ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

Mediate

Patient Friendly Medical Intervention

DELIVERABLE SUMMARY
D2.1.3 – System Architecture Specification

Project number:	ITEA 09039
Document version no.:	v 1.0 Summary

ITEA Roadmap domains:
Major: Group

ITEA Roadmap categories:
Major: Content & knowledge
Minor: Interaction

HISTORY

Document version #	Date	Remarks
V0.1	16-09-2013	Summary, based on D213, version 0.1

TABLE OF CONTENTS

1 INTRODUCTION 5
1.1 Purpose 5
1.2 Scope 5
2 EXECUTIVE SUMMARY 7
3 STAKEHOLDERS AND CONCERNS 9
3.1 Clinical users 9
3.2 Insurance companies 9
3.3 Researchers 9
3.4 Developers 9
3.5 Manufacturers 9
3.6 IT administrators 9
3.7 Helpdesk 9
4 ARCHITECTURAL VIEWS 11
4.1 Structural view 11
4.1.1 Overall architecture 11
4.1.2 Interventional Controller Architecture 12
4.1.2.1 Workflow Controller 13
4.1.3 Sensors architecture 17
4.1.3.1 Steering and Navigation 18
4.1.3.2 X-Ray 18
4.1.3.3 Ultrasound 19
4.1.4 External system architecture 19
4.1.4.1 Pre-Interventional reconstruction 21
4.1.4.1.1 Pre-Interventional reconstruction interface 22
4.1.5 Video infrastructure architecture 23
4.1.5.1 Local Video Distribution 23
Inside the intervention room, the critical requirements are: 24
4.1.5.2 Remote Video Distribution 25
4.2 Behavioral view 30
4.2.1 Minimal invasive Brachy therapy 30
4.2.1.1 Brief Description 30

4.2.1.2 Basic Flow for Image guided conformal Brachy 31
4.2.1.2.1 Basic minimal invasive Barchytherapy workflow 31
4.2.1.2.2 Alternative minimal invasive Barchytherapy workflow 32
4.2.1.3 Architecture for minimal invasive treatments 32
4.2.1.3.1 IGIT architecture 32
4.2.1.3.2 Applications and workstations 32
4.2.1.3.3 Communication between the applications 33
4.2.1.4 The Brachy therapy applications 34
4.2.1.4.1 Brachy planning system 34
4.2.1.4.2 Implant guiding system 35
4.2.1.4.3 Implant reconstruction 36
4.2.1.4.4 Connection of afterloader to implant 36
4.2.1.4.5 Treatment delivery 37
4.2.2 Pre-Interventional reconstruction 37

4.3 Physical view 38
4.3.1 Overview 38
4.3.2 Workspot Examination room 38
4.3.3 Workspot Control room 38
4.3.4 Additional Workspots 39
4.3.5 External Video display 39

5 REFERENCES 40

5.1 References to Mediate deliverables 40

5.2 References to external sources 40

5.3 References to other Itea2 projects 40

1 Introduction

1.1 Purpose

The objective of the Mediate project is to increase productivity and effectiveness in healthcare and reduce patient risk and discomfort by supporting healthcare professionals in the transition from invasive, open surgery to minimally invasive, **image guided intervention and treatment** (IGIT). The purpose of this document is to describe the high level architecture of this IGIT system.

The approach taken for the document is mostly compliant with ISO/IEC 42010 standard "System and software engineering – recommended practice for architectural description of software-intensive systems". This standard has been taken as a guide for building the document, although its structure has been adapted to Mediate own specific requirements

1.2 Scope

Due its technical nature, it will focus mainly on the definition of Architecture and its relationships with the clinical part. It is outside the scope of this document to go into details of Mediate's clinical Use Cases.

2 Executive summary

The objective of the Mediate project is to increase productivity and effectiveness in healthcare and reduce patient risk and discomfort by supporting healthcare professionals in the transition from invasive, open surgery to minimally invasive, **image guided intervention and treatment** (IGIT). This document describes a global architecture for the Mediate system and discusses the components that form it and their relationships.

The major aspects of the global architecture have been studied in the finished Mediate-Wp2 deliverables:

- [D211] has studied the security and privacy aspects of the IGIT architecture.
- [D212] has extracted the requirement on the IGIT architecture from the clinical use cases, as worked out in Mediate-Wp1
- [D221] has studied the available standards for the IGIT architecture.

This document captures the knowledge from the studies mentioned above and the combined expertise of the Mediate-Wp2 partners into the global architecture specification of the Mediate IGIT architecture.

Next to the Mediate internal knowledge (Mediate deliverables and expertise of the partners), and external sources, also a connection has been made towards other projects:

- Itea2 projects.
  The Itea2 projects *HighProfile, Hipip,* and *Care4Me* have a link to the IGIT architecture.
- CENIT project
  Also a connection with *cvREMOD*, a Spanish CENIT project, has been established by means of the Workflow Controller

In this document references are made to these Itea2 projects were applicable.

The figure below gives an overview of the IGIT architecture, as described in section 4.1.1. This figure shows the expertise areas in the red circles.

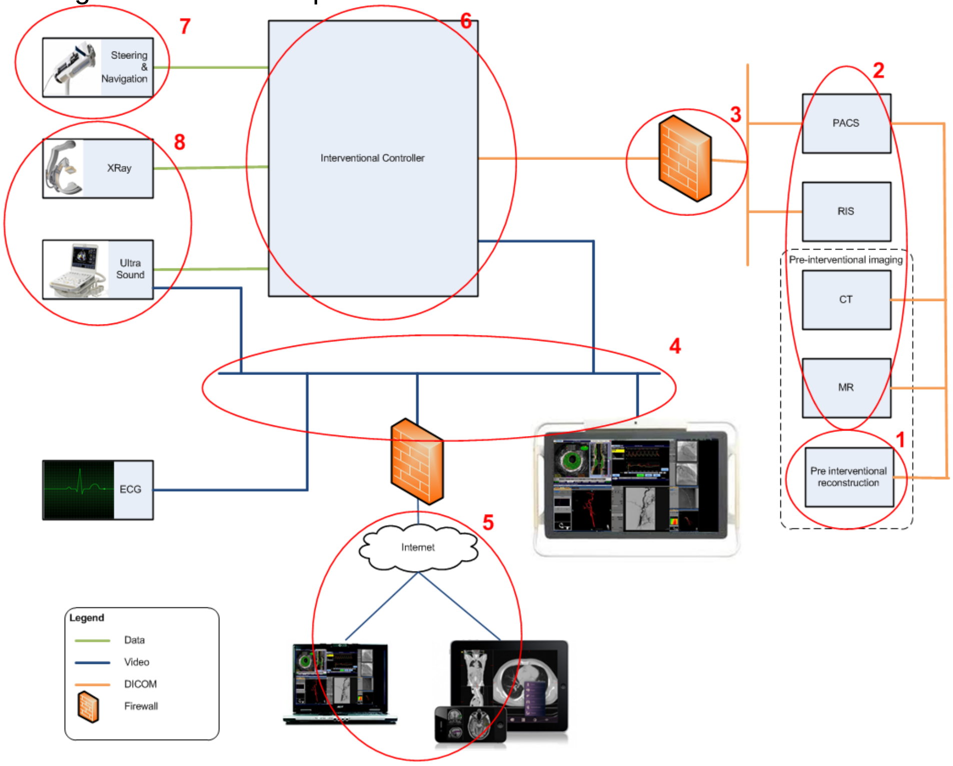


Figure 1 Expertise mapped on the overall IGIT architecture

The expertise of the architecture is divided as follows (based on the numbers of the red circles):

1. The pre-interventional reconstruction is part of the expertise area of IAC3. Also the Itea2 Care4Me project has done investigations in this area.
2. The communication between the IGIT and the external modalities and systems is part of the expertise area of both Atos and Philips. Where Philips provided the information for [D212], Atos provided the information for this deliverable.
3. The firewall is part of Atos expertise and security aspect have already been specified in [D211]
4. The video network is part of the expertise of Barco.
5. The remote video network is part of the expertise of Technolution.
6. The Interventional Controller is part of the expertise of Atos, and can be explained through the example of Gimias tool, coming from the cvREMOD project.
   The processing done as part of the interventional engine has been subject of study in the Itea2 HighProfile project.
7. The steering and navigation is part of the expertise of Haption and Nucletron.
8. The sensors is part of the expertise of Philips.

3 Stakeholders and concerns

This section identifies the stakeholders and their concerns.

3.1 Clinical users

This section describes the concerns of the stakeholder *clinical user.* The clinical uses are the daily users of the system. The concerns of the clinical users were also studied in Mediate-WP2 deliverable *D.2.1.2 Technical report focuses on the use cases*.

3.2 Insurance companies

This section describes the concerns of the stakeholder *Insurance companies*. Although most of the concerns can be shared with Clinical Users, historically the distinction between these two is made explicitly.

3.3 Researchers

This section describes the concerns of the stakeholder *researchers*. A researcher is a kind of user that access to the data mainly for processing purposes. Therefore the access needed is related with the storage of raw data, for example, coming from monitoring and images sources.

3.4 Developers

This section describes the concerns of the stakeholder *Developers.* The developers are people who develop and release new components that are part of the Mediate system. Components include sensors, actuators, controllers, clinical applications etc.

3.5 Manufacturers

This section describes the concerns of the stakeholder *Manufacturers*. This stakeholders are the vendors that build hardware and software resources. Hence, their interest is related with the performance of the systems, bug reporting, versioning/upgrade or final user suggestions, for example.

3.6 IT administrators

This section describes the concerns of the stakeholder *IT administrators*, who are the IT experts in charge of the technical management of the systems. Their responsibilities go from users administration (create, modify, delete) to fixing bugs or reporting, covering also many technical IT-related issues, as reboot of servers, interconnection with other systems surveillance, logs supervision, backup management, …

3.7 Helpdesk

This section describes the concerns of the stakeholder *Helpdesk*. As new devices evolve into bigger and more complex systems, the concept of Helpdesk is being as necessary, especially for system with many users. This role takes care of the “first level” of support to users, handling common issues like plugging/installing of new devices like printers, virus scanning, network shutdowns, ….

