

# Healthcare IT – A Monitoring Primer

## Contents

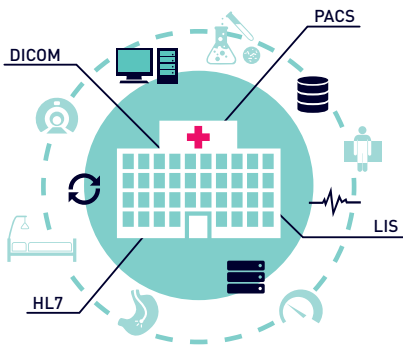
Introduction .....	3
The Healthcare IT Environment.....	4
“Traditional” IT .....	4
Healthcare Systems.....	4
Healthcare Data Format Standards & Communication Protocols.....	5
Medical Devices .....	5
Monitoring Healthcare IT.....	6
Integration Engine .....	6
How to monitor an integration engine .....	6
EHR / EMR .....	7
How to monitor EHR components .....	7
Modalities .....	7
How to monitor modalities .....	7
PACS (Picture Archive and Communication System) .....	8
How to Monitor the PACS.....	8
LIS, RIS, and HIS .....	9
How to monitor LIS, RIS, and HIS .....	10
Network .....	10
Monitoring Software – What to Look For.....	10
Workflow Monitoring .....	11
Use Case – IHE Radiology Scheduled Workflow .....	11

## Abstract

This primer introduces some of the core systems of a healthcare organization and introduces best practices to ensure a complete and reliable monitoring solution specifically for healthcare IT infrastructure.

## Introduction

Hospital IT infrastructure provides IT professionals with a unique challenge: they have the “traditional” IT elements to take care of, and then on top of that, there are the specialized healthcare systems and protocols. All of these elements and systems co-exist in the same infrastructure, and this brings with it some problems.



Infrastructure and network monitoring is not new to IT professionals, and most medical environments are probably already monitoring at least parts of their traditional IT infrastructure. And most IT professionals understand the importance of a good monitoring tool.

But what is a good monitoring tool supposed to do, and how should it be applied to medical environments?

Put very simplistically, monitoring software should do the following:

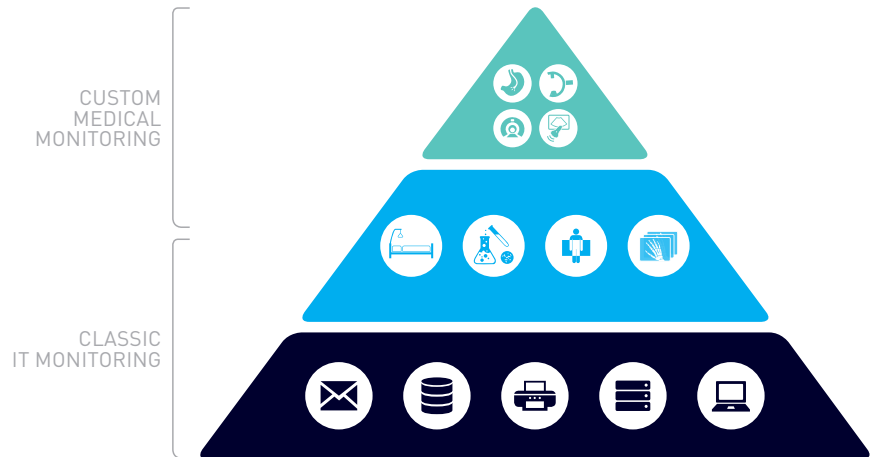
- Monitor the network speeds, and check for bottlenecks, etc. It should do this using common network protocols like SNMP, Netflow, WMI, and so on.
- Monitor devices such as routers, servers, storage, and so on.
- Provide alerts and notifications when certain thresholds are reached, like when bandwidth is running low or when a device becomes overheated.
- Display the status of the infrastructure in a single dashboard view.

This last point is an important one for any environment: to be able to see what the overall health of your IT infrastructure is just at a glance. This applies to healthcare environments as much as any other. And while there is a lot of information out there about how to monitor your IT, it becomes a bit more difficult when considering healthcare IT. This is because a typical hospital environment includes many disparate devices, systems, and protocols.

What we will do in this primer is consider each system and area of a typical healthcare infrastructure, and explain what you need to monitor, and how.

