - satisfied
NF3	Frame rate 60FPS	Check in datasheet, check output stream.	Done – tested, verified
NF4	Power consumption 4.5W	Measure power consumption under load.	TBD
NF5	Code compatibility layer	Is the standardised interface used for IP cores?	We use AXI interface

6.7 End user feedback and recommendation

So far, our use case has not received any relevant points or recommendations by FitOptiVis end user board

7 Use Case 6 – Multi Source Streaming Composition

7.1 Full Demonstrator Description

Figure 7-1: Hospital staff in ER using a large screen with information from several image sources

When a hospital team is operating on a patient, they need visual information presented on a large-screen monitor. The data displayed comes from various devices often from different manufacturers and with varying image properties, ranging from x-ray- or echo-images to detailed graphics.
Even though the image data comes from different devices, to the user the system acts as a single entity with a single point of entry. This is realized using a compositor and coordination device which merges the images from the devices to a large screen display but also controls those devices and manages the video-streams.
At the state of the art, video streams are simply scaled to the right size and composited onto a screen. The scaling algorithms currently used are adapted scaling algorithms intended for medical information, which means that graphics are distorted (especially when scaling down) and there is no management of the video sources themselves neither can the image sources interact.
While image latency in this situation is very important for eye-hand coordination, a typical compositor adds between 20 and 45 milliseconds of delay which is a large part of the total image latency.

The demonstrator for use case 6, Multi Source Streaming Composition, aims to create a new FPGA based compositor engine that:

- At least has the functionality of the Philips Flexvision System:
  - 8 independent Full HD streams composited on a single large (4K) screen using four 10 GB Ethernet interfaces
  - Ability to render onto 2 smaller screens in stead
- Allows for more than 8 streams in case of reduced data-rate streams (smaller image sizes or frame rates below 60 Hz).
- Has enhanced, adaptable downscaling algorithms that do not result in distorted graphics.
  - When the image type is known (e.g. graphics or x-ray images) an algorithm tailored to the specific source can be used).
- Allows for up to two large (4K) screens or multiple smaller screens each with their own screen layout.

- Has an image processing latency below 1 millisecond.
- Needs less than 60 Watts

The core of the demonstrator consists of the following items:

- A image stream router (called MUX in the picture below)
- A set of downscalers each capable of handling up to 4 input streams with a total bandwidth of up to 280 MPixels/second
- Memory for intermediate image storage
- A set of upscalers that directly render the image on the output screen(s) without intermediate storage
- Image overlay generation

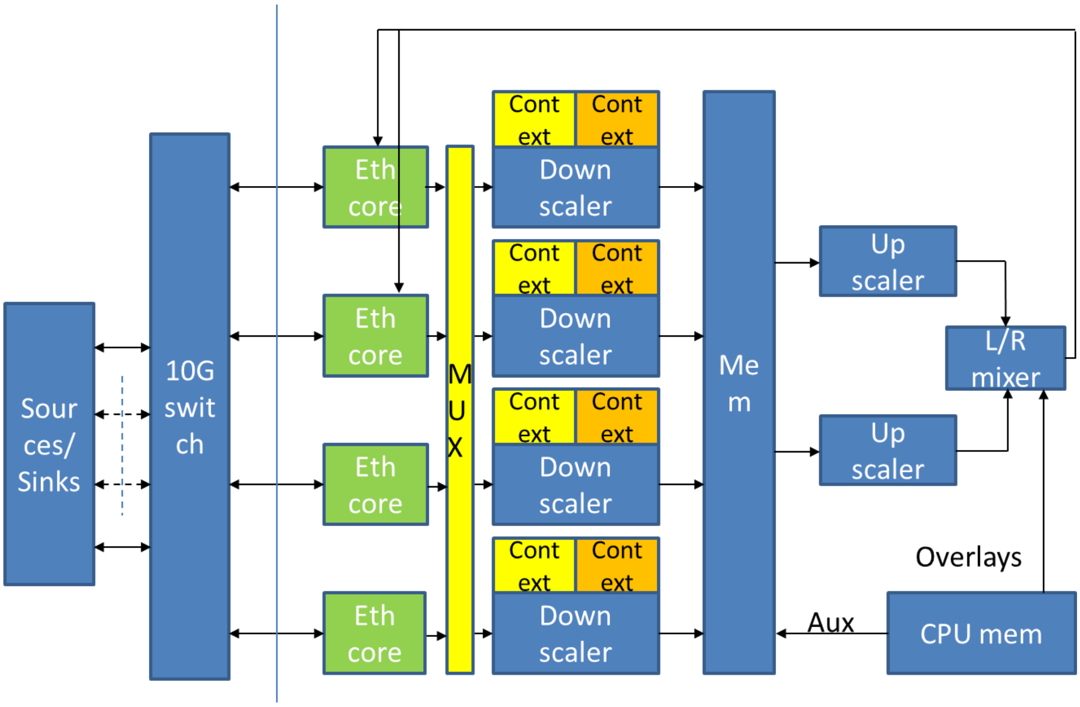


Figure 7-2: Architecture block diagram

Note: We plan to use HDMI output for the demonstrator instead of 10G Ethernet but a commercial product based on this demonstrator would have to use the Ethernet interface.

7.1.1 Reference Architecture

The picture below is the QRML-generated reference picture for the input stage and the downscaler units (simplified). Only this part is modelled because the other stages are always capable of handling the streaming data and thus do not constitute an optimization challenge.

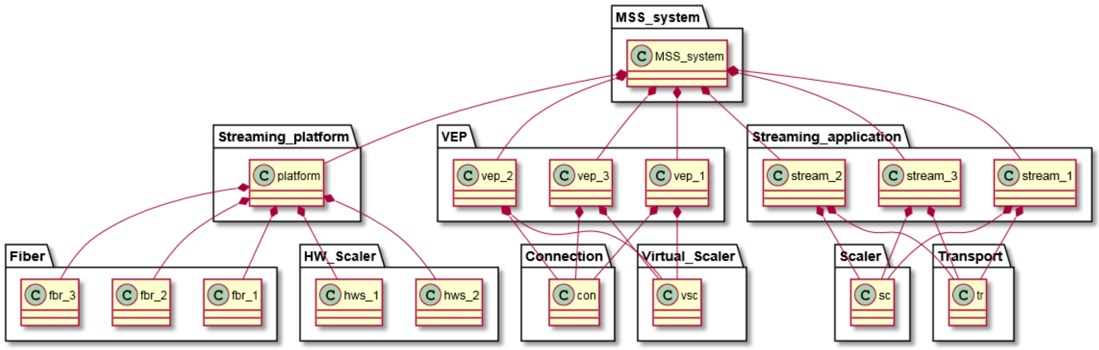


Figure 7-3: Reference Architecture

7.1.2 M24 Partial Demonstrator

The partial demonstrator will consist of:

- A very resource efficient implementation of the downscaler block in a Xilinx FPGA. It will be embedded in a SW only environment without the normal hardware input/output blocks that will be used in the final design hence it will be very slow but fully functional. The processing latency of the HW block is below 100µs.

- A port of the (Philips proprietary) Windows-based GUI for medical equipment to embedded Linux on a Xilinx SOC. This library is key to the look and feel of Philips X-ray systems and will be used in the demonstrator for image overlay generation.

7.1.2.1 M24 to M36 GAP

At M24 a working implementation of the downscaler is available. Several other blocks are in an intermediate state:

- Memory write interface (Downscaler -> DDR memory):
  - Concept ready, implementation in progress
- Memory read interface (DDR memory -> Upscaler)
  - Concept ready, implementation not yet started
- Upscaler:
  - Architecture proven in a software simulation model
  - Implementation of the upscaler control processor ready
  - Architecture of the upscaler arithmetical block itself is ready, implementation not yet started.

7.2 Use Case 6 in FitOptiVis.

Demonstrated Technology	WP in FitOptiVis	Role in Demonstrator	Date for demonstration (M24/M36)
Reference architecture	WP2	Provides the architecture of the demonstrator.	M24
Model driven development	WP3	Optimization of the processing blocks; proof of concept; Generates reference data for verification	M24/M36
Resource virtualization	WP4	Virtualization of the downscaler processing capabilities	M24/M36
Use of components Multistream Video Image Scaler and Compositor (PHL)	WP5	Demonstrator building block	M36

7.3 Use Case 6 Metrics

Following metrics are based on the results of the downscaler implementation shown in the partial demonstrator and calculation of the FPGA resource requirements for the full demonstrator.

7.3.1 User Needs

User Need	Validation method	Status
Good image quality for both “soft” medical images as well as “hard” graphics	Evaluation by image quality experts	Partial: Image quality experts evaluated downscaler results and found a significant improvement w.r.t. original algorithms
Multiple (at least 8) image sources displayed in an arbitrary composition on a large screen	Demonstration	Pending
Single point of access and/or control	Demonstration	Pending
Coordination between image devices	Demonstration	Pending
Low image latency	Review by system design	Partial: Downscaler latency is below 100 µseconds (checked in HW simulation)
10G optical Ethernet	Will be tested in combination with a 10G switch	Pending

User Need	Validation method	Status
Cost reduction	Cost calculation based on HW components used in the demonstrator	Partial: A preliminary calculation using other embedded products as input shows > 50% cost reduction
Power reduction	Analysis of heat distribution in an equipment cabinet when the new device is used instead of the current PC	Partial: A separate study (outside FitOptiVis) has been started about heat distribution in equipment cabinets. Preliminary results show a reduction from 300 -> 60 Watt

7.3.2 Use Case Requirements

ID	Use Case Requirements	Verification Method	Status
A1	FPGA SOC platform (Zynq Ultrascale+ or newer), 64 bit ARM-linux operating system	• Check (on paper) of used components • Measurement of required power in demonstrator	Partial: Platform = Xilinx ZU7EV Ultrascale+ SOC
A2	10Gb/s optical interfaces for video streaming, 1 Gb/s wired network for control	• Check of interface components	Pending
A3	24 bit uncompressed RGB video, extendible to 30 bit	• Design review	Partial: The downscaler uses 24 bit input
A3	> 1 GB, 64 bit high performance DDR memory connected to the Programmable Logic	• Schematic review • Performance calculation	Partial: 2 GB available in the demonstrator, 32 MB required per stream, max 32 input streams.
B1	Seamless change of image composition	• Measurement of reaction times	Pending
B2	No compression allowed	• Design review	Pending
F1	Receive the video input streams	• Use PC's to generate video streams that correspond to the various source types	Pending
F2	Render the video streams on (an) output canvas(ses)	• Inspection of the generated output images	Pending
F3	Transmit the resulting output streams to their destination monitor(s)	• Use actual 4K monitor for display	Pending
F4	Communication with image sources for network traffic optimization	• Use PC's to generate video streams that correspond to the various source types, PC's adapt behaviour based on communication with the demonstrator	Pending

ID	Use Case Requirements	Verification Method	Status
NF1	System in combination with a network switch must be capable to select 8 streams out of at least 24 possible sources	• Show ability to render at least 8 streams • Design review to determine extendibility	Pending
P1	Before optimization: 8 simultaneous Full HD (1920 x 1080) 60 Hz input streams (requires 40 Gb/s network bandwidth); after optimization: peak bandwith <= 18 Gb/s, average ca. 14 Gb/s	• Data rate measurements	Pending
P2	3840 x 2160 60 Hz output stream (ca. 14 Gb/s data rate)	• Data rate measurements	Pending
P3	Latency on one selectable stream < 3 ms	• Latency measurement inside demonstrator hardware	Pending

7.4 End user feedback and recommendations

Received end user comments uses case 6:

End user comments	Resolution/action
The identified user need was: single point of access and/or control, but because the new implementation will allow a number of independent monitor configurations that will create multiple work spots and from each of them access and control shall be possible	Correct: technically this can be implemented as a small extension on the original design but we still need to figure out how to handle collisions (when one source is controlled from two or more works spots at the same time).
The identified user need was: shall work with 10 G optical Ethernet. Comment: it shall work with a 10G Ethernet infrastructure but shall also be prepared for higher bandwidth	The hardware platform for the demonstrator (the Xilinx ZU7EV) supports up to 10 G Ethernet, but for example the ZU11EG supports 100 G Ethernet in a CAUI-4 configuration while still providing the required MALI support.
Note that part of the sources can become real-time connected or disconnected (mobile medical device).	The hardware will automatically detect disconnects and recover when reconnected.
Output resolution varies from 640 x 480 to 4K and in future to 8K	The demonstrator implementation supports up to 60 Hz FHD for input and 4K 60 Hz for output; but the HW architecture allows for 4K 30Hz input and up to 8K 60 Hz output. This will however require a bigger FPGA than available on the demo-board.
Target maximum delay is 1 – 5 msec for composition.	Simulations of the current architecture show that the total processing latency is below 200 μsec, but we need some extra time to cope with network jitter. A total delay close to 1 msec is expected.

End user comments	Resolution/action
New function requests: • Loupe function • Snapshot • Borderless viewports	Simulations have already shown that the upscaler already can do borderless viewports at full speed and without extra delay for rush-in/rush-out in the processing kernel. The loupe function requires an extension to the viewport compiler/driver but is technically feasible. The snapshot function can be implemented in a number of ways, but we will not implement it in the demonstrator because of time restrictions.

8 Use Case 7: Sustainable Safe MRI

Magnetic Resonance Imaging (MRI) systems are medical devices that create in-vitro images from humans by exploiting the Magnetic Resonance effect, a tissue magnetization that exists when it is exposed to strong magnetic fields and that can be controlled by magnetic fields.

As a result, the images show contrast depending on various tissue properties especially on soft tissue, which makes it significantly different from e.g. X-ray devices for which the contrast is largely based on tissue density. Next to the different type of contrast, the absence of ionizing radiation in MRI devices makes MRI a valuable diagnostic modality.