4 Architectural views

4.1 Structural view

This section describes the structural view on the architecture. This structural view consists of the architecture diagram, and components and interfaces.

4.1.1 Overall architecture

The figure below shows an overview of the overall architecture of the IGIT system. The different components and their connections are detailed below.

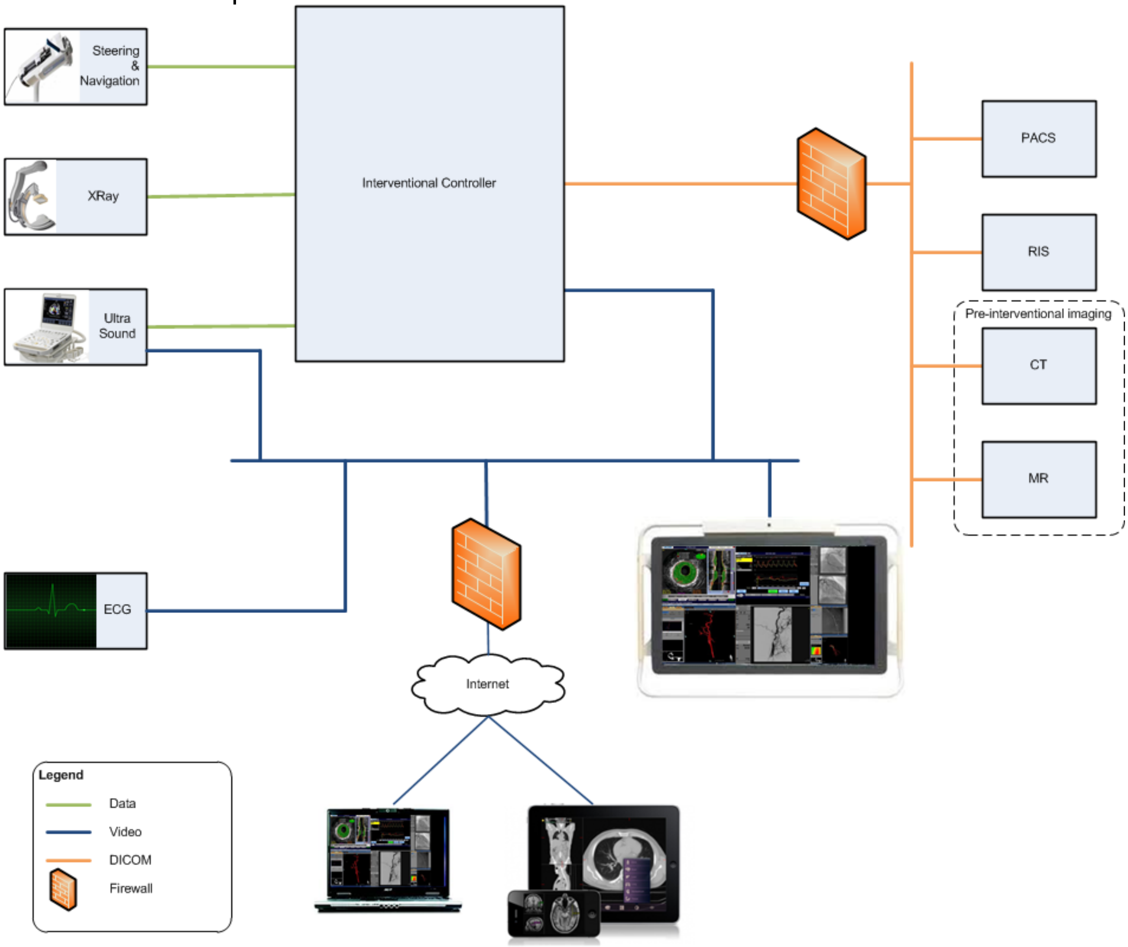


Figure 2 Overall IGIT architecture

Figure 2 shows different high level aspects of the IGIT architecture. The following aspects are shown:

- Interventional controller, shown on the top-middle of Figure 2
- Sensors, shown on the left side of Figure 2. Only a limited number of sensors are shown to simplify the figure. Other sensors will be placed similarly in the architecture.
- External systems, shown on the right side of Figure 2. The external systems contain both pre-interventional imaging modalities (e.g. CT and MR) and, PACS and RIS systems.
- Video infrastructure, shown on the middle of Figure 2.

These aspects are detailed below.

Interventional controller
The heart of the IGIT system is the interventional controller. The responsibilities of this interventional controller:

- Communication with sensors (drawn on the left side of Figure 2) for control (workflow) and retrieval of data (e.g. pixel and meta data)
- Communication with the external systems (drawn on the right side of Figure 2). These external systems include pre-interventional imaging modalities as well as PACS and RIS systems. The communication protocol for these external systems is DICOM, consisting of pixel and meta data.
- Outputs video to the video infrastructure (drawn in the middle of Figure 2, indicated in blue lines)

The internal architecture of the interventional controller is described in section 4.1.2.

Sensors
The sensors contain the inter-operative imaging modalities, actuators, controllers and other sensors like ECG. These sensors can be connected to interventional controller, or directly to the video infrastructure (as shown for ECG), or to both (as shown for ultrasound). More details on the sensors can be found in section 4.1.3

External systems
The external systems are connected via a firewall, as the external systems are seen as “the outside world”. The external systems consist of:

- Pre-interventional imaging modalities, for example MR and CT
- PACS
- RIS
- Pre-interventional reconstruction
- Workstations

More details on the external systems can be found in section 4.1.4

Video infrastructure
The video infrastructure handles the video from various sources to the display. These displays can be in and outside the interventional environments. More details on the video infrastructure can be found in section 4.1.5

4.1.2 Interventional Controller Architecture

The responsibilities of this interventional controller:

- Communication with sensors (drawn on the left side of Figure 2) for control (workflow) and retrieval of data (e.g. pixel and meta data)
- Communication with the external systems (drawn on the right side of Figure 2). These external systems include pre-interventional imaging modalities as well as PACS and RIS systems.
- Outputs video to the video infrastructure (drawn in the middle of Figure 2, indicated in blue lines)

The way to process the information received by the sensors (e.g. pixel and meta data) has been subject of investigation in the Itea2 project High-Profile, see [HIGHPROFILE]. Therefore, this document does not elaborate on this subject.

The Interventional Controller is responsible for the state of the interventional system. It makes sure that all connected systems are set in correct state, and that the appropriate information is shown to the users of the system.

Important aspects of the overall system state are:

- The current patient that is being treated on the interventional system and sharing this with all applications
- Synchronization of the several system parts, when synchronization is needed. An example is when a specific application can only be started after another application has finished processing and has provided information.
- Management of scarce and shared resources, such as sensors. An example is when multiple applications need a sensor with specific sensor settings. The Interventional Controller will provide APIs to allocate and release specific resources.

Some other aspects of the Interventional Controller are

- Giving access to the appropriate data for that patient
- Display of the appropriate information to the users, depending on the system state
- Make sure that all generated data is properly saved and reported

The mapping of the interventional controller and its applications toward hardware components is not part of this architecture document as it has been subject for various Itea2 projects; see [HIGHPROFILE], [HIPIP] and [CARE4ME].

Note that the Interventional Controller is not responsible to implement specific disease interventional applications. The Interventional Controller is the **integrator** of multiple applications.

The internal design of the applications is out of scope of this architecture specification, as these applications are disease specific. The processing done by these applications has been subject of investigation in the Itea2 project Hipip, see [HIPIP].

4.1.2.1 Workflow Controller

In a generic form, a *workflow* can be understood as a set of ordered steps that lead to the completion of a more general task. Taken this concept to the Medical domain, a *Medical Workflow* is a sequence of operations within Medical Context that have a goal of the Healthcare domain. This obviously depends on the level of abstraction that is being taken into account: from the simplest chain of events, for example, “*acquisition of an MRI image*” to the more complex “*prevention, diagnosis, treatment and follow-up of Atrial Fibrillation on a patient*”. The clinical praxis is guided by **Clinical Guidelines**, documents accepted by Medical communities, which formalized some aspects of these Medical Workflows. There is a growing interest on the Healthcare domain in implementing IT tools able to cope with the different peculiarities of these Clinical Guidelines easing the Physician work. Workflow Controller are some of those tools. Within the project, a *workflow controller* stands for an application (or set of applications) that reflects a Medical Workflow, so enabling a point-to-point mapping between what the clinician needs during the medical praxis and what he can see on the application that support his work. It is expected then that the tool changes its appearance and behavior on each step, thus adapting to what the Physician needs more on each step.