# The Healthcare IT Environment

The modern hospital IT infrastructure can be thought of as having several main aspects: the **“traditional” IT**, the **healthcare systems** with their associated **data formats and communication protocols**, and **medical devices**. If you think of it as a pyramid, it looks something like this:



## “Traditional” IT

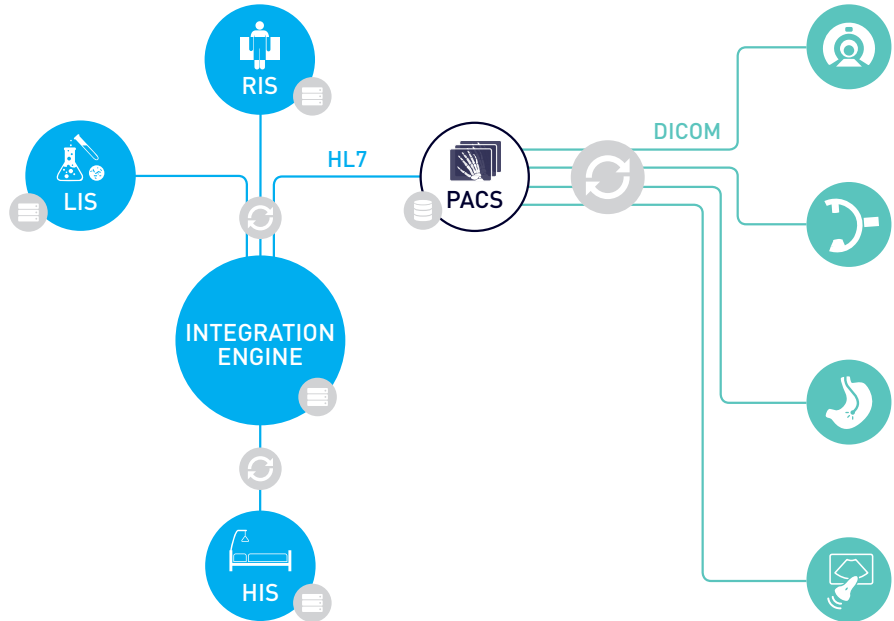
A hospital environment has infrastructure similar to that which you would find in any IT environment, and is used to connect things like workstation desktops, printers, servers, and so on. It also consists of the usual devices such as routers, switches, and domain controllers.

## Healthcare Systems

There are various kinds of systems that can be found in hospitals. The table below lists some of the most common ones.

System	Description
<b>Integration Engine</b>	Central software for handling message distribution between various systems. It receives, modifies and distributes messages or data in multiple formats, usually HL7 and DICOM, but also FHIR and HTTP-based requests.
<b>Radiology Information System (RIS)</b>	System for managing imaging departments digitally. Includes functionality for patient scheduling, resource management, procedure billing, and more.
<b>Laboratory Information Management System (LIS/LIMS)</b>	System for supporting laboratory's functions, such as data exchanges between a hospital and the laboratory.
<b>Hospital Information System (HIS)</b>	Supports the administration and operational functions of a hospital, such as making patient details available, producing invoices, and processing of services.
<b>PACS (Picture Archiving and Communication System)/ VNA (Vendor Neutral Archive)</b>	System for storing and accessing image data from multiple modalities (or imaging devices).
<b>Electronic Health Record (EHR)/Electronic Medical Record (EMR)</b>	Central repository for patient information, including patient demographics, medical history, and medical data like vital signs and laboratory results.

Usually, the integration engine forms the hub of the architecture. The way that the systems might be connected and interface with each other is shown in this diagram:



## Healthcare Data Format Standards & Communication Protocols

As you can see in the figure above, there are two protocols that are commonly used to facilitate the communications between the systems.

Data Format Standard	Description
<b>DICOM (Digital Imaging Communications and Management)</b>	Used to transfer and store image data from radiology, CT scans, and ultrasound imaging in a central system (usually a PACS).
<b>HL7 (Health Level 7)</b>	Used for patient data distribution and ordering between the integration engine and multiple other systems.
<b>FHIR (Fast Healthcare Interoperability Resources)</b>	Used to exchange Electronic Health Records (i.e. global patient details) between hospitals, clinics, doctors' offices, and so on. It is based on a RESTful API. This is not shown in the figure above.

## Medical Devices

There are several types of devices that connect to the hospital's network.