In producing the images, an MRI system can use many mechanisms to manipulate the magnetization, yielding many type of images that are showing functional aspects like, amongst many others, blood flow and directional diffusion in neurons.

Modern MRI devices are, as result of their used technology, capable of bringing serious physiological stress to the patient, in the form of e.g. tissue heating, stimulation of nerves resulting in muscle contraction and acoustic noise. In addition, the energy consumption can be significant, in the orders of tens of kilowatts.

MRI devices use software to control many functions, like providing the operator with an interface to define, in clinical terms, what kind of images must be produced, drive the system properly to create the appropriate MR signals and finally to translate these signals into images for display, analysis and archiving.

The vast flexibility to produce many different types of images, in combination with the need to control side-effects such as physiological stress, create challenges in how to control the device. Typically, these multiple objectives are contradictory, e.g., producing a better quality image in general requires device control strategies that create more physiological stress.

The purpose of Use Case 7 is to create an architecture of an MRI system and a demonstrator using FitOptiVis technology like device abstractions and multi-objective optimization to address these challenges. It will be based on newly developed hardware components that are designed to be more energy-efficient and create less physiological stress.

8.1 Full Demonstrator Description

The demonstrator will be an MRI system, based on the Philips Ingenia Elition X product. The picture below gives an impression of this product.

Figure 8-1: MRI Scanner

The following will at least be modified:

- The gradient coil;
- The body coil (radio frequency coil);
- The system software.

The picture below (from https://science.howstuffworks.com/mri1.htm) shows schematically where the hardware components (gradient coil and body coil) are located in the system. The purpose of these components is to create the strong magnetic fields required to manipulate the MR magnetization, respectively in the audio-frequency range (0-10 kHz) and radio-frequency (~100 MHz) range. As they are close to the patient, this can result in significant physiological stress.

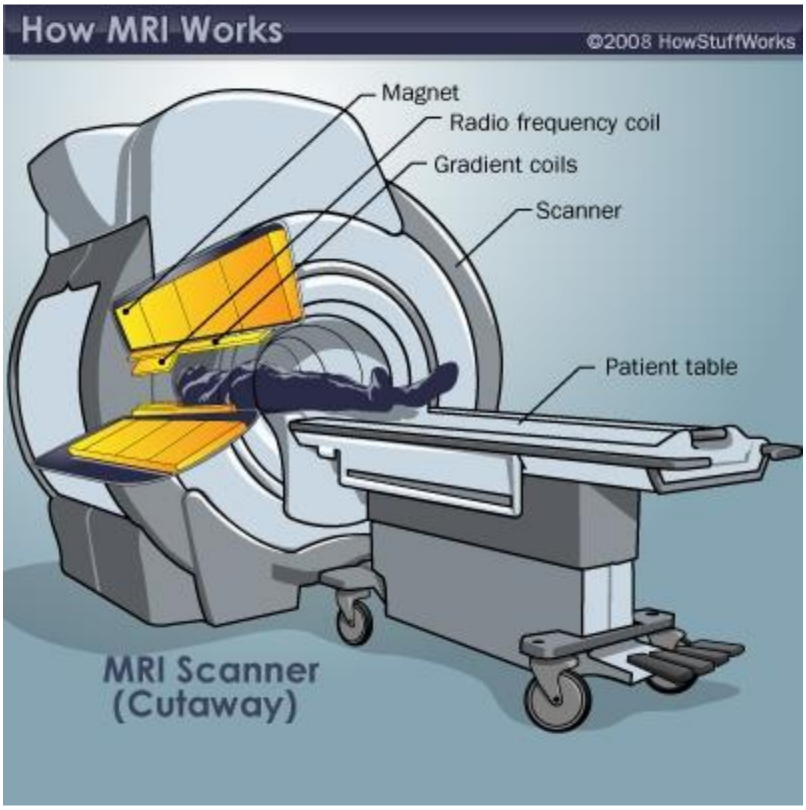


Figure 8-2: Explanation on how an MRI Scanner works

The hardware components are designed to be more energy efficient, which will be demonstrated by comparing the energy consumption for a set of reference scans, likely using the methodology defined by COCIR. COCIR is the European Trade Association representing the medical imaging, radiotherapy, health ICT and electromedical industries.

In addition, the demonstrator will show that the user will be able to control each MRI scan on energy consumption as well as a number of physiological stresses typical for MRI scans such as acoustic noise and stimulation of nerves and muscles, next to currently available controls like scan duration, image quality and safety related items.

8.1.1 Reference Architecture

The figure bellows shows the reference architecture of the software system of the MRI product on which the demonstrator is built. Highlighted are the parts of the software system that is modified for the demonstrator.

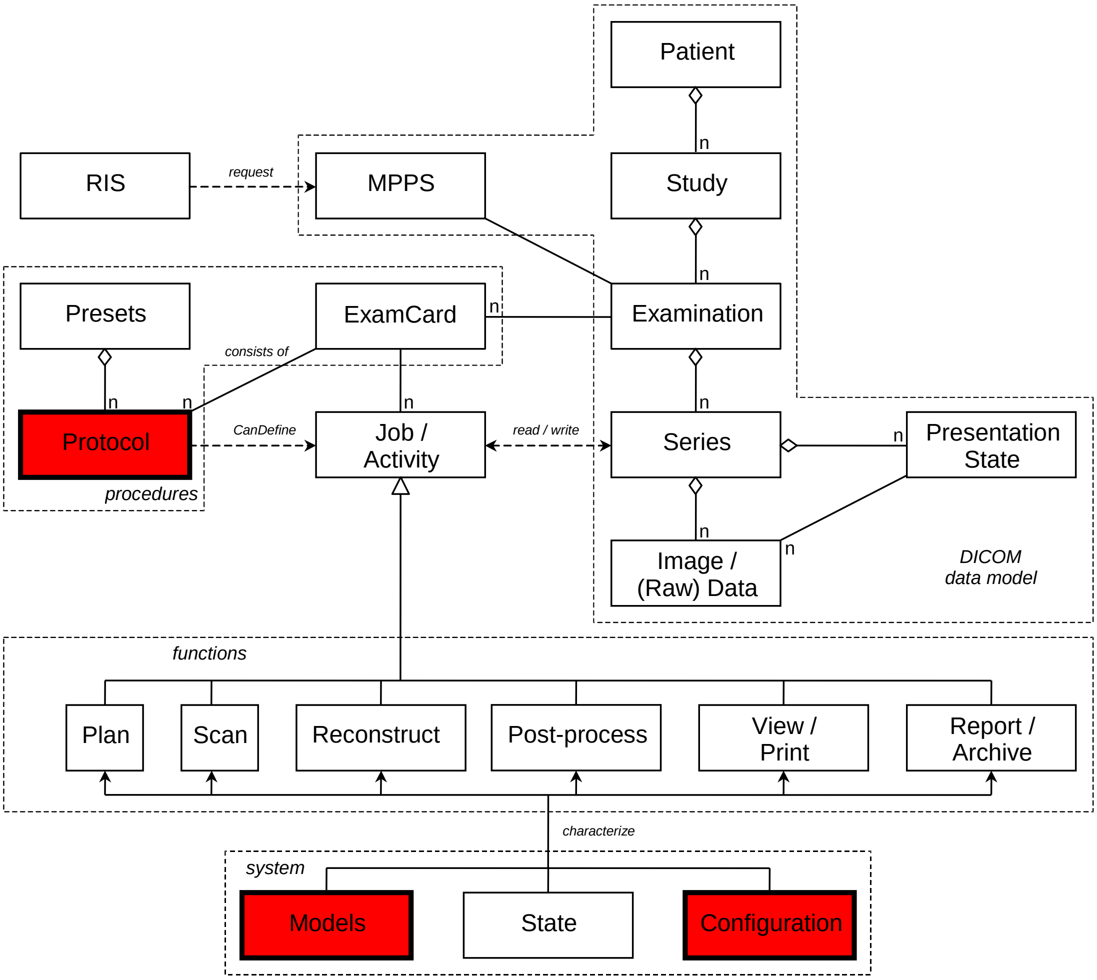


Figure 8-3: Reference Architecture

In further detailing, the picture below shows on a rather high level the involved modules and the type of data that is transferred between these models.

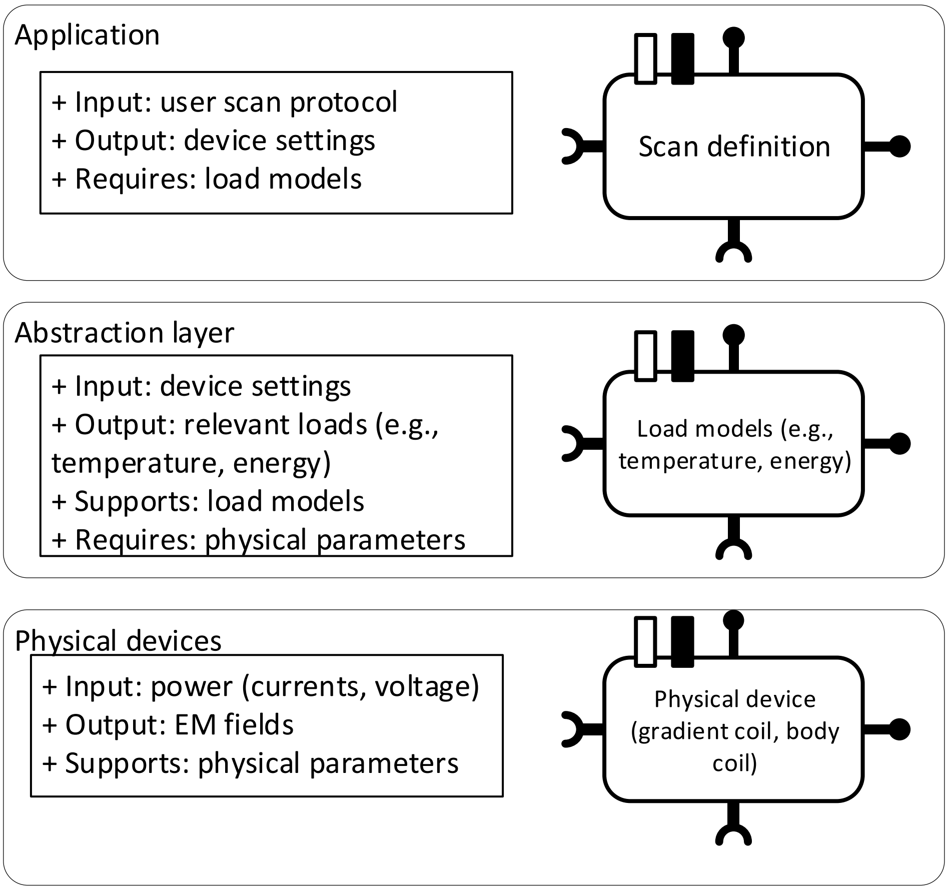


Figure 8-4: Reference Architecture

The real data is extensive and therefore not detailed out in separate parameters:

- User scan protocol: This defines, in clinical terms such as image resolution and volume to be covered as well as contrast-related parameters, the images to be produced. It is typically created or adapted from a pre-defined factory protocol by the operator.
- Device settings: This defines how the device will control the various hardware components in order to produce the desired images. Typically, it includes the many real-time control signals to control the relevant devices in an MRI system. As the manipulated MR magnetization is very sensitive to disturbances, typically the related device settings are applied in a pre-defined scheme with strict timing requirements (up to nanosecond level).
- Load models: These define how, given the device settings, the various devices are loaded, to avoid that they are driven over their capabilities when the device settings are applied. Each model predicts a certain capability or property related to capability, such as component temperatures or energy consumption. Relevant for the Use Case are models for the gradient coil, which includes thermal models, and the body coil, which includes a tissue heating model.

8.1.2 M24 Partial Demonstrator

At M24, the demonstrator will consist of a partially working MRI system with the modified gradient coil included, and adapted software to safely use the gradient coil in the system.

The safety will be demonstrated according applicable safety standards, such as IEC60601-1 (Medical electrical equipment - Part 1: General requirements for basic safety and essential performance) and IEC60601-2-33 (Particular requirements for the safety of magnetic resonance equipment for medical diagnosis).

8.1.2.1 M24 to M36 GAP

Between M24 and M36 the following is foreseen to be done on the demonstrator:

- Add the modified body coil and adapt the system software accordingly to be able to use it properly.
- Add functionality to the system software to allow the operator additional control of energy consumption and some physiological stress factors per scan.

8.2 Use Case 7 in FitOptiVis.

Demonstrated Technology	WP in FitOptiVis	Role in Demonstrator	Date for demonstration (M24/M36)
Reference architecture	WP2	Provide the architecture of the demonstrator	M36
Energy efficiency	WP3 WP5	Enable reduced energy consumption per scan	M36
Predictive modelling	WP5	Predicting the behaviour of system components with respect to power demand and heat dissipation to patients	M36
Model-driven engineering	WP3 WP5	Precisely estimate, control and minimize the thermal load whilst providing superior image quality	M36

8.3 Use Case 7 Metrics

8.3.1 User Needs

User Need (from D1.1, Chapter 2)	Validation method	Status
Enabling increasing performance of MRI systems	Compare user-level performance parameters with previous generation.	Partially, with respect to gradient performance related scans
Control physiological stress	Ask the user for a score (1-5) how well physiological stress is controlled. 3 or higher is considered Passed.	Pending
Control energy consumption	Ask the user for a score (1-5) how well energy consumption is controlled. 3 or higher is considered Passed.	Pending

8.3.2 Use Case Requirements

ID	Use Case Requirements (from D1.2	Verification Method	Status
A1	The energy consumption of a scan can be derived from available scan parameters and system properties.	Check that predicted energy consumption is available to the user	Pending
A2	The acoustic noise level of a scan can be derived from available scan parameters and system properties.	Check that predicted acoustic noise level is available to the user	Pending
A3	The system shall be safe according established standards.	Show compliance with safety standards	Partial (basic safety)
B1	It shall be possible for the user to control, per scan, energy consumption and produced acoustic noise.	Check that the predicted energy consumption and acoustic noise can be controlled	Pending
B2	The user shall be able to define the optimal scan, i.e., the best compromise between image quality, energy consumption and acoustic noise.	Check that the predicted energy consumption and acoustic noise can be controlled	Pending
F1	If the user selects a scan protocol, the system will provide information related to optimal image quality.	Check that identified IQ parameters are provided when user selects a scan protocol	Pending
F2	If the user modifies the scan protocol, the system will provide immediate feedback on resulting image quality, acoustic noise and energy consumption.	Check that identified IQ parameters, are provided	Pending
F3	If the user modifies the scan protocol, the system will support automatic optimization to targets for energy consumption and/or acoustic noise and/or image quality.	Check that system will adapt the protocol corresponding to the target parameter is changed	Pending
NF1	The prediction of energy consumption shall be accurate within 10%.	Measure difference between predicted and actual for selected protocols	Pending
NF2	The prediction of acoustic noise shall be accurate within 5 dBA.	Measure difference between predicted and actual for selected protocols	Pending

8.4 End user feedback and recommendations

The feedback from the end user workshop did not reveal relevant issues or concerns for this Use Case.