Although this correlation between the steps of the workflow and what the clinician needs on each of them may not be always direct, what is needed most of the times is at least some “connection points” between the medical world and the application.
In fact, there are many times when the IT Analysis of a medical application is done taken as basis the Medical Workflow in which the physician is involved during his daily work. On those occasions, the resultant application reflects exactly the steps of the Medical Workflow, thus improving the transition to the new application. The design of a whole architecture is then influenced by this way, since one of the objectives of it (and, specifically of Mediate’s) is to ease the management of IT tools on both intervention and pre-intervention rooms.
One quite-appropriate way to interact with the clinician for those workflows-based applications is by means of Wizard-shaped user interfaces. A structure of these is shown on the next figure:

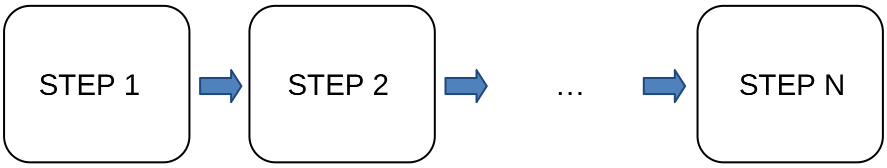


Figure 3: Steps sequence of a workflow

On these Wizard-shaped UI, the Workflow Controller shows a screen to the physician, where each of the steps is shown one by one, as the Physician is developing tasks. A strong emphasis is made on the fact that most of the steps must be auto-completed, so proposing values for the variables the Physician must filled. For example, if the Physician wants to perform a reconstruction of some vessels from MRI images and the algorithm (i.e., step X) that does that needs some inputs, the Workflow Controller will fill the parameters of the algorithm with the most common values. This way, the step can be done pseudo-automatically and the Physician has the chance of changing values if he wants.
More Mediate-specific examples of Medical Workflows are shown on D121:

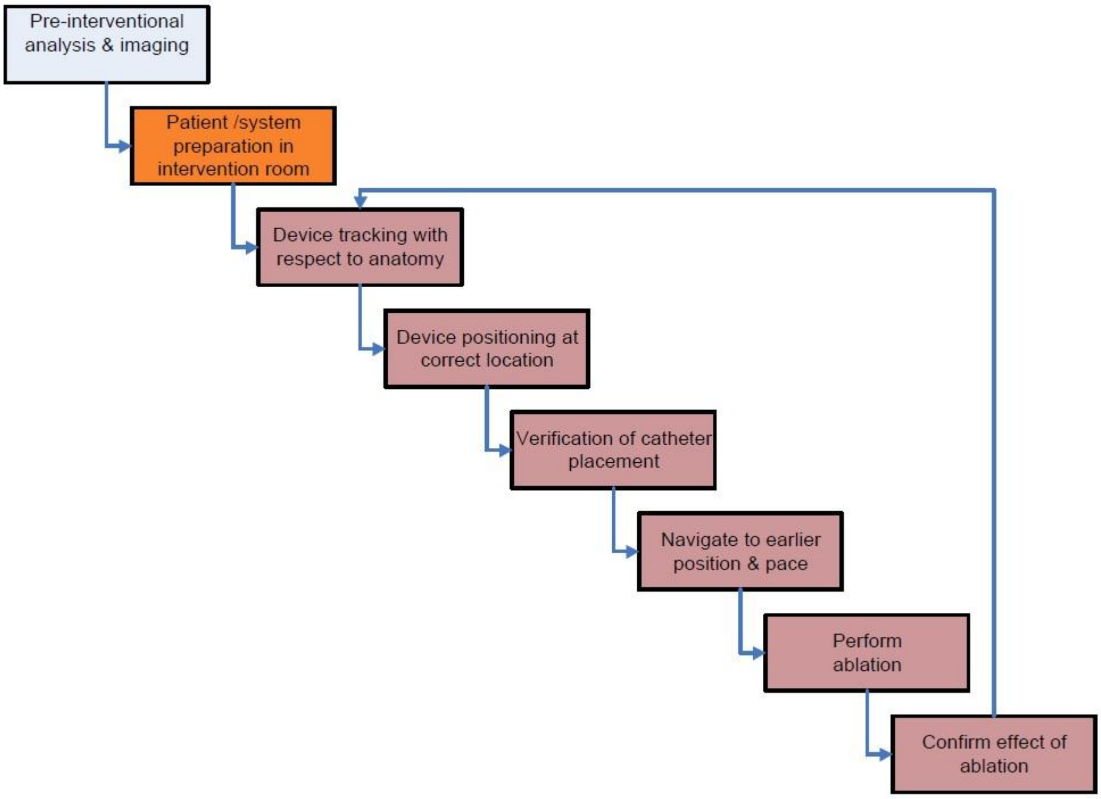

Pre-interventional analysis & imaging
Patient /system preparation in intervention room
Device tracking with respect to anatomy
Device positioning at correct location
Verification of catheter placement
Navigate to earlier position & pace
Perform ablation
Confirm effect of ablation

Figure 4: Example of Mediate Workflow

These medical workflows have sometimes different roles involved on them: physicians, nurses and referrals, for example, can interact within the same workflow with no appreciable distinction.

From the technical point of view, a Workflow controller stands for the application in charge of the subordinate applications that interact with the low-level systems. On the last years, new emergent technologies are being introduced into the Workflow controllers to ease the process of transferring data from a step to the following. One example of these is the Cloud technologies: by means of the Cloud, a common repository of data can be easily accessed from all the applications with a minimal control needed from the controller part.

Not only data, but also services can be “clouded”, resulting in a skeleton architecture for the workflow controller:

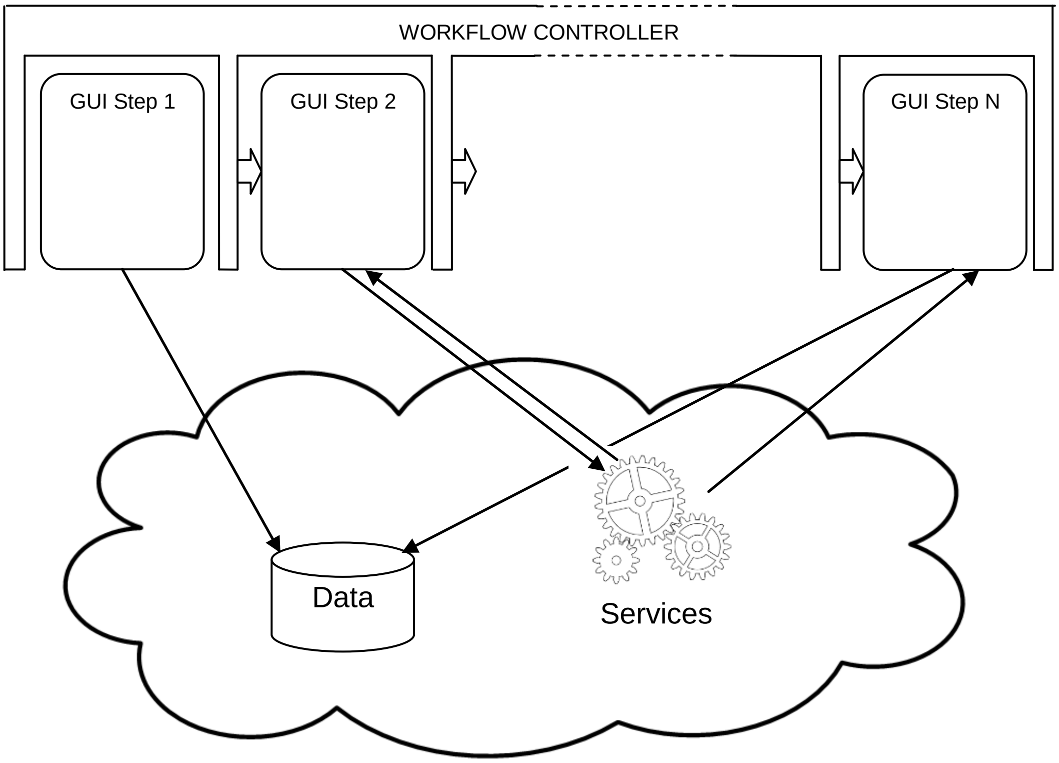


Figure 5: Skeleton architecture for the Workflow Controller

In this abstract architecture, the workflow controller only deals with sync issues between the steps of the workflow or wizards, while each one the steps, which can be a different application or be integrated into de workflow itself, deals with the User Interface management. The resulting architecture can be viewed as an evolution of the MVC (Model-View-Controller) paradigm, where the Model role is played by the Cloud, the View by each one of the steps and the Controller by the Workflow Controller.

Internally, the Workflow Controller can be viewed as a core provider of connectors where ad-hoc plug-in can be specifically developed and plugged into the system when needed. This turns out into an easy-to-maintain (and scale) tool: with a common API for the interconnection, as new functionalities are being needed new plug-in can be written in a variety of languages that come up with algorithms to solve specific problems. A good example of this kind of architecture focused on the medical domain is Gimias.

Gimias is “*a workflow-oriented environment focused on biomedical image computing and simulation*” [GIMIAS], whose core is written in C++: it provides a way for building medical workflows by dividing them into steps. Each one of the *GUI steps* is designed and compiled within the Gimias library, thus enabling a common appearance. Inside these steps, the processing algorithms can be added in the form of plug-in, being the unique requirement that they can be called from a C++ application: the algorithms within can be written in any programming language. Gimias provides a unified way to exchange data among the steps called the DataTree.

A real example of pre-interventional workflow focused on aneurism characterization developed in Gimias contains several steps:

1. Data selection. A GUI with dedicated plug-in to connect to PACs, HIS and other data storage systems.
2. Data analysis. The data selected on previous step is then analyzed by the clinician. He can run several algorithms (plug-in) taking 2D images to segmentation and characterize the aneurysm and vessels. The result will be a 3D reconstruction of these vessels and the aneurysm. The GUI allows rotate the view, change contrast/brightness and other generic functionalities over the image.
3. Aneurism management. Taking the previous step images, others algorithms can calculate physical variables of the aneurysm (neck diameter, volume, etc)
4. Stents/coils simulation. The physician, according to what he learned of the aneurysm, can choose a stent or coil from a library and place it on the aneurysm.
5. Hemodynamic simulation. The physician can run a simulation of the stress over the vessels wall and so predict how the aneurysm will react to the therapy, foreseeing if it could eventually break.
6. Report generation. Finally, a report with the images, measures and recommendations of the physician can be prepared and printed.

As it is seen, the framework provides, next to the common data interface, a unified GUI for all the steps and the flow of data among them. Specific tasks are performed within each step by specific algorithms, plugged into the framework by means of dedicated plug-in.

4.1.3 Sensors architecture

The Sensor architecture is designed to add a layer of abstraction between the Infrastructure API and the API of the manufacturer-provided driver. This design promotes loose coupling:

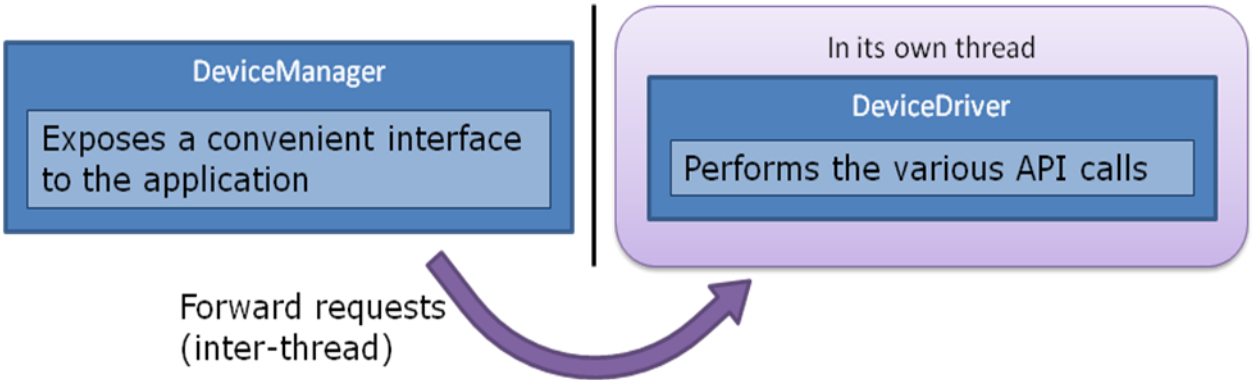
- An application should never perform API calls directly but it should use the associated Sensor architecture instead.
  The rational is modularity for easing code maintenance, code testing and code reusability.

Loose coupling reduces the impact of modifications (when adding, removing and/or modifying functionalities) thus ensuring a high level of modularity. It also allows for a common interface over different devices.
The main specifications of the Sensor architecture are:

- Isolate the API calls in their own thread to avoid freezing the application when accessing the device;
- Expose a convenient interface (both methods and data) to encapsulate the various API calls.

These specifications result in a twofold design:

- A *DeviceDriver* part that performs the various API calls and
- A *DeviceManager* part that forwards the requests from the application to the DeviceDriver.


DeviceManager
Exposes a convenient interface to the application
In its own thread
DeviceDriver
Performs the various API calls
Forward requests (inter-thread)

These proposed rules provide a development framework to use a new driver but do not address design choices such as:

- What does trigger the retrieval of data?
- Where to store the data? How to pass the data back to the application?

These choices depend on the targeted devices.

4.1.3.1 Steering and Navigation

Steering and Navigation require data describing the patient and the surgeon tools. For example, the STL files providing the 3D model of the bones inside the intervention area is needed. Specific XML files including the link between anatomic points and 3D model must be stored inside data base and load by at the beginning of the intervention in the interventional controller. The 3D model of the tools used by the surgeon is also obtained for the interventional controller.

The steering and navigation framework helps a user to implement the association of objects with 3D or 6D references that will manage the position and orientation of the object, meaning 6D the transformation between 2 frames: It defines the position of a body in space (3 translations and 3 rotations)

Then, the steering and navigation devices localize the references of all objects.

The framework implements data storage and use the Device Manager. Data volume is relatively low but rates are very high as soon as objects are in movement or data is coupled with robotics devices. Concerning latency, this must be minimum: for haptic applications it should not be greater than 20 ms.

The relative position of the tools regarding to the intervention area is then updated and coupled to 3D models:

1) To be managed into a physical simulation
2) To be displayed by the application on monitor through video network.

Both the simulation component and the GUI are connected to the haptic device in order to guide the surgeon during the intervention.

4.1.3.2 X-Ray

The X-Ray interface helps a user to implement a generic workflow providing abstraction the following functions:

- Configuration of the device including calibration procedure and acquisition parameters
- Image acquisition
- Image filtering & reconstruction

- Image storage for real-time use by DeviceManager (see section 4.1.3)
- Image conversion to 2D data containers compatible with the infrastructure API

4.1.3.3 Ultrasound

The Ultrasound interface helps user to implement a generic workflow providing abstraction the following functions:

- Configuration of the device including calibration and acquisition parameters
- Image acquisition
- Image filtering & reconstruction to produce a 3D slicer for ultrasound volume based and a 2D view for ultrasound images
- Image storage for real-time use by DeviceManager
- Image conversion to 2D/3D data containers compatible with the infrastructure API

4.1.4 External system architecture

As it is described on [D221], the DICOM standard was initially used for handling, storing, printing and transmitting medical images over networks. Nowadays, the Standard is still evolving towards transmission of non-image data, like for example DICOM ECG ([FIN2011]).

The communication towards the Interventional Controller is made through a Firewall, to ensure that only safe traffic can enter the core. Since the networks at both sides of the firewall are inside the hospital, the traffic between them is known. The Firewall implemented can be not only a packet filter but also an enhanced application traffic filter, if needed. Additional security measures, like a DMZ or a bastion node, can be assessed depending on the level of security needed.

Concerning the communication protocol itself, this must be based on DICOM protocol: to form a PACS network, two kinds of entities are needed: the **PACS Server**, which will be located outside the firewall and the Interventional Controller, and the **PACS Clients**. These entities (named **DICOM Nodes**) are typically identified by their network address (IP address), a communication port (TCP/IP port) and a name.

The PACS Server stores the DICOM images and has a service to give quick response to retrieval/store queries coming from the clients:

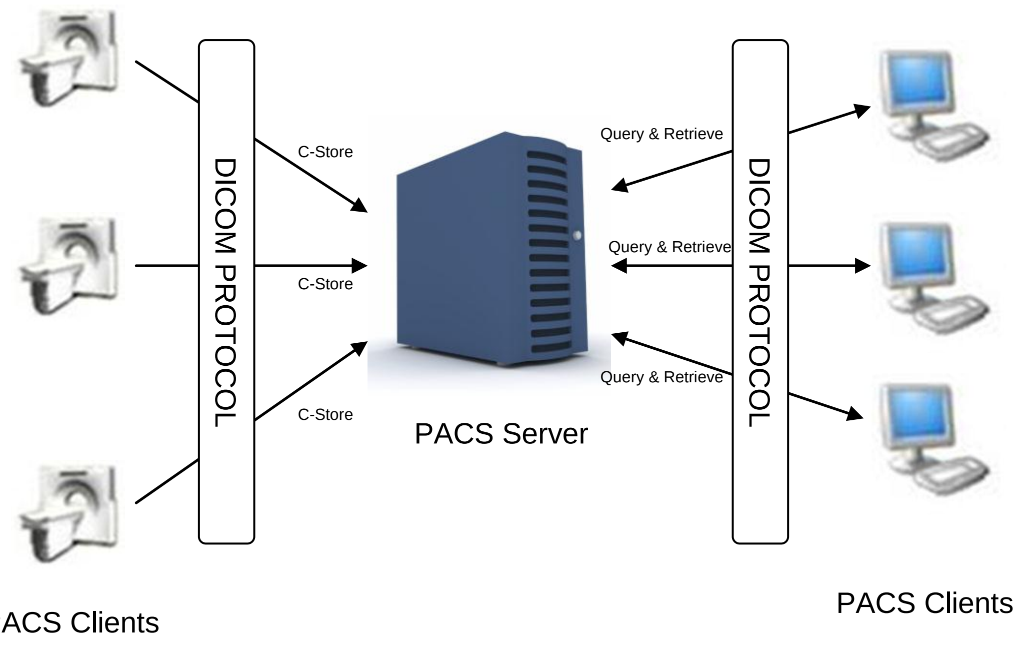


Figure 6: PACS server and clients

As this flow of data must pass through the Interventional Controller, two approaches are feasible:

1. To add IC itself as a DICOM node, thus making it part of PACS network.
2. To add each client on the inner network as DICOM node, letting the IC as a simple proxy of data flow (between PACS server and Clients) in this case.

Concerning DICOM protocol itself, it contains several commands that enable usual actions on query/retrieval environments. They are divided on Composite (C-XXX) and Normalized (N-XXX) operations:

- C-Store. Known also as “DICOM push”.
- C-Find. A query.
- C-Move. It creates and copies on the same operation.
- C-Get. Similar to C-Move.
- C-Echo. Similar to the ping command on other networks.
- N-Create. It creates a dataset.
- N-Get. It requests a single dataset.
- N-Set. It allows updating a single dataset.
- N-Action. For general use, for example, check about the status of an image.
- N-Delete. It deletes a particular object.
- N-Event-Report. It reports to the client information about an event, for example, feedback about a previous command.

4.1.4.1 Pre-Interventional reconstruction

The pre-interventional reconstruction components has the objective of reconstructing anatomical and physiological information taking as a basis the acquired images (CT, MRI) in each clinical case. This extra layer of information is not obvious from the original images, and it can only be qualitatively estimated by the experienced clinician, while the pre-interventional reconstruction provides quantitative details and therefore improves the planning of interventions.

The anatomical information is obtained through image processing techniques, while the physiological information is mainly obtained through numerical simulation. The simulation uses as inputs the anatomical models obtained through image processing, population-validated physiological models and specific physiological parameters from the patient. The results of that simulation are physiological parameters such as pressure, fluxes, temperature, etc.

These simulations are typically performed on advanced computer systems. The accuracy of a simulation is proportional to the time it takes, which calls for high performance computing. A single computer does not have enough processing power. Therefore simulations are frequently performed on cluster systems. In the general case these clusters are remotely accessed, and they are known as grid systems.

The figure below shows an overview of the pre-interventional reconstruction component architecture.