The second user workshop should confirm that the stated objectives are still relevant for the MRI market.

9 Use Case 8 – Robots calibration

9.1 Full Demonstrator Description

9.1.1 Overview

The **robot calibrator** demonstrated as UC8 is used to calibrate robots (most often collaborative ones) - i.e. **position the robot in the global coordinate system** and **determine its parameters (arm lengths, offsets)**. It consist of 2 main components – point tracker that can localize POIs (Point Of Interest) in 3D world coordinate system and mathematical apparatus that estimates robot parameters from parametrical robot model and point cloud acquired during the calibration movements.

The point localization engine works on optical principle and uses novel approach of differential imaging of active LED markers that removes registration problems in stereovision applications and thus strongly increases detection robustness and reliability, allowing the solution to be used even in harsh industrial conditions. The approach was proposed by UWB and REX in scope of the project and according to the survey there are currently no systems working on this principle.

The solution targets at mid-level applications where it is not possible to afford high-tech calibration methods but still precision is of a high concern. It is estimated that manufacturing costs of a system capable to localize points with sub-millimeter accuracy in working area of 1x1x1m and corresponding calibration precision will be around 4000,-EUR.

Also, intention is to make the calibration process for the user as simple and straightforward as possible. The operator first places 2 cameras working as a stereovision system and mocap (motion capture) LED markers in the scene and on each robot arm (Figure 9-1). The reference markers depicted in green determine the coordinate system. They shall be placed on fixed places - on the walls or fixed components of the technology, and their real-world coordinates need to be known. At least 3 of these tags must be placed. On the other hand, the POI markers on the robot body (depicted in yellow) can be placed arbitrarily, only with regard to their visibility from the cameras (majority visibility, it is of course not possible to ensure visibility in each robot position) and with regard to stability of calculations – i.e. the further from the previous joint, the better.

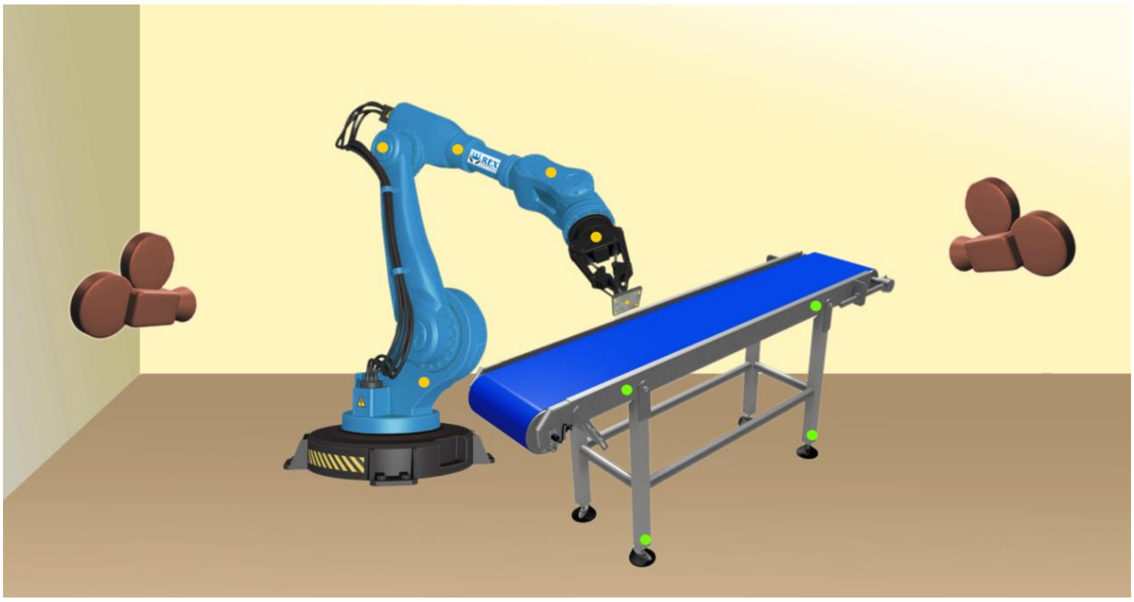


Figure 9-1: The principle of robots calibration using stereovision

The robot then executes a series of pre-defined calibration movements. During the series of calibration movements an array of point clouds (one cloud per marker) is obtained. Analysis of the data together with the known calibration trajectory derives the topological location of the marker (before / after certain joint), the type of the joint (linear / rotary), robot parameters (axis length, offset) and finally exact location of robot in the global space.
Structure of the robot calibration solution and relations between components are depicted in Figure 9-2.

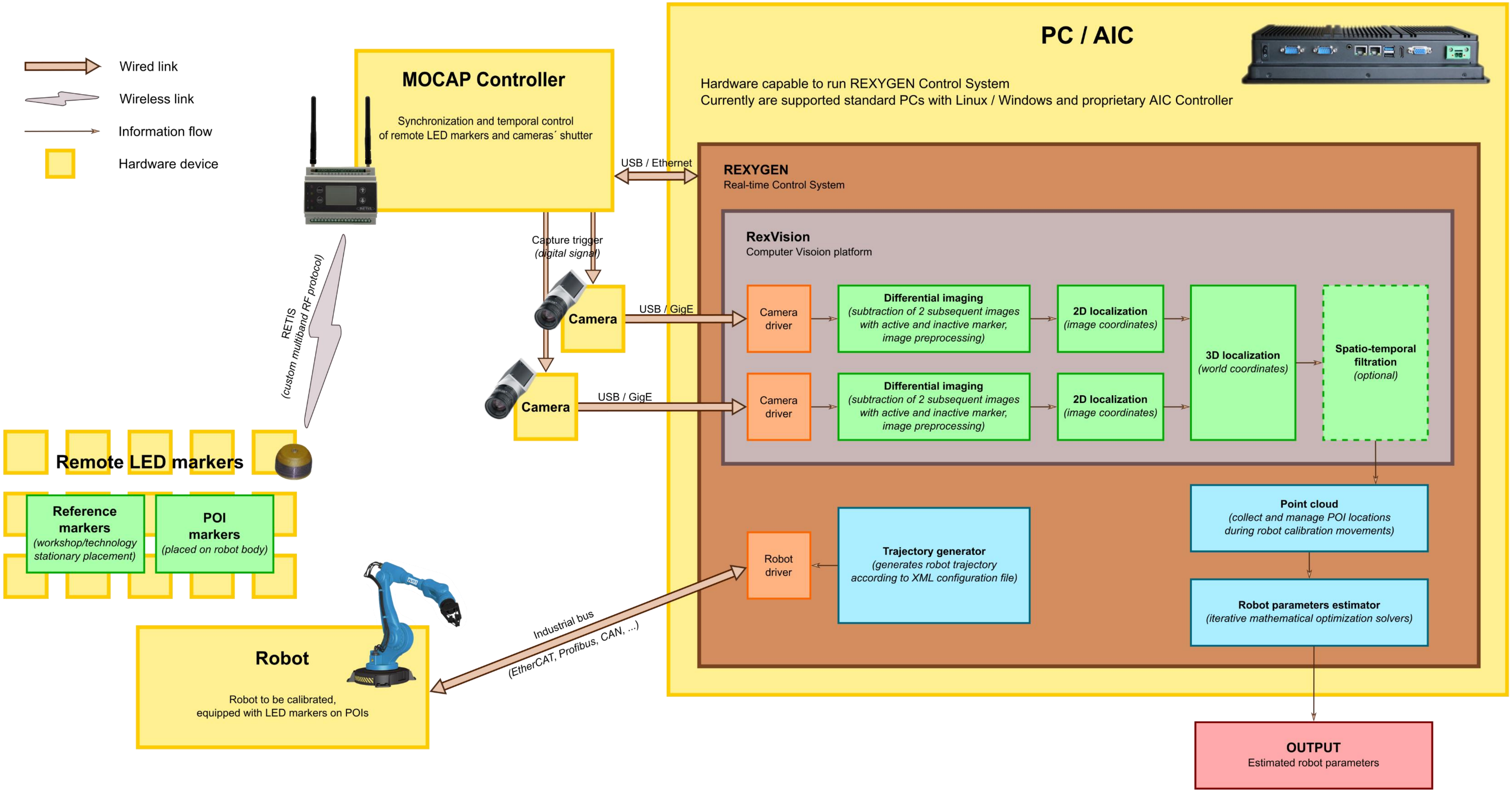


Figure 9-2: Structure of the robot calibration solution

9.1.2 Point Tracking Subsystem

Fast and robust detection and subsequent localization and registration of POIs in the image are key tasks in many control applications requiring 3D localization of POIs from input image data. There are many methods for detecting and registering POIs. Most often are used passive tags made of highly reflective material, actively radiating tags (usually LEDs) or distinctive printed tags (typically QR codes). The first two types of markers allow very quick detection of POIs in the image, but subsequent registration (matching corresponding points in images from 2 stereo cameras) is more problematic as the characteristics of the individual POIs are almost identical. This disadvantage is eliminated by tags in the form of QR code, where each tag has a unique characteristic and registration is unambiguous. However, the disadvantage here is longer processing time of localization of the mark in the image.

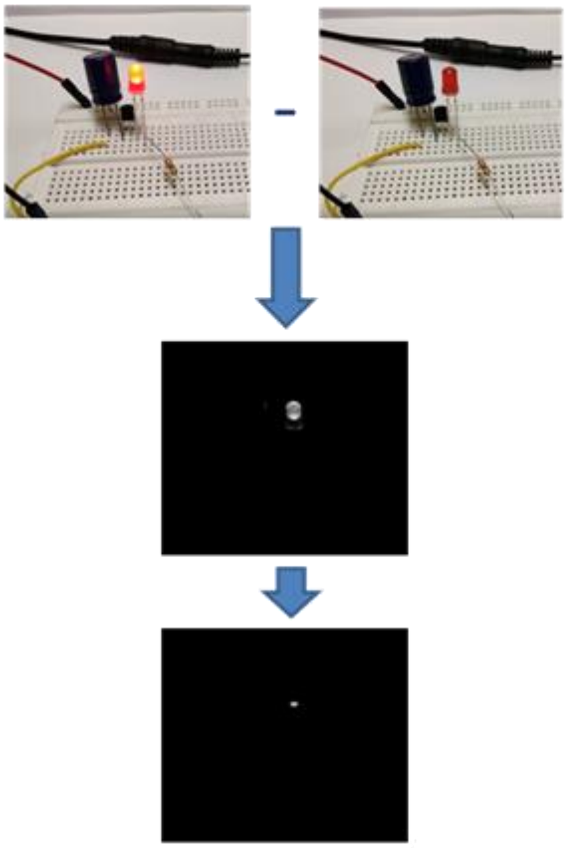


Figure 9-3: Principle of differential imaging

In the proposed solution, UWB tries to combine the advantages of both approaches – i.e. to enable rapid localization and simultaneous unambiguous registration of POIs. The original, yet unpublished solution is also not used in any commercial system. The principle of the method is to obtain differential images of the scene, where the images with the LEDs on and off are subtracted from each other. This method is very robust, as the 2 successive images differ only in the on / off LED marker. The change in the differential image is very significant and therefore detectable and localizable using simple and very fast image processing algorithms (thresholding, morphological operations, center of mass evaluation).

The active LED markers based on Silicon Labs EFR32 MCUs are wirelessly synchronized with the camera controller. This coupling allows generation of LED flashes whose timing and duration are synchronized with the shutters of cameras connected to the controller with microsecond accuracy. Very accurate synchronization is achieved using proprietary communication protocol developed by UWB and REX as part of the TAČR Delta 4.0 project “Reliable Time-Sensitive Networks in Distributed Cyber-Physical Systems for Real-Time Control Industry 4.0 Applications” (No. TF04000048).
In quick succession, the camera controller takes a pair of images - one with the marker light on and the other with the marker off. The position of the POI in image coordinates is determined from the differential image. Thanks to the knowledge of which of the markers in the scene is lit, it is possible to register POIs between differential images from multiple cameras. Registration succeeds almost in all cases (because only the POI being processed is visible in the scene) and fails only when the LED marker is significantly reflected on a shiny surface or in a mirror and generates a fake image in the scene.