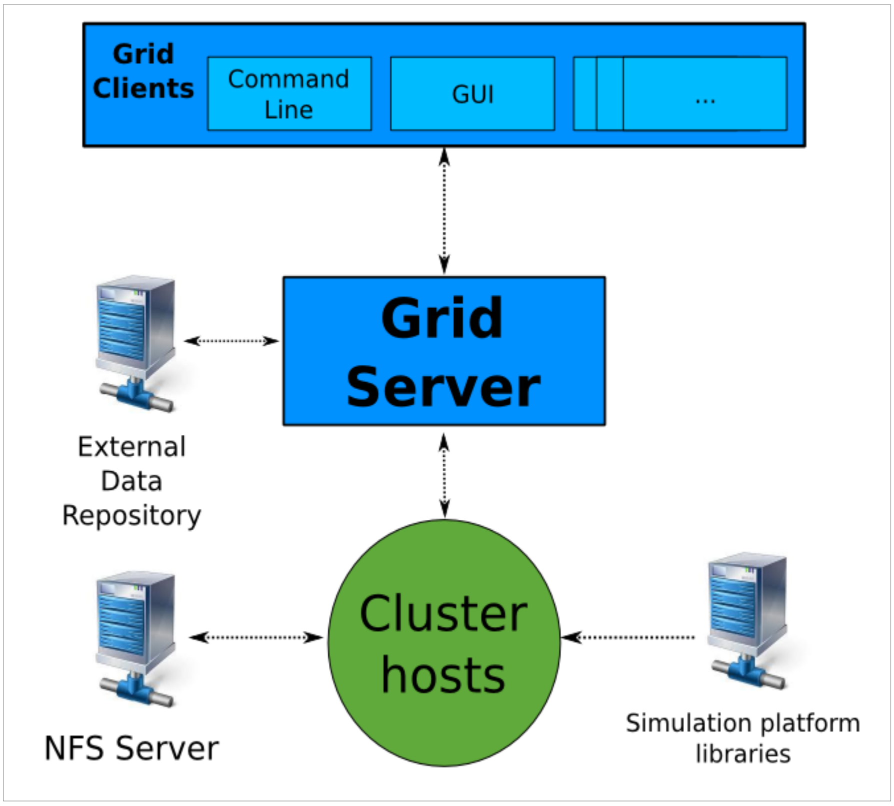


Figure 7: Pre-interventional architecture

A RMS (Resource Management System) software is required to manage the cluster and run parallel tasks (by using MPI, the de facto standard). In order to access either a cluster in a remote way or multiple clusters at a time, another layer is needed: grid middleware. A grid system is composed of a client layer and a server layer. The client layer is used to communicate user applications with the cluster hosts, and provides an abstraction to the machine configuration. The server layer connects with the RMS, transforming the client tasks into RMS tasks and returning the tasks results to the clients. Communication with and within this component must be authenticated using digital certificates. The grid system should have the ability to access external repositories (such as a PACS) to retrieve input data and send results, and this is granted through authenticated communication (and delegation for data staging efficiency).

The Itea2 project Care4Me, see [CARE4ME], investigated how to map the reconstruction components to hardware components, e.g. advanced computer systems. Therefore, this document does not elaborate on this mapping towards hardware.

4.1.4.1.1 Pre-Interventional reconstruction interface

The output from the pre-interventional system simulations are HDF5 (Hierarchical Data Format) files. This kind of files may be transformed into DICOM files and other formats, and pushed either into the PACS or directly to the Interventional Controller. This can be achieved with scientific software such as IDL (see [IDL]), that provides an API for reading and writing both formats and perform modifications at the pixel level.

4.1.5 Video infrastructure architecture

4.1.5.1 Local Video Distribution

The video infrastructure is responsible for distributing the video from the inputs to the various endpoints, inside or outside the intervention room. In order to have a flexible distribution environment that can easily interface with various parts in the hospital or even beyond, an IP-based architecture is the most optimal. Depending on the location of the video (inside or outside the intervention room), uncompressed or compressed video is used.

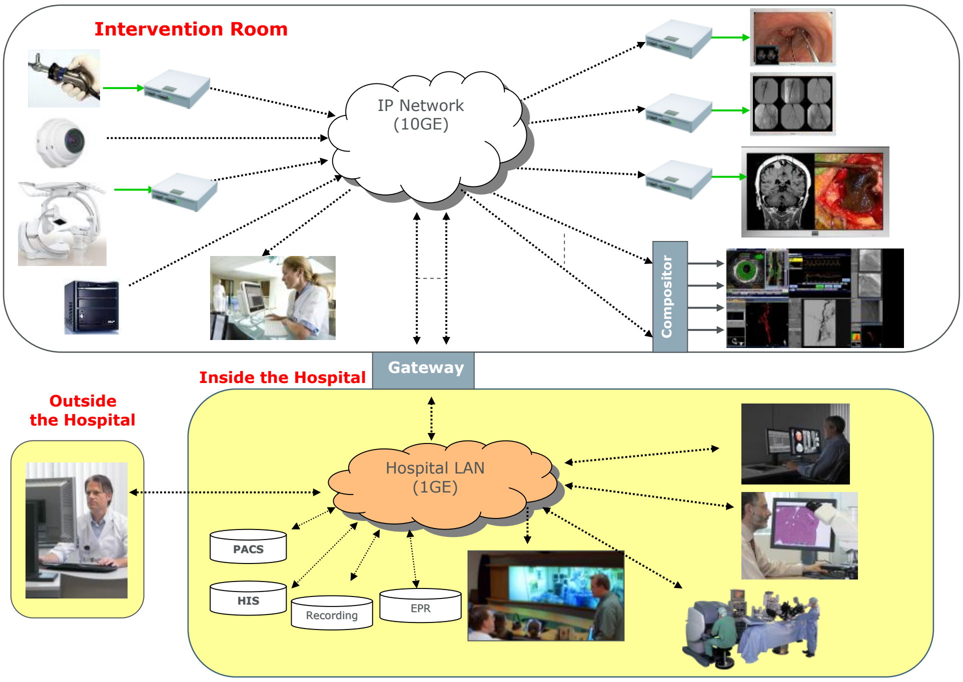


Figure 8. Video distribution network

Inside the intervention room, the critical requirements are:

- ultra-low latency
- pixel-perfect image quality
- ease of use
- flexibility in connection of sources
- compatibility with any type and resolution of video input signal

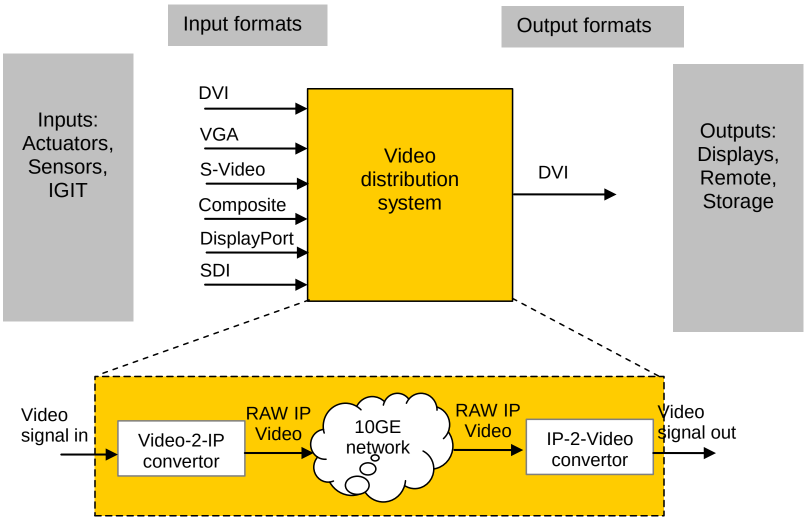


Figure 9. Video distribution inside the OR

All these requirements can be met via the use of uncompressed (RAW) video over an IP network. As the video resolution of the sources can go up to full HD, the IP network must be able to process more than 3 Gbps per stream. As a consequence, a 10Gbps network is required.

At the endpoints of the network, the video is converted between IP and the video signal out in order to be able to show the image content on the displays or store it on a local storage device.

For large resolution displays like the 56” 8MP display in the intervention room, the video from the various sources needs to be integrated in a video composition before being sent to the display.

4.1.5.2 Remote Video Distribution

In section4.1.5.1, the distribution of video streams to local displays has been discussed. Based on a dedicated 10GBE Ethernet infrastructure, video streams can be distributed with guaranteed low latency and high quality. Although this approach is very well suited for local video distribution, it is not very well suited for all use cases. Especially in use cases in which video will be distributed to remote locations, over networks with a lower (and less stable) bandwidth, the approach discussed in the previous section will not be a suitable solution. For these cases, a different approach is needed.

For distribution of video streams to remote locations, an approach is needed that is able to deal with lower bandwidths (which may be fluctuating). It has to deal with LAN infrastructures of ≤100Mbps and internet connections.

In addition to bandwidth issues, security aspects have to be considered as well. Patient health information cannot be distributed without security and privacy controls.

Requirements

For this application, the requirements are different from the local video distribution application:

- Latency can be significantly longer, since hand-eye coordination is not relevant for this case. Unless voice/video communication is being used between the local user and the remote user, a delay of several seconds is not problematic.

  To support voice/video communication between the local user and the remote user, a latency of at most 1 second is preferred.

  Preferably, the user should be able to configure the maximum latency.
  This parameter can be taken into account in the compression settings.
  A longer latency will result in better compression results.

- For the transfer of the image stream to remote locations, more compression will be needed than on the local video network. On the local video network (10GBE) compression has to be lossless to maximize image quality on the local displays. For remote video distribution (on ≤1GBE LAN or the Internet) , lossless compression will not reduce the bitrate sufficiently. Lossy compression will be used.

  During the project, the image quality needs to be evaluated to determine which compression settings result in an adequate balance between bitrate, latency and image quality.

- Preferably, the remote user should not be forced to install special software on the remote client (PC, tablet, phone). Therefore, wide-spread standards should be used. For video compression, we will focus on H.264 and VP8. It should be noted that not all Internet browsers support all video formats. This is a point of attention: selecting the right formats for the selected browsers.

  At this moment, we consider it acceptable if the user has to install wide-spread plugins for his webbrowser like Adobe Flashplayer or VLC.

  The video capabilities of HTML5 will be explored to avoid hassle for the remote user: when he/she connects to the available video streams, they should play automatically.

- Since video streams will be sent via the Internet, attention has to be given to privacy and security, especially since health information will be distributed.

  Encryption will probably be needed.
  Restricted access to the available video streams should also be secured by means of passwords, certificates, etc.

The top-level architecture for the remote video distribution approach has been sketched in the figure below.

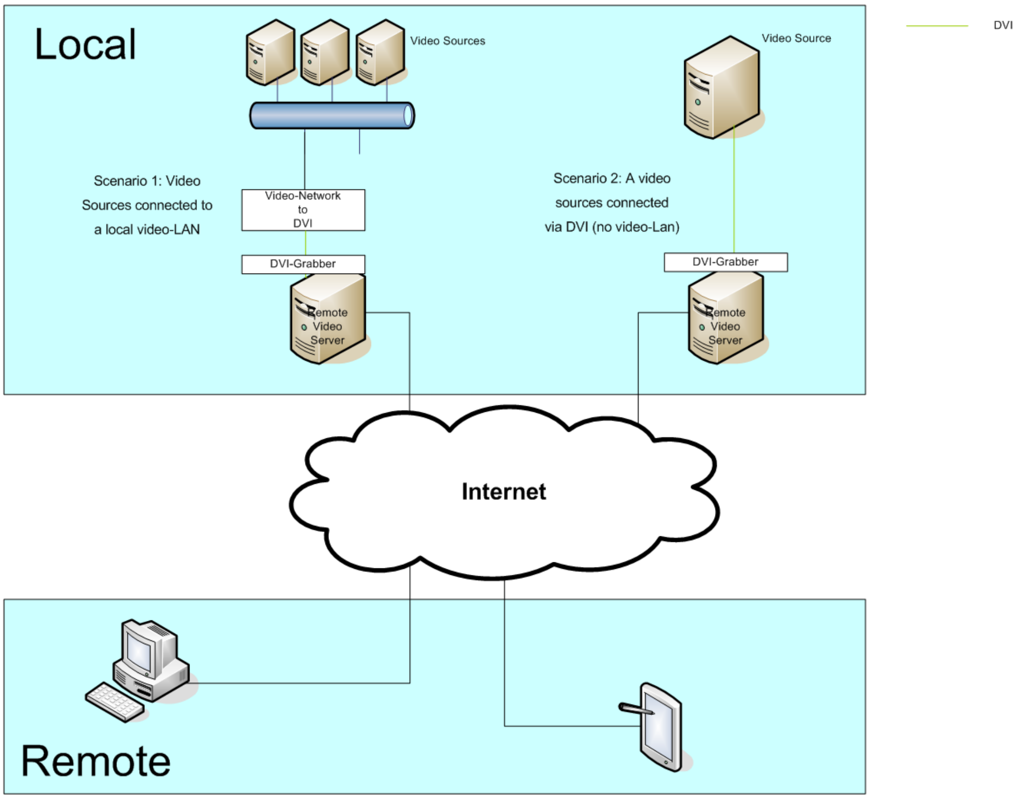


Figure 10. Remote video distribution approach

In this figure, two scenarios have been sketched. In the first scenario video is being distributed over a local video-LAN as discussed in the previous section. Connected to this LAN is a box that converts the video stream back to DVI. This DVI stream can be taken into the “remote video server “by means of a DVIGrabber.
In the ideal situation, the video-lan to DVI conversion box can be omitted: the remote video server can be connected to the local video LAN by means of a 10GBE NIC.

In the remote video server, the video is converted into a video stream format that can be used over the Internet. For this purpose various standards (and standard solutions) are available. During the project the choice will be made for a convenient standard for distribution of video streams over the Internet. Based on current insights, we assume that more than one video encoding standard will be needed to support all available web browsers which can be used to view the video stream remotely. For video compression, H.264 and VP8 are likely candidates.

The remote user can use a standard browser on a PC, a smart phone or a tablet to connect to the remote video server via the Internet. In the browsers a plug-in might be needed, or a browser can be used which supports HTML5.

As an alternative to scenario 1 (based on a remote video server connected to a video LAN), the remote video server can also be equipped with multiple DVIGrabber, which can be connected to video sources directly (scenario 2 in the figure).

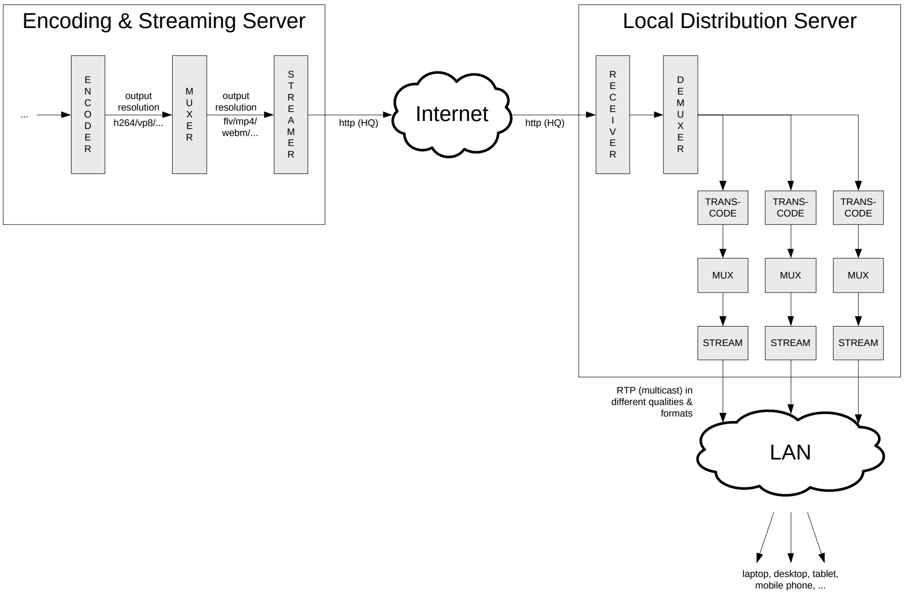
Two types of use cases are foreseen for remote video distribution:

1. A group of users connect to the server for viewing a grabbed video stream. This is similar to a live webcast. In this case, sending individual streams to individual users will require a large bandwidth at the server side.
   When all users are on one location (e.g., in a conference room) it will also require a large bandwidth at the client side.

   For these cases, a better approach is to create a small number of live streams (e.g., HD and SD) and to multicast these streams to the connected users. A user can select one of the streams based on the available bandwidth.

   Since multi-cast cannot be used over the Internet, this can only be used for distribution over a LAN. The local video distribution discussed in the previous section will use a dedicated 10GBE network. The remote video distribution with multicast can be used on a much slower LAN.

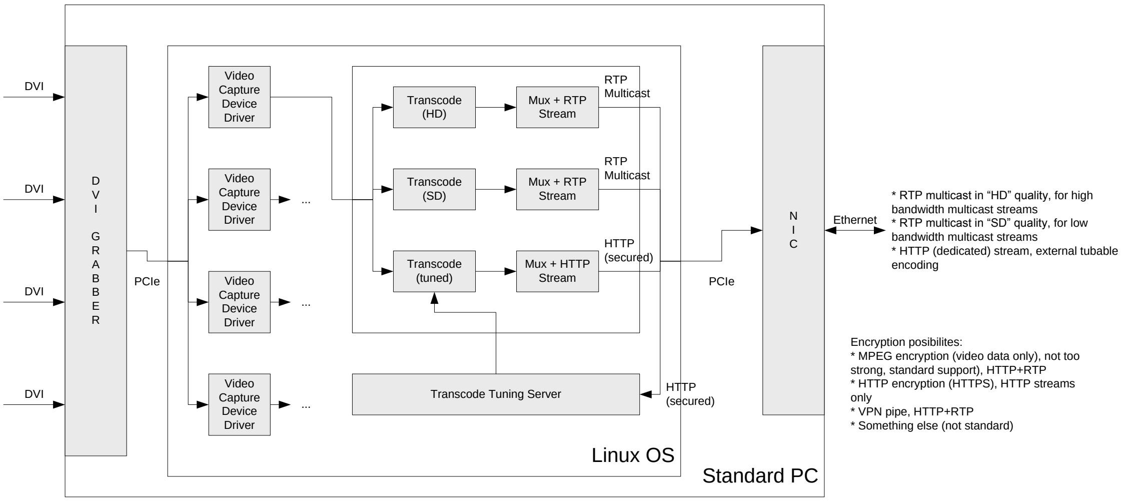
   When multicast is to be used for a group of remote users that are not on the LAN, a connection needs to be set up between the local LAN and the remote LAN. A server can use unicast to send the image stream over the Internet to a server on the LAN of the remote users. On the remote LAN, multicast can be used to distribute the stream to the users on the remote LAN.



2. A single user connects to the server, the user can interact with the remote video server. In this case, the user can do more than just view the live stream. He might want to zoom/pan, pause, rewind, etc. For this case, functionality would be needed at the client (in the browser) to interact with the server. This can be done via a simple web application. On the server side, selections made by the user are converted into parameters for the compression engine.

   In this case, a more advanced method can be used to match the quality of the image stream with the available bandwidth: based on the throughput of the connection, the compression parameters can be dynamically set to adapt to fluctuations in the bandwidth and to settings changed by the client.

The diagram below sketches the software processes involved in a set-up that supports both use cases.



* RTP multicast in “HD” quality, for high bandwidth multicast streams
* RTP multicast in “SD” quality, for low bandwidth multicast streams
* HTTP (dedicated) stream, external tubable encoding

Encryption posibilites:
* MPEG encryption (video data only), not too strong, standard support), HTTP+RTP
* HTTP encryption (HTTPS), HTTP streams only
* VPN pipe, HTTP+RTP
* Something else (not standard)

Figure 11. Involved software processes in remote video distribution

For security and privacy, secure internet connections will be used with encryption and password based login.