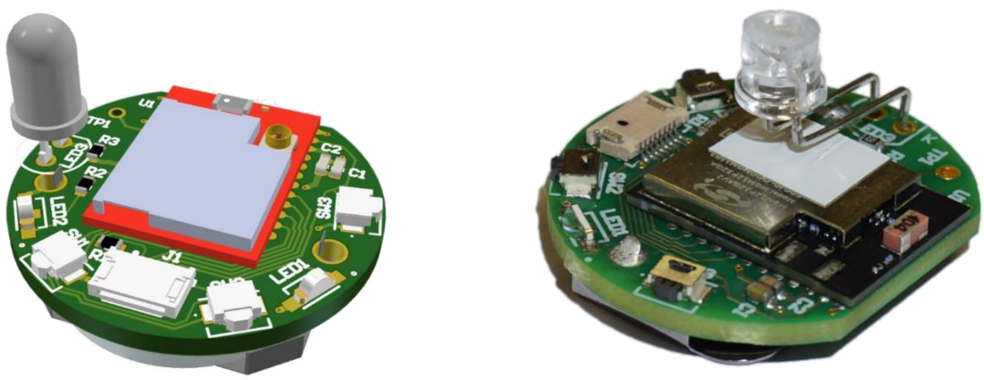


Figure 9-4: Active LED marker – 3D model and final PCB

Figure 9-5: Final LED marker encapsulated in its chassis

Image coordinates of POIs from 2 (stereovision) or even more cameras give us coordinates of the POIs in camera’s projection planes. As the correspondence between points in projections from different cameras is known (due to the novelty proposed mechanism), the full 3D coordinates can be directly calculated from the 2D projections.

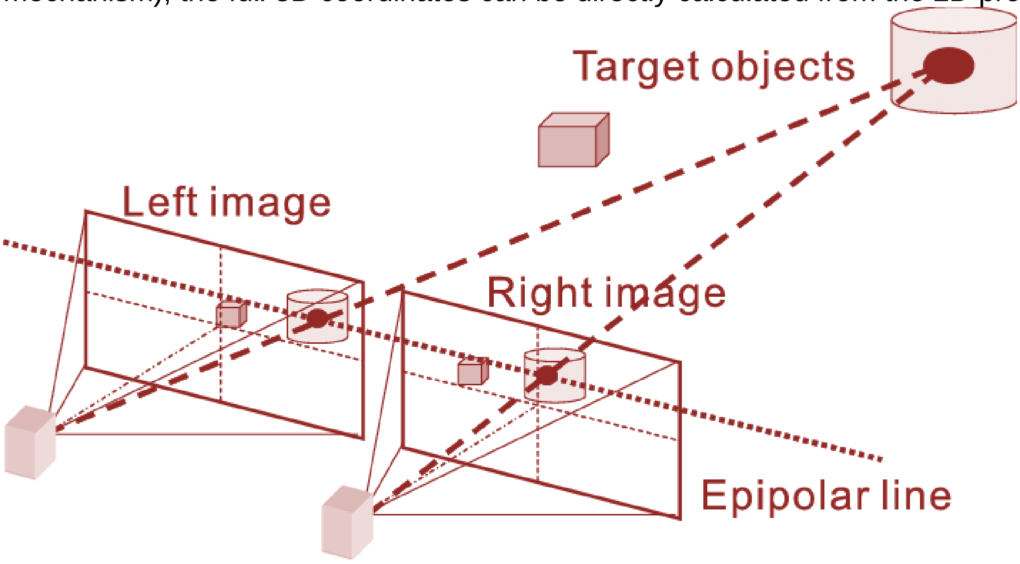


Figure 9-6: 3D localization from 2 projections (stereovision)

9.1.3 Robot Parameter Estimation

During the analysis of the robot calibration problem it was shown that it is reasonable to perform the calibration in 2 steps:

- Joint-by-joint calibration of every robot kinematic pair (from the base to the end-effector). This is the basic mode of the calibration where the kinematic parameters are estimated one after another from simple movements when always only one axis is operated while the others are stationary. The main advantage is the possibility to estimate the overall robot kinematic model without necessity to know the nominal kinematic parameters (e.g. initial estimation).
- Overall complex calibration - calibration procedure for complex robot moves where the end-effector position and orientation are measured and the robot kinematic parameters are estimated (from the initial values/initial estimation) from comparison between commanded (reference) end effector trajectory (measured by the localization system) and the end effector trajectory recomputed via forward kinematics (dependent on robot kinematic parameters). The end effector position difference is propagated as the position error to the non-linear least squares iterative method.
  This fine-tuning calibration procedure is an iterative non-linear mean square optimization method based on overall calibration of complex robot movements starting from initial conditions given by the parameters estimated during the previous step. The correct starting point determined in the previous step is an imperative as the functional is strongly non-linear with many local extremes. Without proper initialization the iteration process ends in a local minimum. Moreover, the overall calibration method brings some problems which have to be solved:
  - End-effector position error computation (simple in the translation, more complicated in the orientation)
  - Robot kinematic model identifiability/observability – some kinematic parameters may not be observable from proposed end-effector position measurement
  - Calibration error estimation

9.1.4 Trajectory generator

The component takes care of generating robot trajectory according to configuration file and in cooperation with robot driver it controls the robot movements. Supported are both robots with position feedback and without it. The robot is commanded through robot driver. Various kinds of robots are supported by existing drivers and if necessary, additional drivers for new robot types can be implemented. The configuration file is based on XML and describes:

- Robot configuration – number and type of joints, position, speed and acceleration limits
- Calibration trajectory – spatio-temporal path description including calibration stops and delays required for accurate measurement

The example below describes 5 DOF robot with 1 linear and 4 rotary joints and commands it to move sequentially two axes:

```xml
<Document>
  <Robot>
    <Joint Name="Base" Type="Rotation" Min="0" Max="180" MaxSpeed="20" Accel="5" Delta="0.1"/>
    <Joint Name="Elbow" Type="Rotation" Min="30" Max="60" MaxSpeed="45" Accel="10" Delta="0.5"/>
    <Joint Name="Arm" Type="Linear" Min="-500" Max="500" MaxSpeed="2500" Accel="5000" Delta="0.5"/>
    <Joint Name="Wrist1" Type="Rotation" Min="0" Max="360" MaxSpeed="60" Accel="10" Delta="1"/>
    <Joint Name="Wrist2" Type="Rotation" Min="0" Max="360" MaxSpeed="60" Accel="20" Delta="1"/>
  </Robot>
  <Motion CalibDelay="0.1">
    <MoveTo TravelTime="20" InitDelay="1.0" EndDelay="6.0" Speed="50%">
      <Base Pos="1"/>
      <Elbow Pos="32"/>
      <Arm Pos="3"/>
      <Wrist1 Pos="180"/>
      <Wrist2 Pos="5"/>
    </MoveTo>
    <CalibMove TravelTime="9" Steps="10" StepDelay="0.5" StartDelay="5" EndDelay="8" Speed="10%">
      <Base Start="0" End="10"/>
      <Elbow Start="45" End="45"/>
      <Arm Start="10" End="10"/>
      <Wrist1 Start="90" End="90"/>
      <Wrist2 Start="90" End="90"/>
    </CalibMove>
    <CalibMove TravelTime="8" Steps="12" StepDelay="0.3" StartDelay="6" EndDelay="7" Speed="40%">
      <Base Start=" 9" End="9 "/>
      <Elbow Start="40" End="60"/>
      <Arm Start="10" End="10"/>
      <Wrist1 Start="90" End="90"/>
      <Wrist2 Start="90" End="90"/>
    </CalibMove>
    <MoveTo TravelTime="20">
      <Base Pos="0"/>
      <Elbow Pos="45"/>
      <Arm Pos="10"/>
      <Wrist1 Pos="90"/>
      <Wrist2 Pos="90"/>
    </MoveTo>
  </Motion>
</Document>
```

9.1.5 Implementation

The final robot calibration demonstrator is realized as an application implemented in real-time control system REXYGEN. It integrates components created by various tools and using different technologies. The implementation details are following:

- **Active LED markers:** Small hardware devices containing LED as a light source and RF transceiver used for precious time synchronization with the MOCAP controller and to obtain LED flash schedule. Based on EFR32 ARM chipset. Firmware developed in C++.

- **MOCAP controller:** Motion capture controller. Takes care of time synchronization between the REXYGEN control system, cameras' shutters and remote LED markers. Propagates flash schedule from the REXYGEN core to the end RF devices. Based on EFR32 ARM chipset. Firmware developed in C++.
- **Differential imaging:** Algorithm implemented as REXYGEN subsystem – i.e. schematic created from interconnected function blocks.
- **2D localization:** Algorithm implemented as REXYGEN subsystem – i.e. schematic created from interconnected function blocks.
- **3D localization:** Implemented in C++ as a REXYGEN function block.
- **Spatio-temporal filtration:** Algorithm implemented as REXYGEN subsystem – i.e. schematic created from interconnected function blocks.
- **Point cloud management / Data postprocessor:** Implemented as PYTHON algorithm (via "Python" functional block for user-defined code implementation in REXYGEN) recomputing the point cloud to robot links localization (translation, orientation) in defined representation.
- **Calibration methods / Robot parameters estimator:** Implemented as PYTHON algorithm. The NUMPY (fundamental package for scientific computing with PYTHON) is supposed to be used for advanced calibration method procedures (based on linear algebra, optimization, etc.).
- **Trajectory generator:** Implemented in C++ as a REXYGEN function block.
- **Robot kinematic controller:** Robot motion controller is implemented via REXYGEN Motion control library blockset (supported by the PLC Motion Control standard). The standard motion control functions (axes/coordinate jog, prescribed robot trajectory following, etc.) are supported.
- **Output / HMI:** The standard SW tools provided by REXYGEN control system for visualization and operator interface (based on web browser technologies – HTML and animated SVG).

9.1.6 Verification and validation

The solution will be verified against professional industrial robot where the robot kinematic parameters are exactly known and thus validation of the proposed calibration engine can be performed. The workflow can be specified as follows:

- Calibration engine initialization (calibration robot controller and configuration, visual tracking system calibration, etc.)
- The robot calibration movement running – the point cloud from the tracking system is obtained
- Data postprocessor initialization and running, input data for robot parameters calibration is obtained
- Calibration method running results in exact kinematic robot parameters
- User data output (error log, estimated kinematic robot parameters, etc.)

9.1.7 Reference Architecture

The structural QRML model of the robot calibration UC can be found in appendix UC8.qrml.

9.1.8 M24 Partial Demonstrator

The current (M24) development status of the robot calibrator components is following:

- **Active LED markers:** Both HW and FW components are fully developed, **functional and ready**.
- **MOCAP controller:** Preliminary HW platform implemented and equipped with simplified firmware. **Functional and ready**.
- **Differential imaging:** Corresponding REXYGEN function blocks developed, algorithm schema implemented. **Functional and ready**.
- **2D localization:** Algorithm implemented, **functional and ready**.
- **3D localization:** Not implemented yet.
- **Spatio-temporal filtration:** Not implemented yet.
- **Point cloud management / Data postprocessor:** Implemented in simplified form. **Ready for the M24 tests**.
- **Calibration methods / Robot parameters estimator:** The first calibration method (joint by joint) is fully implemented in Matlab (later it will be ported into Python) and ready for M24 tests. The overall complex calibration is implemented partially and will not be performed during M24 tests.
- **Trajectory generator:** Fully implemented and tested -> **functional and ready**.
- **Robot kinematic controller:** Fully implemented and tested -> **functional and ready**.
- **Output / HMI:** First version implemented. **Ready for M24 tests.**

As obvious from the list above, most of the UC components are implemented and ready for use. However 2 key components are still missing – 3D localization module and calibration solver for the complex calibration method.
Due to the missing components, in M24 will be presented separately the point tracking functionality and the robot calibration functionality:

1. Active LED markers, MOCAP controller, differential imaging and 2D localization will be used to demonstrate POI localization in 2D (in a plane).
2. The robot will be operated by the trajectory generator and will automatically perform predefined calibration movements. However the POI coordinates will not be measured (as the 3D localization is not ready yet). Instead they will be simulated by nominal markers positions gained from robot model and planned moving trajectory and will be artificially loaded with errors/measurement noise. The simple joint-by-joint calibration algorithm will be then fed with (simulated) point cloud manager output and the calibration algorithms will be verified in Matlab.
   To verify the output of the partial demonstrator, a comparison of known robot parameters (for initial experiments Staubli TX40 robot will be used) and the calibrated (estimated) kinematic parameters will be performed.

9.1.8.1 M24 to M36 GAP

In the remaining period (M25-M36) of the project the currently missing components will be developed (3D localization module, spatio-temporal filtering, and calibration solver for the complex calibration method). Also partially implemented components will be improved and finalized (point cloud manager, HMI). All the components will be then integrated together to form compact solution and carefully tested.