4.2 Behavioral view

This section describes the behavioral view on the architecture. This behavioral view consists of the component collaboration explaining how the system works.
In order to make this section more concrete, a use case has been selected and mapped to the architecture. This approach enables the reader to understand the architecture and it's components by an actual clinical use case. Moreover, the mapping of an actual clinical use case as serves as a validation of the architecture. The selected clinical use case is the minimal invasive Brachy therapy.

4.2.1 Minimal invasive Brachy therapy

4.2.1.1 Brief Description

This use case describes the actions involved to perform minimal invasive Brachytherapy. Brachytherapy is a radio therapy, by which cancer is killed by radioactive radiation. The goal in Brachytherapy is placement of radioactive sources as close as possible at the tumor and spare the organs at risk in the patient.
To achieve this goal of giving as much as possible radiation on the tumor and as less as possible at the healthy organs, a radio therapy planning is made based on imaging and software calculations to calculate the dose distributions in the patient.
The next step in the process is to place catheters, needles or applicators so that the radioactive sources can go into the patient according the therapy planning.
To make the implant with catheters or applicators in the patient is a difficult task and has to be monitored and controlled. Mostly a radio therapy planning must be updated with the actual implant. When this is finished the actual treatment with the radioactive source can be executed.
This is a remote process, the patient with implant is positioned in a bunker and the afterloader with radioactive source places the radioactive source automatically according the approved treatment planning.

Actors in this use case:

- Imaging laborant who makes the CT/MR/US images of patient with or without implant
- Physicist who makes the treatment planning based on the images.
- Doctor who makes the implant (mostly an operation with anesthesia)
- Physicist again to update the actual treatment planning, and approve the planning
- Laborant to connect the afterloader to the implant in the radiation bunker
- And Laborant to control the radioactive delivery according the approved plan and approve treatment record.

The approach to this use case is rather general, which means the Brachy treatment and planning is described by means of a high level workflow. Then use will be made of a not yet existing method to do a "minimal invasive implant" for Brachytherapy which makes in this case the workflow faster and simpler than the actual existing workflows for the so called "image guided Brachy therapy"
Imaging can be X-ray, CT, MRI SPECT and ultrasound.
Implants are in general catheters, which can be needles, catheters, or so called applicators, which are in fact complicated forms of catheters appropriate for radioactive sources, and specific for different body sites for Brachytherapy treatment. (cervix, vaginal, rectum, prostate, breast and so on)

4.2.1.2 Basic Flow for Image guided conformal Brachy

The idea is to describe Brachytherapy flow which is supported by 3D imaging so that tumor and organ are planned in a 3D imaging setting, and treated according the 3D planning. The treatment and planning data are described in 3D patient coordinates. This means that the implant is part of the conforal 3D planning, and the Brachy implant is represented in 3D patient coordinates.

4.2.1.2.1 Basic minimal invasive Barchytherapy workflow

1) Diagnostic imaging; 3D Imaging of the patient, determine tumor and organs at risk.
   Can be some weeks before the actual treatment takes place.

2) Pre-planning; Make 3D conformal pre-planning based on the diagnostic 3D images.
   The pre-plan can be made some day before patient comes in for treatment.

3) Therapy imaging; The patient comes in and an actual 3D image of the patient is required.
   Tumor size position or patient itself are changed.

4) Actual pre-plan; Update pre-plan to actual tumor and organs at risk.
   Based on diagnostic or therapeutic imaging, dependant on hospital or OK (operation room) imaging available.

5) Brachy Implant; place catheters according the pre-plan.
   In general the implant will deviate from the planned implant since the implant itself will deform the patient and position of tumor and organs, and user will have deviations in placement itself.

6) Actual treatment plan; Final update of the Brachy plan with the actual position of the implant related to the actual 3D image of patient, with implant, tumor and organs at risk.

7) Connect implant to afterloader; The afterloader is connected to the implant by means of transfer tubes. The user checks the correct connection of afterloader to the catheters according treatment plan.
   The physicist checks the actual “radio activity” of the radioactive source, by which dwell times are updated to give the planned dose in the patient. Approval of the treatment is performed.

8) Radioactive treatment; all personal goes out of the room. The radioactive treatment is started remote from the control room. Radioactive source steps to dwell positions with defined dwell times through the implant according the Brachy treatment plan,

9) Finish treatment; The treatment is closed at the control room.
   The radioactive source is back in the vault of the afterloader. The room is safe and the patient can be uncoupled. A treatment record is made so that the treatment activities are stored in DICOM format and can be used for verification.

4.2.1.2.2 Alternative minimal invasive Barchytherapy workflow

1) Diagnostic imaging.
2) Pre-planning (3D conformal planning)
3) Therapy imaging (In the OK, operation room, before the implant)
4) Actual pre-plan (pre-plan to actual tumor and organs at risk).
5) Image guided Brachy Implant.; place catheters according the pre-plan.
   Applicator, needles, or catheters are placed under real time image guidance.
   Well known methods are real time ultrasound (2D), X-ray fluoro, real time projective imaging, or non real time CBCT fast position verification
6) Conformal registration of the implant.
   3D imaging is required to register the implant in relation to the patient, tumor and organs at risk.
   In Brachy planning an “applicator library” acts as a help tool to reconstruct the implant in the 3D image, the implant is registered in patient 3D coordinates in a DICOM Brachy file.
7) Actual treatment plan; Final update of the Brachy plan with the actual position of the implant related to the actual 3D image of implant , patient with tumor and organs at risk.
8) Connect implant to afterloader; and Approve plan
9) Radioactive treatment, remote process by afterloader.
10) Finish treatment, make treatment record.

4.2.1.3 Architecture for minimal invasive treatments

4.2.1.3.1 IGIT architecture

See for an overview of the architecture Figure 2. Figure 2 shows different high level aspects of the IGIT architecture. The following aspects are shown:

- Interventional controller, shown on the top-middle of Figure 2
- Sensors, shown on the left side of Figure 2. Only a limited number of sensors are shown to simply the figure. Other sensors will be placed similar in the architecture.
- External systems, shown on the right side of Figure 2. The external systems contain both pre-interventional imaging modalities (e.g. CT and MR) and, PACS and RIS systems.

The architecture of the internal controller is shown in **Error! Reference source not found.**
It can be seen that the internal controller is composed of a number of applications. The applications have drivers which connect with the sensors, by which the examples here are realistic chosen, also in the Brachy use case with a needle steering and navigation application, and depending on the real time imaging in the operation room an X-ray fluoro or ConeBeam CT application, and for future real time imaging an ultrasound (2d or 3D) application

4.2.1.3.2 Applications and workstations

Brachy case specific setting for the workstations
Most of the time two workplaces are used for the Brachy setting:

1. workstation A in the operation room for the real time and interventional applications
2. workstation B in planning room for static planning application outside the operation room.
   In practice planning is executed on separate departments of hospital floors and the connection between the workstation is over the hospital network.

4.2.1.3.3 Communication between the applications

The so called static data are data which are transferred between the applications and from the internal controller over the network to hospital data management systems. Brachy treatment planning records, and Brachy treatment delivery records and patient data sets are stored via DICOM in therapy patient information management systems, like Mosaiq and Aria, and diagnostic patient management systems as PACs and RIS.
DICOM:

- DICOM RT image (CT, MR, US, and so on.)
- DICOM RT struct (radio therapy structure set, in which the targets and organs are contoured)
- DICOM RT plan (the Brachy plan in which dwell positions and dwell times, source activity)
- DICOM RT treatment record (the result of the treatment records the actual dwell times and positions)
- DICOM RT dose (the dose distribution in the patient calculated by the planning system)

Important to know that all positions inside the DICOM sets are so called patient 3D Coordinates; which means that the systems for sensoring and image registrations must be calibrated to a defined patient coordinate system. A zero position (0,0,0) must be defined, mostly called the **iso center**.

Oncology patient management systems are for example Aria(Varian) and Mosaiq (Elekta.)

MOSAIQ is a complete patient information management system that centralizes radiation oncology, particle therapy and medical oncology patient data into a single user interface, accessible by multi-disciplinary teams across multiple locations. As a true open systems solution, MOSAIQ connects seamlessly to any linear accelerator or Brachy afterloader and treatment planning system from any vendor, giving centres the flexibility to choose the optimal treatment solutions for their practice and their patients.

ARIA is a comprehensive information and image management system that aggregates patient data into a fully-electronic medical chart. ARIA's oncology-specific EMR can streamline care delivery and prepare your practice for the future of medical information management.

ARIA is a comprehensive software solution designed to manage all of your department’s clinical and administrative activities. The electronic medical record (EMR) is rapidly becoming the standard for managing patient information. In keeping with the times, ARIA aggregates treatment records and digital images into a complete, oncology-specific EMR.

4.2.1.4 The Brachy therapy applications

4.2.1.4.1 Brachy planning system

1) Import DICOM image set
2) Contour in image set, target (tumor) and organs at risk, result structure set DICOMRT struct.
3) General dose setting; Set radioactive source type and strength, and total dose for body site
4) Set dose constraints, for example for a dose optimization module with requirements for minimum dose on target, and maximum limits for organs at risk.
5) Define catheters in the 3D image set.
   Catheter position can be calculated by an optimization module, or user defined manually placed.
6) Define per catheter the dwell positions and dwell times for the radioactive source.
   The dwell positions and dwell times are mostly done by an optimization module.
   Manual adaptation of dwell times and positions is possible
7) Store the Brachy plan in DICOM set, patient and session dependant.
   resulting in DICOM RT image, RT struct, RT plan and RT dose

Pre-conditions
DICOM 3D image sets must be available on which target (tumor) and organs at risk are visible
Sometimes more image modalities are used to achieve the required visibility to contour the target and organs at risk. So called image fusion is then part of the Brachy planning.
Visualisation tooling to represent 3D images by means of 3 2D views; transversal, sagittal, and coronal cross sections.

Post-conditions
Evaluation tools are part of the Brachy planning system to verify the dosage in the patient.
In fact visualization tools to check dosage in target and in organs at risk.
Several presentations to visualize dosage distributions exist.

4.2.1.4.2 Implant guiding system

1) Import Brachy plan (DICOM)
   The catheter positions are defined in the RT plan DICOM data, as dwell positions in 3D patient coordinates.
2) Calibrate the isocenter position and directions in the 3D patient coordinate system.
   Can be done in software as well as in hardware. Critical part of preparation in radiotherapy
3) Present catheters in 3D view
   At the moment three 2D images represent the 3D image (sagittal, transversal, and coronal view)
4) Select catheter to be placed.
5) Visualize the catheter placement by real time imaging.
   Can be ultrasound, MRI transversal or sagittal image, or with X-ray projective images
6) Manipulate and place catheter according Brachy plan data under image guidance.
   When a catheter or needle template is used (see prostate) the position and direction of the needle are defined and controlled by the template.
7) Do all catheters by repeating 4 and 5.

Pre-conditions

Calibration of iso centre position and directions
The DICOM RT struct and DICOM RT plan by which target and organs, and the catheter positions are based on the patient coordinate system of the image device (The MRI or CT).
The implant is in real practice and is defined by “room coordinates”.
The room coordinates of the implant must be matched with the patient coordinates.
Real time visualization is mostly done by means of 2D imaging (MRI, US, X-ray),
or in exceptional cases 3D US is possible.

Post-conditions

The user verifies that all catheters are placed.
Making a fast image and count (for example an X-ray image or CBCT scan).

4.2.1.4.3 Implant reconstruction

1) In the operation room make a 3D image scan
2) Set/match coordinate system and directions to patient coordinate system
3) Automatically or manual reconstruct catheters in the 3D image.
4) Store image set as a DICOM RT image and catheter positions in DICOM RT plan

Pre-conditions
The catheters must be visible in the 3D image
Sometimes markers are needed to help the reconstruction, this means hardware adaptations on the catheters are necessary
The 3D scan is in room coordinates, so again the 3D scan in the room must be matched to patient coordinates and directions.

Post-conditions
The implant reconstruction must be re-calculated so that this 3D image with catheter coordinates can be stored in RT image and RT plan. Since the catheter positions are originally represented by dwell positions in “patient co-ordinates” in the Brachy plan

4.2.1.4.4 Connection of afterloader to implant

1) Import DICOM RT-plan
2) Present catheter connections on screen
3) Connect afterloader canals to catheters
4) Verify correct connection
5) Verify all catheters are free to transport the radioactive source

Pre-conditions
Check correct patient and lesion

Post-conditions
When catheters or transfer tubes are blocked solve blockage before going into treatment delivery.
Change canal or catheter when needed, update delivery plan so that catheters have the correct dose.

4.2.1.4.5 Treatment delivery

1) Approve canal – catheter connections
2) Verify actual radioactive source activity and calculate final dwell times.
3) Approve treatment delivery plan.
4) safety and alarms for radio activity checked
5) Run radio-active treatment
6) Store actual dwell positions and dwell times per canal(catheter)
7) Finish treatment and store delivery in DICOM RT treatment record

Pre-conditions
The Brachy plan has calculated dwell times with a virtual source activity of 10 Ci (high dose rate Brachy) The actual source activity is calculated from the radioactive source calibration.
The calibration data are stored in the afterloader firmware and imported in the treatment delivery application, see action 2.

Post-conditions
The treatment record data are available in RT treatment record DICOM format and can be imported by a Patient management system.

4.2.2 Pre-Interventional reconstruction

The pre-interventional processes are directly launched by users. The operation flow is as follows:

- The user, via a desktop client application, authenticates in the grid system with a certificate. This certificate is unique to each user.
- The user sends the simulation from the client application. The grid server transfers the simulation to the RMS cluster with permissions delegated by the user. All cluster machines are configured with all necessary dependencies to run the simulation using MPI and HDF5.
- Once the simulation is finished on the RMS, the grid system detects this event and sends the output files either to the interventional controller or to an external repository such as the PACS.

4.3 Physical view

This section describes the physical view on the architecture. This physical view consists of the physical location and the physical connections of components in the system.

4.3.1 Overview

The figure below gives an global overview of the physical view of the IGIT system.

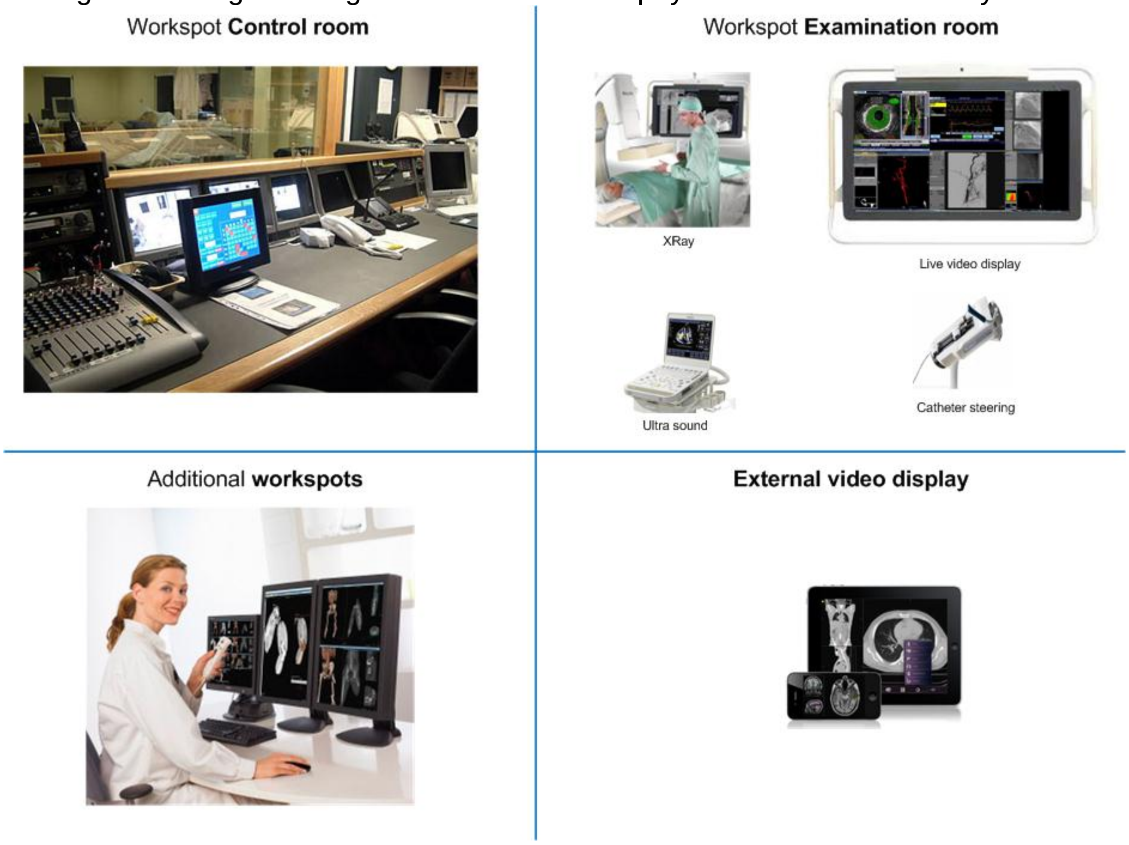

Workspot **Control room**
Workspot **Examination room**
XRay
Live video display
Ultra sound
Catheter steering
Additional **workspots**
External video display

Figure 12 Physical IGIT architecture

There are 4 different work-spots defined. Each work-spot is briefly described in the following sections.

4.3.2 Workspot Examination room

The patient is located in the examination room. Therefore, the imaging modalities, e.g. X-ray, US etc. are located in this examination room. From this examination the modalities can be controlled and the output the IGIT system, in terms of video, is shown real-time on the displays, see section 4.1.5.1.

4.3.3 Workspot Control room

For most IGIT treatments X-ray will be used. To protect the operator for this X-ray radiation, the control room is situated in a separate room. The control room can be used to:

- Control the modalities, e.g. start/stop X-ray, catheter steering
- Patient administration activities e.g. retrieve new patient information from the RIS

- Post processing / analysis of clinical data
- Control connections with PACS systems.

Also, the output the IGIT system in terms of video is shown real-time on the displays.

4.3.4 Additional Workspots

Next to the control room, which is situated next to the examinations room, additional work-spots are required. These work-spots can be used for:

- Post processing / analysis of clinical data
- Control connections with PACS systems.

These work-spot can be located locally (inside the hospital) and remotely (e.g. via Internet connection).

4.3.5 External Video display

The video output of the IGIT system can also be shared externally (see section 4.1.5.2):

- For consulting another physician
- For educational purposes

5 References

5.1 References to Mediate deliverables

Tag	Description	ID
[D211]	Security and privacy requirements	Mediate-D2.1.1
[D212]	Technical report focused on the use cases	Mediate-D2.1.2
[D221]	Standards and Interoperability	Mediate-D2.2.1

5.2 References to external sources

Tag	Description	ID
[IEC42010]	System and software engineering – Recommended practice for architectural description of software-intensive systems	ISO/IEC 42010 First edition 2007-07-15
[FIN2011]	*A review of ECG storage formats*, by Raymond R. Bond, Dewar D. Finlay, Chris D. Nugent and George Moore	International Journal of medical informatics 80 (2011)
[GIMIAS]	www.gimias.org	www.gimias.org
[IDL]	http://www.exelisvis.com/ProductsServices/IDL.aspx	IDL

5.3 References to other Itea2 projects

Tag	Description
[HIGHPROFILE]	Itea2 project High Profile
[HIPIP]	Itea2 project Hipip
[CARE4ME]	Itea2 project Care4Me