9.2 Use Case 8 in FitOptiVis.

Demonstrated Technology	WP in FitOptiVis	Role in Demonstrator	Date for demonstration (M24/M36)
Reference architecture	WP2	Provide the architecture of the demonstrator	M24
Design Time Resources	WP3	Startup design to develop FPGA accelerated image processing	M24
Runtime reconfiguration	WP4	The ability to extend hardware acceleration support also for smaller FPGA boards	M36
Component - Wireless LED SYNC light	WP5	Remotely controlled active LED markers used to mark POIs on robot body/gripper	M24
Component - Point localization from stereovision	WP5	Computation module extracting 3D coordinates from POIs localized in stereo images	M36

The robot is populated with LED markers controlled remotely and turned on and off synchronously with the camera capture process. This way, the individual LEDs can be distinguished from each other and localized in 3D. In current development phase (during the FitOptiVis project) REX and UWB plans to build the solution on CPU based platforms as indicated in Figure 9-2.
However later on the algorithms should be moved to FPGA based platform to speed up and encapsulate the localization process. We assume that each camera will be equipped with Zynq based accelerator which will implement the following features:

1. Capability to collect camera snapshots.
2. Control camera frame capture.
3. Communicate snapshots and detected positions to the following CPU-based components
4. Implement hardware accelerated flow containing:
    a. Colour conversion (if needed). Depending on used camera the image must be converted to grayscale. We assume this operation will be only needed in testing stages as final cameras will be grayscale.
    b. Gaussian Blur filter to remove camera noise and smoothen detected point boundary.
    c. Image thresholding to get the binary image.
    d. Morphological operations erode and dilate to clean up the thresholded output.
    e. Connected component analysis to detect the LED light region and measure its moments (area and center of mass with subpixel precision)

The way how to accomplish this transition is based on in FitOptiVis developed tools and can be shown on UTIA demo. The demo uses one camera observing plate with balls of a different colour. The different colours are used to uniquely identify balls instead of operating synchronous LED lights at this stage. The goal was to prototype the processing chain, test its functionality and benchmark the hardware accelerated

components. The possibility to use individual accelerated operations and compose them into larger blocks was also tested.
Design steps:

1. The embedded board hardware was chosen to be TE0820 SoM with TE0701 carrier (Trenz Electronic).
2. The application was first prototyped using PC with OpenCV.
3. After that, the OpenCV application was ported to embedded board.
4. Components to be hardware accelerated were then replaced by their xfOpenCV counterparts and software model of the hardware was compiled and tested.
5. One function after another was then compiled to FPGA logic using Vivado HLS tool.
6. At the end, one big hardware accelerator containing all required functionality was implemented

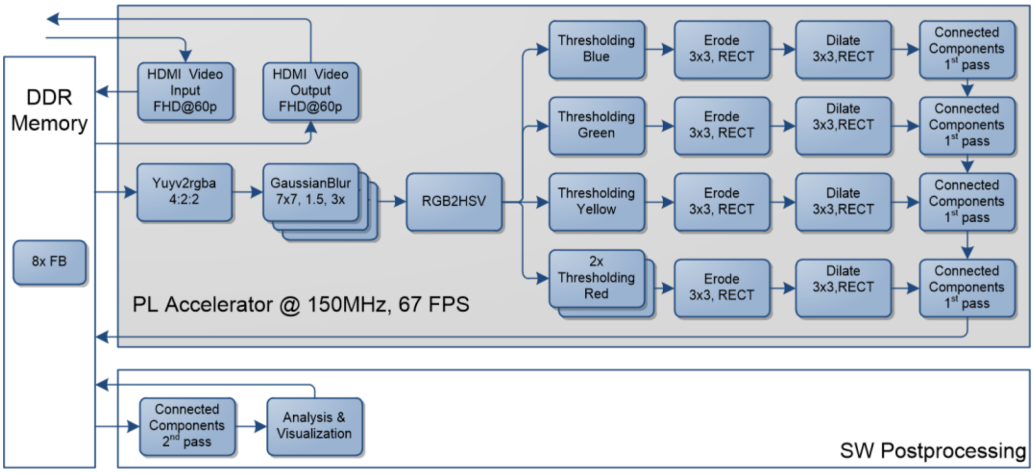


Image 1 - Ball detection demo application

The architecture of the resulting user application running on Zynq is shown in Image 1. The individual operations tested by this demo are:

1. Yuyv2rgba 4:2:2 – used platform captures camera frames in YUV422 format. This operation converts it to RGB.
2. Gaussian Blur – This operation filters out camera noise. The core makes smooth boundaries for consecutive thresholding.
3. RGB2HSV – Conversion of color space to hue-saturation-value format. In this format colors may be more easily separated.
4. Thresholding – Conversion of HSV to binary image.
5. Erode & Dilate – Morphological operations to clean the thresholded output.
6. Connected component 1st pass – Searches for continuous regions and their connections.
7. Connected component 2nd pass – Merging of identical regions found in 1st pass.
8. Analysis and Visualization – Filtering the Connected component output by applying limits to detection bounding box area and side ratio. The block also displays detected balls as graphic overlay to the processed image.

The demo shows, that design time resources developed by UTIA in FitOptiVis WP3 (Debian OS for TE0820 with HDMI I/O) can be used to implement hardware accelerated image processing for UC8. The implemented demo can be used as a starting point to implement accelerated LED position localizer. The demo as it was presented on FitOptiVis meeting in Prague and on Embedded World 2020 can be seen in Figure 9-7.

Figure 9-7: Demo as presented on FitOptiVis meeting in Prague

9.3 Use Case 8 Metrics

9.3.1 User Needs

User Need	Validation method	Status
Calibration of newly placed, moved or reconfigured robots with regards to world or workpiece coordinate system	Check, whether (and how) the developed prototype can perform the desired task – robot calibration. Considered indicators are mainly precision, user experience, process duration and system price.	The system as a whole will be evaluated when it is completed. Planned for M36.

9.3.2 Use Case Requirements

ID	Use Case Requirement	Verification method	Status
A1	EtherCAT connectivity allowing direct manipulator/robot control	Check that the robot can be controlled using EtherCAT protocol	Implemented by design – the hosting REXYGEN real-time control system contains EtherCAT drivers.
A2	Remotely controllable LED markers	Check that the LED markers can be remotely controlled with the required precision.	Remote LED markers system is implemented and validated.

ID	Use Case Requirement	Verification method	Status
A3	Possibility to connect at least 2 cameras - stereovision	Check that the control platform can manage 2 video input sources/cameras.	Yes, by design - the hosting REXYGEN real-time control system supports multiple cameras.
A4	Underlaying real-time control system allowing time-aware robot control and binding to technology	Check that the control platform can interface real-world technology – i.e. availability of input/output drivers for standard industrial protocols and corresponding sensors and actuators.	Implemented by design - the hosting REXYGEN real-time control system provides motion control and drivers to control various robot types.
D1	Modular, configurable solution	Demonstrate variability of the solution and its modular structure	Yes, by design – REXYGEN real-time system is highly configurable, modular system.
B1	Camera shutter synchronized with LED markers	Prove synchronicity between camera shutter and LED flashes	Camera integration with the LED controller was finished during M24. Detailed validation is planned for M25-M27.
B2	Robot control according to definition file describing robot structure and joint movements	Validate actual robot movement against the defined trajectory	Functional - Implemented and validated.
F1	The system moves robot joints while observing the marker LEDs	Check that the system is acquiring camera images during robot movement	To be checked after system integration.
F2	Marker LEDs are periodically switched on/off, camera taking synchronous pictures with that action and generating differential images	Check that the system can correctly generate differential images	Currently in implementation, validation planned for M25-M27.
F3	3D localization of LED markers from stereovision	Compare results obtained with the proposed method to results obtained from existing professional solution (for example OTUS tracker)	3D localization from stereovision component will be implemented during Y3. Validation will follow after implementation.
F4	Fitting robot model to measured 3D point cloud	Calibrate robot with known parameters and compare the obtained results with the actual robot parameters	Mathematical apparatus basically finished, SW implementation in progress.
F5	Publish computed robot parameters to the robot controller	Check that the computed robot parameters have been correctly transferred to the robot controller	To be checked after system integration.
NF1	Robot/manipulator emergency stop	Check that the emergency stop is able to stop the robot and bring it to safe state at any execution point of the calibration process	To be checked after system integration.
NF2	The system issues error/warning upon detection of unexpected marker movements/locations	Check that invalid spatial constellations raise error/warning message	To be checked after system integration.

ID	Use Case Requirement	Verification method	Status
P1	Sufficient computational power to perform 3D localization from stereovision and evaluate robot parameters in reasonable time (~max few seconds)	Check CPU utilization and algorithm execution time and ensure that it is in expected range	Sufficiently powerful host device has been selected. Exact system utilization will be validated after system integration.

9.4 End user feedback and recommendations

The results of the end-user-board survey show that end users generally acknowledge the need of the presented robot calibration solution, especially for human-collaborative robot scenarios. When considering the partial task of object localization and tracking, both 2D and 3D localization/tracking are important to the end users. Similarly, 6DOF tracking (i.e. both position and orientation) are matter of interest. Some end users were interested in tracking of moving objects in real time. However, the placement of active LED markers may be limiting in some cases.
The survey indicates that the development is quite well aligned with the end user needs. Also in the future it is planned to listen to the needs and comments of the potential/actual users, extract new beneficial ideas and integrate them as new features into the final product.

10 Use Case 9 – Surveillance of a Smart Grid Critical Infrastructure

10.1Full Demonstrator Description

The full demonstrator should emulate smart-grid equipment controlling remote electrical power substations with the assistance of smart video-surveillance and biometric recognition applications. The ultra-reliability demanded by critical infrastructure is provided by the High-availability Seamless Redundancy protocol, enforced by potential harm protection given by smart surveillance and biometric recognition. Compact cooperation of these heterogeneous subsystems, as well as the interconnection of distributed equipment demanding mixed-critically traffic is provided by Time Sensitive Networking (TSN).

For the sake of clarity, this section is structured in two subsections: The first one details the subsystems integrating the demonstrator, while the second section describes a possible deployment of this full demonstrator.

10.1.1 Subsystems

This subsection describes the different platforms used to implement the four subsystems conforming the Use Case: Smart-Grid, Video-Surveillance, Biometric recognition, and Time Sensitive Networking (TSN).

10.1.1.1 Smart-grid subsystem

The smart-grid specific devices are responsible of electrical power transformers and switches present on remote electrical substations. This equipment demands deterministic, bandwidth guarantee and best-effort Quality of services between smart-grid peers, video-surveillance and biometric recognition stations and central station for control and monitoring. In addition, the inclusion of an HSR ring network for communications within the Smart Grid guarantees that the traffic managed by the TSN network is distributed correctly through all the RTUs involved in the proper operation of the Smart Grid. Achieving the critical data generated in the Smart Grid and in the Smart Surveillance to the Smart Grid will not be lost in a simple network failure.
The RTU can work in different operation modes:

- Remote control operation mode: The control of the RTU and its acquisition blocks is possible to be performed from the highest level of the Smart Grid (SCADA, RTU Master Station, …)
- Local control operation mode: The control of the RTU is done directly by the CPU, and the operation from another remote place is not possible or refused.

These types of operation modes are needed for safety reasons. To prevent any command is executed in an electrical substation by mistake when maintenance or setting up works are being carried out locally

Smart Grid	
Name	**Description**
HUe	HUe is a high-performance CPU developed by Schneider Electric for RTU systems. HUe performs the control functions for the complete system, centralizes the information acquired by other modules, and executes the programmable logic control, communication protocols and user-specific applications. HSR Network component is implemented in the HUe.
AB DO	Acquisition block developed by Schneider Electric for RTU systems, providing 8 digital outputs. These digital outputs are relays that the HUe manages according to the needs of the Smart Grid.
Hirschmann RSP35	HSR RedBox. Element to interconnect the HSR ring network with the TSN network.

10.1.1.2 Video-surveillance subsystem

The Video-surveillance subsystem performs person tracking of an area surrounding sensible power electrical equipment. The video surveillance should trigger messages whenever an intrusion occurs to Smart-grid equipment. Besides, the biometric recognitions should be triggered to grant access of authorized operators.

Smart Video-Surveillance	
Name	**Description**
Jetson TX2	System on Module. Camera control. Edge processing.
Jetson AGX Xavier	System on Module. Camera control. Edge processing.
Manta G-125C	CCTV IP camera.
PC Ubuntu 18.04	Central station. Cloud processing.

10.1.1.3 Biometric recognition subsystem

The biometric recognition subsystem is triggered by the video-surveillance subsystem whenever a person is tracked on a restricted area. These equipment reports the central station if the access is granted or denied.

Biometric recognition	
Name	**Description**
Jetson TX2	System on Module. Camera control. Edge processing.
5 MP Fixed Focus MIPI CSI Camera	High-resolution camera to detect face patterns

10.1.1.4 Time Sensitive Network (TSN) subsystem

The TSN provides a well-known connection interface (Ethernet 100/1000-Base-T) and isolation of the multiple mixed-critically traffic traversing the network, including deterministic traffic, required by Smart-grid applications. Time synchronization is provided to the different subsystems to facilitate the cooperation between them as well as coherent monitoring between all of them.

TSN	
Name	**Description**
TSN bridge	MPSoC Xilinx Zynq-7030 4 interface stations with routing and synchronization capability. 4 different priority levels. Time-driven traffic scheduling. Frame pre-emption Standard compliance: 802.1Q, 802.1Qbv, 802.1Qbu, 802.3br.

10.1.2 Final demonstrator deployment

The final demonstrator shows the integration of the different subsystems: smart grid, video-surveillance, biometric identification, and Time Sensitive Network.

The demonstrator includes: at least 2 edge nodes for the vision-based subsystems (Jetson TX2 platforms) attached to at least 2 cameras to perform the biometric identification and the video-surveillance tasks, the time sensitive communication network that includes a topology with at least 3 bridges (2 for the vision nodes and an additional one for the smart grid traffic) plus an additional connection to the server that centralized all the information and decision making, and the node for the smart grid system that also includes the additional HSR ring.

The video-surveillance system allows for setting up the secured perimeter, tracking the potential targets (operators or others) also along the multi-view camera system, and logs all the events that happen around the monitored secured perimeter. The software for the monitoring and the communication with the smart grid are running on the Jetson TX2

nodes. Vision nodes start working when a signal is received from the centralized server, starting all the services and applications, and updating any information from the cameras (if changes in the camera settings occurred).
The centralized server runs continuously, listening from the different nodes and showing the video streams from the cameras, and the information about the secured perimeter, including tracking information and person detection events.

The Smart Grid network traffic is integrated in the TSN network. Simulating communication that manages remotely the control of a digital output of a RTU. The digital output signal will be activated and deactivated with a frequency of 5 seconds.

Interoperability network traffics between Smart Grid systems and Smart Video-Surveillance subsystems are integrated in the TSN network. Modbus Master-Slave communication is developed to monitor the status of the alarms generated by the Video-Surveillance system. The video-surveillance system continuously monitors the secured perimeter, detecting people in the area. Once a person is detected, it is tracked and an alarm is triggered, for example, when this person enters the secured perimeter around the electric substation.

When this alarm is triggered by the Video-Surveillance system, it is transmitted to the Smart Grid system using this Modbus protocol. Then, the RTUs of the Smart Grid system will change to local control operation mode. The output digital signal controlled remotely must stop, remote command for activation or deactivation is not allowed in this case.

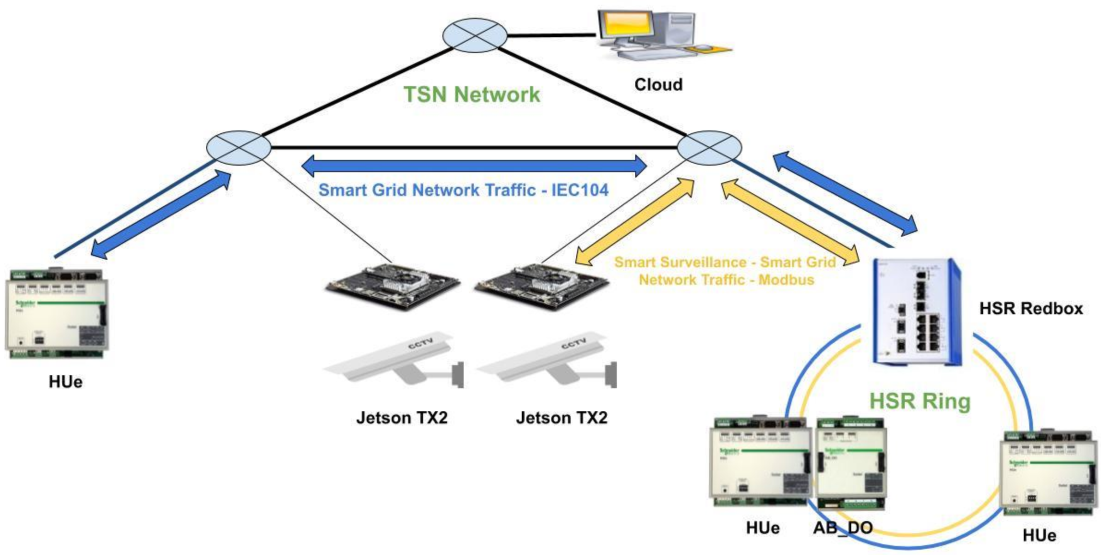


Figure 10-1. Overview of the final demonstrator

10.1.3 Reference Architecture

The next diagram shows the reference architecture for the System for Surveillance of a Smart Grid Critical Infrastructure. The complete QRML DSL description can be found in the Appendix Section.

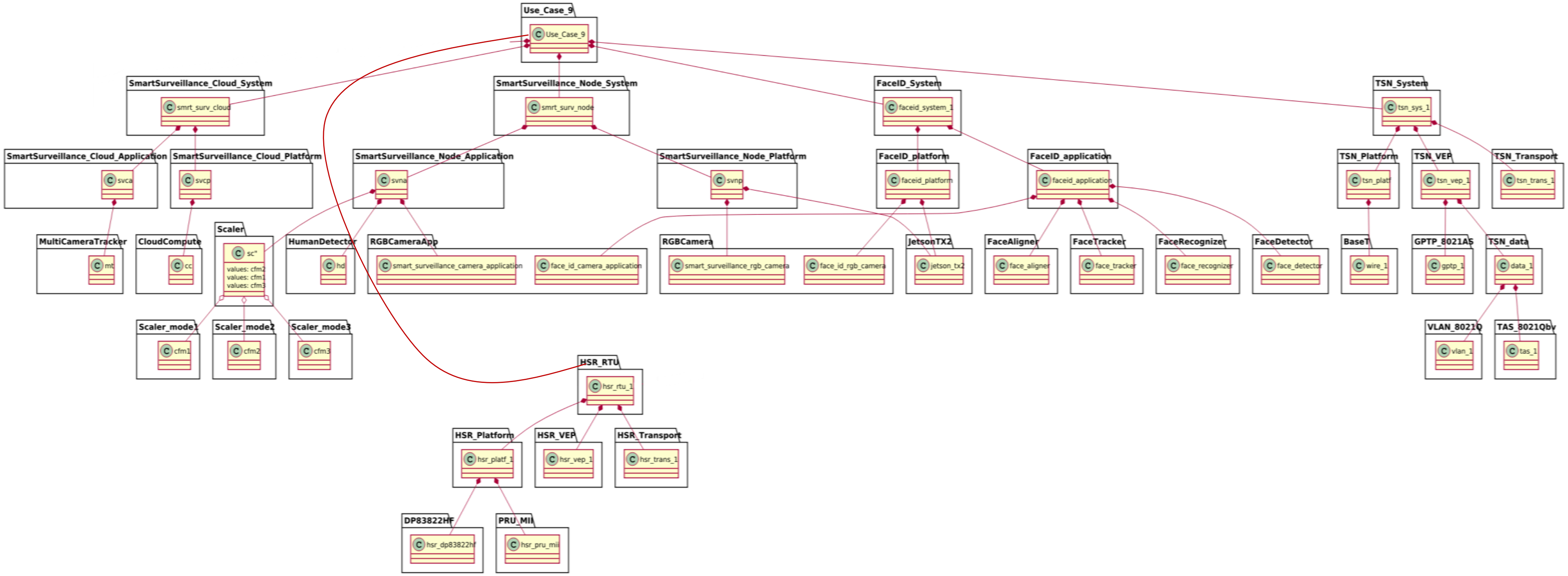


Figure 10-2: Reference Architecture

10.1.4 M24 Partial Demonstrator

The partial demonstrator shows that the different subsystems can be fully deployed without disturbing between each other. The corresponding deployment is shown in the figure below. Three TSN bridges will conform a ring topology. One node provides connectivity to the cloud node or central station, which retrieves monitoring from the different equipment conforming the demonstrator and supports major processing of the video-surveillance subsystem.

The rest of the equipment conform the arrangement present on a remote electrical substation. Two Smart-grid HUe's are connected via the TSN network between them and to video-surveillance edge processing nodes.

This arrangement will allow to demonstrate deterministic communication between Smart-grid HUe's concurrently to the video and control streaming shared between surveillance cloud and edge stations. Besides, the video-surveillance subsystem will have the capability to deliver to the smart-grid alert messages over the Modbus protocol.

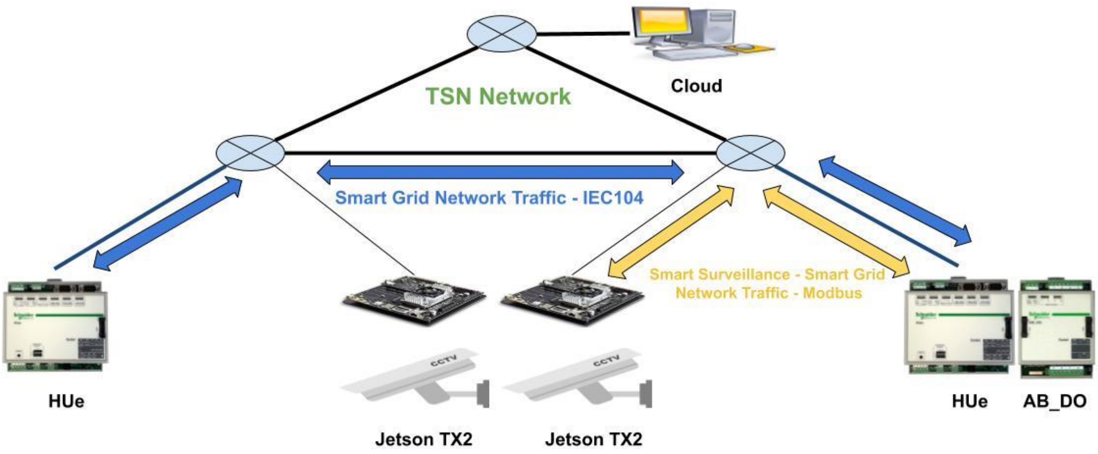


Figure 10-3. Overview of the partial demonstrator for M24

10.1.4.1 M24 to M36 GAP

The final demonstrator will provide full integration between the different subsystems conforming the use case. Besides, some additional developments will be presented:

- High-availability Seamless Redundancy on Smart-grid specific equipment.
- Fast switchover of the TSN time reference, in case of grandmaster failure

10.2Use Case 9 in FitOptiVis

Demonstrated Technology	WP in FitOptiVis	Role in Demonstrator	Date for demonstration (M24/M36)
Runtime monitoring	WP4	Metrics such as data bandwidth and energy for the video-surveillance subsystem are monitored on runtime	M24
Runtime reconfiguration	WP5	Reconfiguration of the video-surveillance subsystem based on the data bandwidth usage	M36
Real-time processing using hardware accelerators	WP5	Image-video processing in real time for robust tracking	M24
Assess deterministic timing in distributed systems and latency control	WP3	Provide time synchronization to HSR and optionally, other end stations.	M36
Integration of time-dependent signals and images	WP4	Provide time synchronization to demanding TSN stations	M36
Provide accurate timestamping mechanism	WP4	Time Synchronization accuracy	M24
Hybrid communication schemes based on TSN	WP5	Convergence of time-critical messages, control, and video streaming	M24
Use of components: Time Sensitive Networking (TSN) Person Tracking (UGR) HSR Implementation in RTU (SCHN)	WP5	Demonstrator building blocks	M36

10.3Use Case 9 Metrics

10.3.1 User Needs

User Need	Validation method	Comment
Integration of smart-grid and biometric identification, and video surveillance services for enhanced protection and	Verify concurrent time-critical and best-effort communication between smart-grid, identification, and video	Partial. Independent parts ready

monitoring of the CI perimeter with TSN	surveillance devices in the presence of burden traffic. Accurate time synchronization between Smart-grid and active vision devices.	
Hybrid communication between electrical substation devices and remote	Verify VLAN tagging/untagging of traffics object of RT-QoS. Differentiation and prioritization. Latency deviation of time-critical messages	Partial. In-lab verification results done. Integration pending
Biometric identification of operators	Operator recognition matching against database information	Validated
Guaranteeing smart grid communication through a Time Sensitive Networking	RTU communications analysis in a Time Sensitive Networking	Partial. In-lab deterministic latency verified in presence of burden traffic.
High-availability Seamless Redundancy in a RTU	Communications analysis in a simple failure simulation	Pending

10.3.2 Use Case Requirements

ID	Use Case Requirements	Verification Method	Comment
A1	The system is distributed by RTU components, TSN switches, and cameras/acquisition modules.	Communication capabilities between the different subsystems and the sensors/network components.	Partial. Communication with 3rd party tested, but integration with UC equipment not addressed.
A2	Time sensitive Network shall provide timestamp services and priority channels for different traffics	Coherence between user-provided configuration and provided QoS (i.e. no latency violation). Measure synchronization accuracy between time aware stations in the network	Partial. Functionality implemented and tested, but integration is pending.
A3	The system must integrate data from the smart grid RTU and, the video surveillance subsystems and the biometric identification system using the same network	Check the interoperability of the smart grid data and the smart video-surveillance vision data using common protocol, and with the biometric identification system.	Partial. Functionality implemented and tested, but integration not addressed.
A4	High-availability Seamless Redundancy (HSR) will be implemented for smart grid devices	No packet loss of smart grid commands for a simple failure	Pending

B1	Accurate timestamping mechanism shall be provided to network users (e.g. smart grid, and smart video surveillance, and biometric subsystems) to facilitate performance measurements and latency control	A good timestamp mechanism results on low time synchronization deviation.	Verified.
B2	Time distribution shall be provided to network users to allow coherent handling of time-dependent signals and images from heterogeneous pipelines	Distributed processing applications	Partial. gPTP implemented on TSN stations.
B3	The network users shall be able to specify multiple traffic types and require specific RT-QoS (bounded latency, guaranteed bandwidth) for each one	Measure latency deviation of time-critical messages in the presence of lower priority, burden traffics	Pending
B4	If an alarm is triggered, the operator can ask for an image from the camera that triggered that alarm	Check the triggering of alarms for specific behaviours considered potentially harmful/of interest if a person gets too close to the secured perimeter/critical physical components.	Verified.
B5	Improved substation supervision by combining smart grid, and smart video surveillance, and biometric identification data	Check monitoring routines including vision, identification, and smart grid data triggers	Partial. Energy monitoring consumption for different component alternatives and through the monitoring platform FIVIS
B6	Critical data availability must be ensured for different devices located in substations	TSN application layer should receive well-formatted frames accomplishing RT-QoS	Partial. In lab results

F1	The system must provide network time synchronization	Measure synchronization drift between multiple TSN stations.	Validated
F2	The system must ensure deterministic delivery of critical traffic in the network	Check deterministic delivery time deviation.	Validated
F3	The system must ensure bounded latencies and guarantee a specific data bandwidth for time-critical data in hybrid communication schemes.	Check that bandwidth is guaranteed in the presence of lower priority burden traffic.	Partial. Traffic scheduling for this use case not already addressed.
F4	The video processing smart video-surveillance and biometric identification subsystems must be able to detect intruders/operators or analyze the video input to detect suspicious behaviour	Check accuracy of action classification biometric identification of people and tracking accuracy	Validated. Accuracy of tracking done using different benchmarks and on site. For on-site testing we use 20 test persons.
F5	The video processing component must carry out tracking of intruders/operators around the secured perimeter of our Critical Infrastructure	Check accuracy of tracking and occlusion management	Validated. Accuracy of robust tracking moving from one camera to a different one in the multi-view system.
F6	Enhanced supervision of electric substation infrastructure must be ensured	Check monitoring routines including smart video-surveillance, biometric identification, and smart grid data triggers	Pending
P1	Deterministic latency jitter: less than 1 us.	Use timestamp mechanism to measure end-to-end latency.	Validated
P2	Network time distribution jitter: Less than 50 ns.	Measure time synchronization accuracy	Validated

P3	Support of multiple TSN streams	Check traffic requested RT-QoS accomplished in the presence of lower level burden traffics	Validated
NF 1	Zero-time recovery must be performed after communication failure	Cause a communication failure and check recovery.	Pending
P4	Real-time image processing should perform at approx. 20 fps	Check performance for tracking	Validated. Tracking performance in frames per second and scalability for the multi-view camera system
P5	Biometric identification should run at 300 ms/face recognition	Check performance for face detection	Validated. Face detection and recognition execution time verified for the multi-view camera system
P6	Biometric identification has an power cost of 5.5 W/frame	Check power consumption using a digital multimeter	Validated. Biometric identification system power consumption using battery as a source.

10.4 End user feedback and recommendations

During the Y1 end-user board meeting held in Eindhoven on September 10th 2019, the following topics were discussed. We include our comments about the End-user board feedback:

Comment by the board	**Discussion by UC9 participants**
“EUB member is missing to use audio (probably integrated in the cameras) for supporting the vision system, especially in areas with low light conditions or dark zones. My suggestion it is to integrate audio for supporting the vision surveillance system and also for detecting failure by classifying the frequency patterns”	Audio is not one of the multimodal cues used in FITOPTIVIS. For disambiguation, we are considering robust tracking mechanisms that let us follow targets around the secured perimeter and do re-identification e.g. when a person leaves the field of view of one camera and enters the field of view of another one, using video-only information.
“EUB member is missing energy brokerage where timestamping is crucial”	Although the suggestion is interesting, we would need an end-user that works in this specific field. We agree with the EUB member and also consider very accurate timestamping

	(nanosecond range for TSN) a crucial feature for this application.
"Personnel Safety monitoring and support is also high in interest"	We agree and have included scenarios in which we change the operating mode in the presence of operators. We also have a partner VISI that is doing operator biometric identification.
"Yes, we recognise such user needs. Perhaps we could add end-to-end QoS assurance at the communication level, i.e., latency, bandwidth, maximum error rate, availability… requested by the specific surveillance application. In many cases, such requirements are specified in standards or regulations"	This TSN solution provides such guarantees by means of synchronized time-aware traffic scheduling and shaping. This allows the convergence of video streaming for surveillance, control and smart-grid time-critical messages on the same network.
"Other solutions [for synchronization] can be checked, as GNSS based synchronization. GNSS as the primary source, can be susceptible to intentional and unintentional interference due to the low power signal propagation. Nevertheless, new advance on GNSS technology could overcome some of their today limitations"	Although this topic is not addressed in this use case, the gPTP grandmaster usually relies on atomic clocks, GNSS, terrestrial radio, etc. as time source. In fact, these time sources take precedence over the conventional time source (the internal oscillator) on grandmaster election.

Our expectation and possible questions for the Y2 meeting are as follows:

- Usability: How interesting is the adaptation of the secured perimeter to the needs of the user? I.e. an application that lets the user draw the secured perimeter around the substation, with a bird-eye view of the area.
- Devices: How relevant is power consumption in this kind of sites? Electric substation can be isolated and also, access to the network may not be easy if applied to other critical infrastructures such as e.g. water reservoirs.
- How interesting is using a biometric identification system instead of RF cards? Think about flexibility, robust identification in case of stolen keys, etc.

11 Use Case 10 – Autonomous Exploration

11.1 Full Demonstrator Description

11.1.1 Introduction

For the past few years space debris has become an increasing cause of concern for space missions. Different kinds of space junk are flying around earth at different heights. Although there is a really low probability of any spacecraft hitting any of this debris, the increasing number of megaconstellations of more than 100 satellites is increasing the probability of a crash.

The different spatial agencies have started to think into different solutions for freeing the orbits from space junk. One of the possible approaches is to launch satellites capable of capturing debris and, at the end of the mission, burn itself with the debris into the atmosphere.

Space junk comes in different kinds and shapes. One of the biggest pieces of space garbage that can be found are dead satellites. With an average life of seven to fifteen years there are hundreds of decommissioned satellites flying around earth. The Autonomous Exploration final demonstrator will consist of a video unit capable of identifying satellite models. A satellite with this video unit could be capable of identifying decommissioned satellites and grab them in order to burn them when falling into earth or to drive them to a graveyard orbit.

The video unit will be capable of reconfiguring itself with different configurations that could allow different performances and ways of consuming power in order to adapt to the several environmental conditions that could be found during a space mission.

11.1.2 Hardware Platform

The final hardware platform that will be used in the M36 demonstrator will consist of:

- ZCU102/ZCU104 Evaluation Boards: These evaluation boards have a Zynq UltraScale+ MPSoC. The MPSoC consists of a processing system and a processing logic (FPGA). The processing system has, in both platforms, a quad-core ARM Cortex – A53 and a dual-core ARM Cortex-R5.
  The main difference between the MPSoC in the ZCU102 and the ZCU104 evaluation kits is the processing logic (PL). The FPGA is slightly bigger on the ZCU102 but for this demonstrator both FPGAs serve its purpose.
  This MPSoC is not used in the space industry and it is not planned to use it in future missions. However, the combination of an FPGA with several processors is usually present on the different data processing chains of the spacecraft making this video unit representative for demonstration purposes.
- Leopard Imaging LI-IMX274MIPI-FMC: This module is an FMC camera with a CMOS sensor that is compatible with the MPSoC evaluation kit in terms of connection through an FMC port. It uses a CMOS Image Sensor manufactured by Sony, the IMX274. The interface is MIPI, which is widely used on the mobile market.
  CMOS sensors are not common in the space industry where CCD sensors are actually the standard. Several CMOS sensors are currently being used for use in space applications by TASE. The IMX274 could be used on low-cost applications based on COTS components. This kind of missions are not the main

target for TASE, however using this sensor for demonstration purposes provides expertise for using other CMOS sensors.

- NVIDIA Jetson Nano: A powerful platform consisting on a quad-core ARM-A57 running at 1.43GHz with a 128-core Maxwell GPU. The Jetson Nano has been developed in order to run multiple neural networks in parallel for applications like image classification, object detection … .
- NVIDIA Jetson TX2 (optional): if some complex neural networks require additional computation capability, an additional element will be added to the platform. This board is a more powerful device than the Jetson Nano. It is built around a 256-core NVIDIA Pascal™ GPU and loaded with 8GB of memory and 59.7GB/s of memory bandwidth. This platform will be used in case the Jetson Nano cannot support the selected CNN.
- Intel RealSense Depth Camera D435i: A stereo camera that calculates depth and has also a real image RGB camera. It also has an infrared projector that projects an IR pattern to improve depth accuracy in scenes with low texture.
- A host computer connected to both processing platforms (Jetson Nano and ZCU MPSoC) through ethernet.

11.1.3 Overview of the M36 demonstrator

The final demonstrator will use the hardware described in Section 11.1.2 for processing a video stream in order to identify two different satellite models. The two models will be identified by the video-processing system in such a way that they could be distinguished between them and from other kinds of objects.

The final demonstrator will have two different cameras. The LI-IMX274MIPI-FMC, based on the IMX274 sensor by Sony, will be the RGB camera of the system. This is a camera based on a CMOS sensor and will be controlled by the FPGA logic of the Zynq UltraScale+. The image capture will be performed by the FPGA that after performing some preliminary pre-processing of the image will send it to the NVIDIA Jetson Nano. In this platform the RGB images obtained with the FPGA and the depth pictures obtained by the Intel RealSense Depth Camera will be mixed and given as input to a Convolutional Neural Network (CNN) that will be used to recognize the target satellite models. The processed images will be sent to a host computer that will act as On Board Computer (OBC) of the spacecraft. This computer will be also in charge of monitoring the power consumption of the platform and to simulate the environment that affects the different setpoints of the whole video chain.

For the identification of the satellite models a set of 100 images from each satellite will be used for training the pre-trained convolutional neural network in order to obtain a high precision on the identification process. A pre-trained model is a model that was trained on a large benchmark dataset to identify a set of features that improve the classification problem. Accordingly, due to the computational cost of training such models, it is common practice to import and use pre-trained models from published literature (e.g. VGG, Inception, MobileNet [8]). Hence, instead of starting the training process from scratch, it is started from features and patterns that have been learned when solving a similar problem.

The system will be able to reconfigure itself at runtime. The possible states of the reconfiguration will be determined by the OBC/host computer that will simulate different environmental scenarios that will change the behavior of the system taking into account different setpoints that will allow to lower the communication data rate, power consumption and general performance of the system. Some of the components of the system will change from one device to another. For example, the image compression could be implemented either on the PL of the FPGA or on the Jetson Nano (Edge).

11.1.4 Reference Architecture

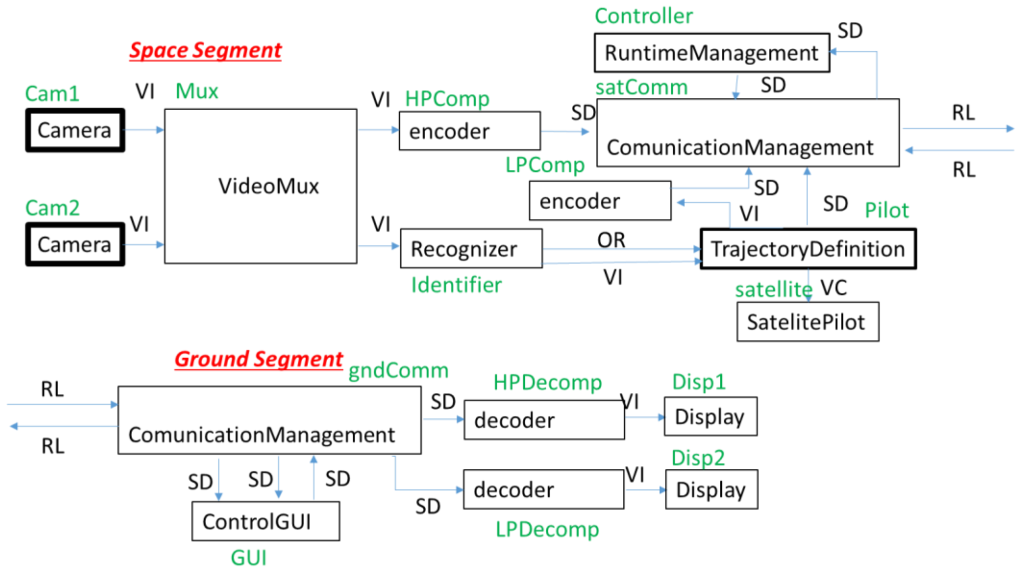


Figure 11-1: High-level reference architecture for UC10 demonstrator.

Figure 11-1 presents a complete view of the UC10 demonstrator. It has two parts: the space and the ground segments. The space segment includes all modules that will be integrated in the satellite while the ground segment integrates the remote control system. Both subsystems are connected with a radio link.

The space segment includes two video chains. The first chain (upper path in Figure 11-1) compresses a high-resolution image while the second chain (lower path) recognizes a satellite in the image and provides a low-resolution image with the recognition result. The target satellite position is used to guide the host satellite to the target.

The *Camera* component models color (RGB) and depth cameras. The demonstrator integrates two cameras: a high-resolution color camera (cam1) and a depth camera (cam2). The depth camera also provides a color image (RGBD camera). The *Camera* component includes the image sensor and the functionality that the sensor requires to generate a video frame. This component provides a video interface that sends raw video/depth frames to other components. It has several implementations from FPGA-based hardware implementations to software solutions.

The *VideoMux* component is a video multiplexer that sends camera's frames to the image recognition or image compression chains. It also combines the RGB and depth frames and performs basic image transformations such as image scaling and colour conversion. This component supports several configurations that modifies the video path, image scaling or colour conversion. The demonstrator also includes hardware and software oriented implementations.

The *Encoder* component implements an image compression algorithm. The demonstrator will implement the CCSDS 122.0-B2 image compression standard. This component also provides jpeg compression. The demonstrator includes hardware and software implementations of the space image-compression standard.

The *Decoder* component decompress the image that was compressed by the encoder. The demonstrator will only provide software implementations of this component.

The *Recognizer* component identifies a satellite in an image using a CNN-based algorithm. It uses colour and depth images to determine the position of the target satellite. It always provides the position of the satellite in the image. If a depth image is provided, the component will generate the distance to the target.

The *TrajectoryDefinition* component will define the path that the host satellite should follow in order to reach the identified (target) satellite. It uses the output produced by the *Recognizer* (satellite position and distance) to define the host satellite trajectory.

The *SatellitePilot* component will receive *TrajectoryDefinition*'s orders and move the satellite accordingly.

The *CommunicationManagement* components will communicate the satellite with the ground module. They act as a temporal multiplexors sending several data streams from the satellite to ground and vice versa. This component provides data to the satellite radio link.

The *RuntimeManagement* component evaluates real-time quality traces (e.g. power consumptions) and modify the system configuration to comply behaviour constraints.

The *Display* component presents the images that are captured in the satellite.

The *controlGUI* component is responsible of the remote management of the satellite. It also monitors the satellite parameters.
The system has been described with an extension of the QRML DSL that has been defined in WP2 (SDSL, Service oriented DSL). The extension supports service definitions with some additional keywords, such as “providesinterface”. The description (about 1000 lines of code) will be included in an additional section. The next image presents the specification of the space segment.

```
component space_segment {
        outputs radiochannel outp;
        providesinterface RadioLink ip1;
        requiresinterface RadioLink ir1;
        requires rfcap reqrfcap;

        component CameraRGB cam1;
        component Camera3D cam2;
        component VideoMux mux;
        component Encoder hpcomp;
        component Encoder lpcomp;
        component Recognizer identifier;
        component TrajectoryDefinition pilot;
        component SatellitePilot satellite;
        component CommunicationManagement satcomm;
        component RuntimeManagement controller;
```

```

        //Service Communication
        ip1.cam1 connected to ir1.mux;
        ip1.cam2 connected to ir2.mux;
        ip1.mux connected to ir1.hpcomp;
        ip2.mux connected to ir1.identifier;
        ip1.hpcomp connected to ir1.satcomm;
        ip1.identifier connected to ir1.pilot;
        ip2.identifier connected to ir2.pilot;
        ip1.pilot connected to ir1.satellite;
        ip2.pilot connected to ir3.satcomm;
        ip3.pilot connected to ir1.lpcomp;
        ip1.lcomp connected to ir2.satcomm;
        ip1.controller connected to ir4.satcomm;
        ip2.satcomm connected to ir1.controller;
        ip1.satcomm connected to ip1;
}
```

Next figures show the reference architecture for the use case. In this case the system is split into space and ground segment. The figure shows all the components that are used on UC10 as well as their different implementations.

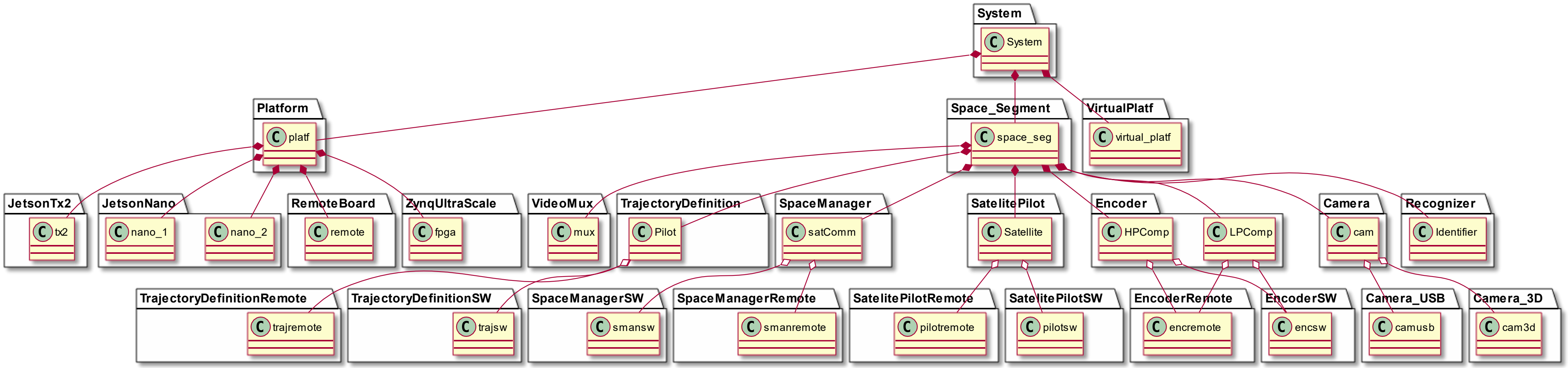


Figure 11-2 Reference Architecture (Space segment)

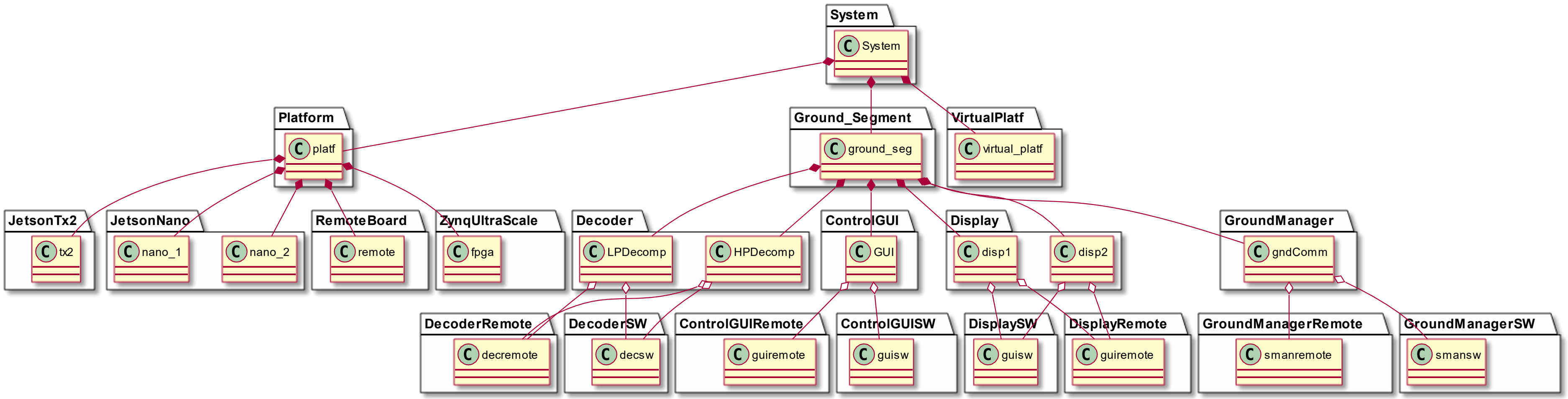


Figure 11-3 Reference Architecture (Ground segment)

11.1.5 M24 Partial Demonstrator

For the M24 partial demonstrator, an initial implementation of the video chain has been developed. The core functionality of the demonstrator (satellite identification) has been achieved.

The camera that is currently being used is the RGB camera present on the Real Sense camera. The images of this camera are fed to the Jetson Nano where the CNN acts in order to identify the satellite models that face the camera. The image then is sent to an HDMI monitor that displays the identified objects.

Several CNN models (e.g: Faster R-CNN and SSD MobileNet [8][9]) have being explored in order to decide which one is better suited for the UC10 demonstrator. Small and light Object Detection CNNs such as SSD MobileNet [8][9] offer fast performance on both Jetson Nano and Jetson TX2. More complex networks like Faster R-CNN increase MobileNet's accuracy but won't run on Jetson Nano. However, they will only run on Jetson TX2 but slower.

The partial demonstrator will include a software implementation of all the system components. It will also demonstrate runtime reconfiguration. For example, the encoder and decoder components will modify the compression algorithms from jpeg to CCSDS 122 or vice versa.

In the partial demonstrator, some components have limited functionality. For example:

- The *Camera* component supports some real sense cameras and webcams with USB connection.
- The *VideoMux* component supports an input camera (Real Sense Camera) that provides frames to all system video chains.
- The *Encoder* implements a preliminary version of the space image-compression standard with limited performances.
- The *Recognizer* will not recognize satellites. It will recognize person and some types of objects (e.g. monitors).
- The *RuntimeManagement* implements a basic algorithm that uses simple metrics. Additionally, the platform performance monitors that are integrated in this component will have limited functionality. These monitors provides real-time performance estimations of hardware platform parameters such as power consumption.
- The *ControlGUI* is a preliminary version of the remote control panel with basic functionality.

The trajectory definition, satellite pilot and communication management components provide a first version that will be improved in the M36 demonstrator.

11.1.5.1 M24 to M36 GAP

From M24 to M36 the focus of the work will be on two main activities.activities. On the first place the Hardware Capture of the image from the LI-IMX274MIPI-FMC camera needs to be developed with the preprocessing embedded on the block that controls this camera. The compression block for the processing logic (PL) of the MPSoC needs to be developed also. Additionally, other components (e.g. VideoMux) will provide different implementations (e.g. FPGA hardware implementation) or algorithms (e.g. RuntimeManagement component).

Once that these blocks are developed, the focus will be on the reconfiguration of the system. The decision tree for the alternative scenarios that will be present in the demonstrator needs to be developed.

Finally, measurements on the different metrics that will allow to prove the high efficiency of the video processing system will be performed during the final months of development.

11.2 Use Case 10 in FitOptiVis.

Include in this section a table with a relation between your demonstrator and FitOptiVis technologies.

Demonstrated Technology	WP in FitOptiVis	Role in Demonstrator	Date for demonstration (M24/M36)
Reference architecture	WP2	Provide the architecture of the demonstrator	M24
SDSL	WP2	Component description	M24
Parallelization with OpenMP	WP4	Speed up the processing of images	M24
Runtime reconfiguration	WP4	Real-time system reconfiguration	M36
Real-time processing using hardware accelerators	WP5	Image-video processing in real time for object recognition	M24
Use of components: High Performance Space Image Collection and Processing (TASE) Image Transmission Interface (TASE) Reconfigurable video capture and processing (UC)	WP5	Building blocks of demonstrator	M36

11.3 Use Case 10 Metrics

11.3.1 User Needs

User Need	Validation method	Comment
Autonomous Navigation	Test the autonomous platform on a simulated environment and check if it can navigate and change according to the different conditions	Pending
Smart Data Compression	Check that different kinds of compression are applied related with the several setpoints	Pending

User Need	Validation method	Comment
	determined by environmental conditions.	
Smart Object Recognition	To Be Defined	Partially Validated

11.3.2 Use Case Requirements

ID	Use Case Requirement	Verification method	Comment
A1	Input data from camera (or set of cameras) to the processing system	Check that the data from the camera is received properly in the FPGA	Verified
A2	Transmission of collected data (images) to RAM memory (DMA/VDMA)	Compare the stored data against the received data	Verified
A3	Collection of the data (images) from the Processing Logic (FPGA fabric)	Check that the produced images show the field of view of the camera	Verified
A4	Processing of the video/image data for object recognition through different image processing algorithms (Harris, ANMS, SIFT…)	Compare the extracted feature points of the used pipeline against results extracted from software running on a computer with OpenCV	Partially verified
A5	Storage of processed data (images) into RAM Memory (DMA/VDMA)	Compare the stored data against the received data	Partially verified
A7	Environmental analysis through several sensors (battery, sunlight…)	TBD	Pending
A8	Transmission of processed data (images & environment) through Ethernet to other system components to perform downlink/uplink (to earth/satellite)	Data reception on host computer and image representation	Pending
B1	Detection of hazards (from images and environment) and paths to follow based on processed data	Compare the extracted feature points of the used pipeline against results extracted from software running on a computer with OpenCV	Pending
B2	Data storage based on quality of processed images. Dump data without useful information to prevent data-storage / data transmission bottleneck. (Smart Data Storage).	TBD	Pending
B3	Adaptation of data transmission based on data quality, link, bandwidth availability and environmental conditions	Measure the bandwidth of the data link and rate of obtained images.	Pending
B4	Object classification based on processed images	Check that objects are identified correctly	Pending

ID	Use Case Requirement	Verification method	Comment
B5	Autonomous decision and adaptation based on classified objects and environmental conditions (e.g. reduction of analyzed data with low battery, change of data analysis effort -use simpler algorithms- to reduce battery usage…)	TBD	Pending
F1	Reconfiguration of the system to adapt to environmental (data link/battery/…) conditions	Check that the processing chain changes to the different setpoints in relation with the environmental conditions.	Pending
F2	Classification of data (images) in terms of quality and priority	TBD	Pending
F3	Report of extreme environmental conditions (low battery, low sunlight, radiation induced failures…)	Receive report on host computer	Pending
NF1	System should be available to perform data transmission without loss of data transmission window (4 windows of 2 minutes per day)	TBD	Pending
NF2	The system shall prioritize tasks taking into account environmental conditions	TBD	Pending
NF3	The system shall be able to give a status report each time a data link is available	TBD	Pending

11.4End user feedback and recommendations

The feedback received by the end user of this use case encourages to keep working on the direction that this use case is headed. Several interesting applications of the proposed video-processing chain using FitOptiVis technologies have been found by the end user besides the proposed scenario of the actual Use Case. It could be used as a general architecture for any space application that needs on-orbit processing or for planetary exploration purposes.

The main concern commented by the end user of this use case is that this kind of video processing relies on Commercial Off The Shelf components. This is why the current scenario is targeting the so called *new space* missions which rely on lower cost and quality components that allow higher performance but also drive missions to deal with higher risks instead of targeting traditional space segment.

12 Conclusions

This deliverable serves its purpose as a showcase of how the different technologies developed in the FitOptiVis project can be put into application in different industrial scenarios that allow the implementation of such technologies in fields that vary from medical applications to the space market.

Every use case has successfully exposed the road to achieve its applied demonstration of the FitOptiVis toolchain with one or more scenarios that will be ready in M36 (May 2021) and how, in order to achieve that final demonstration, partial demonstrators for M24 have been developed.

This document also shows how the different demonstrators act as the glue logic for this project. The different technical work packages (2 to 5) are in charge of developing various technologies and this work package is the integration stage of all of them into different image processing chains.

Finally, it also shows how the different use case providers have taken into account the comments addressed by the end users that made very interesting recommendations in the First FitOptiVis End User Workshop that will allow a better adoption of the FitOptiVis technologies into different markets.

13 References

[1] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuca, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, “Emerging mpeg standards for point cloud compression,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 1, pp. 133–148, March 2019.

[2] S. Schwarz, M. Pesonen, and J. Nurminen, “Real-time decoding and augmented reality playback of the emerging Moving Picture Expert Group V-PCC Standard”, International Broadcasting Convention 2019, pp. 3-7, 13-17 September 2019.

[3] PCC Test Model Category 2v0, ISO/IEC JTC1/SC29/WG11 N17248, Macau, China, October 2017.

[4] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.168+SCM-8.7
[5] H. Hoppe, T. DeRose, T. Duchamp, J. A. McDonald, and W. Stuetzle, “Surface reconstruction from unorganized points,” in Proc. SIGGRAPH, 1992, pp. 71–78.

[6] Nokia Technologies, “Video Point Cloud Coding (V-PCC) AR Demo,” https://github.com/nokiatech/vpcc

[7] Euee Jang, Marius Preda, Khaled Mammou, Alexis Tourapis, Jungsum Kim, Danillo Graziosi, Sungryeul Rhye, and Madhukar Budagavi, “Video-Based Point-Cloud-Compression Standard in MPEG: From Evidence Collection to Committee Draft”, IEEE Signal Processing Magazine, May 2019, pp. 118-123.

[8]https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

[9] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).