One specific category of devices, the imaging devices, is called "modalities". Included in this category are devices like MRI machines, CT Scanners, ultrasound devices, and X-Ray machines.

Now that we have established exactly what we mean by "healthcare IT", we can take a look at how it can be monitored.

# Monitoring Healthcare IT

In this section, we will take a look at some common systems in healthcare IT, and consider how they can be monitored.

## Integration Engine

The integration engine is a central and important piece of software for message distribution between the different medical system like HIS, EHR, RIS, and LIS. The integration engine connects multiple systems by receiving, modifying and distributing messages or data in multiple formats. These messages are usually HL7 and DICOM messages, but could also be FHIR- and HTTP-based requests.

The interfaces that receive and distribute messages are called “Channels” and these can be “Inbound” or “Outbound”. In most scenarios at least two channels are required, as the inbound channel receives data from a source system and the related outbound channel transmits the data to the target system. Most integration engines also support caching or saving messages that could not be sent due to network errors or problems on the target system. This means that these messages can be sent again when the target is available, ensuring that no messages are lost. In case of a failure the integration engine will also alert the designated IT specialist as to the problematic channel.

As you can see, it is vital that the integration engine is always up, 24/7. Because of its importance in a healthcare infrastructure, it must be part of the monitoring concept. Monitoring the integration engine will not only provide you with information about its health, but also give you clues as to the status of all other connected systems or devices.

### How to monitor an integration engine

Most vendors provide a REST API that you can use to bring the integration engine into your monitoring solution. Typically you can retrieve two types of statistics from this API:

- the **general system health** like free memory, free disk space, and CPU load.
- **Channel-centric statistics**, including the amount of received or sent messages and counters for error, queued or filter messages based on the channel type.

While it is possible to monitor the integration engine using the REST API, you also want to consider monitoring it “externally” using monitoring software. The reason: if the integration engine fails, it could be that the REST API is not accessible.

Here are some aspects of the integration engine you can monitor with monitoring software:

- **Number of messages processed for each channel** of the integration engine. If too many or too few messages are processed, or they contain errors, you might have an issue somewhere in the network. Once you’ve identified an anomaly in the number of messages, you can then investigate the connected systems to establish the cause of the issue.
- **Available RAM on the integration engine server.** This is vital to monitor, since a huge number of messages are processed in multiple workflows by the integration engine. If it runs out of memory, processing or transmitting some messages might be delayed.
- **Read and write latency of the connected database.** Keep an eye on the database the integration engine writes to and reads from. If the latency for these actions is too high, the integration engine might have to wait to finish certain tasks.

## EHR / EMR

The EHR (Electronic Health Record) or EMR (Electronic Medical Record) is a digital format for recording patient information, and includes patient demographics, medical history, medical data, and laboratory test results. Records can be shared between health institutions using Enterprise Document Sharing (XDS).

In a hospital's infrastructure, there is probably hardware – usually a server – that is used to store patient details, and to access the Health Information Exchange (HIE), a central repository where all patient details are kept.

The EHR is crucial to diagnosing patients and treating them. It provides the patient's history, past treatments, vital signs, and more, and these things need to be considered together when making decisions about treating the patient. To make sure that the EHR is always accessible and updated, it is important to monitor the network and hardware components involved with EHR processing.

### How to monitor EHR components

- **Basic hardware monitoring.** The usual principles of monitoring hardware apply, such as checking available memory, CPU load, etc.
- **Monitor backups.** Patient data is vital, so monitor backups to make sure they complete successfully, that there is enough storage space, and so on.
- **Read and write latency of the connected database.** As with the integration engine, you should check the latency of read and write operations to the HIE database.
- **Ensure that the EHR is accessible from the Health Information Exchange repository.** Ensure that the HIE URL is accessible. You can also check the interface (usually XDS or HL7/FHIR).
- **Get information using the EHR components' APIs (if possible).** Many EHR vendors include an API that you can use to integrate your monitoring solution. The metrics from these interfaces can provide detailed insights into the performance and availability of the components.

## Modalities

Modalities refer to the devices and equipment used for creating medical images, such as Magnetic Resonance Imaging, X-rays, Computer Tomography Scans, and ultrasounds.

The images captured by modalities are usually stored on a Picture Archiving and Communication System (PACS), where they can be retrieved or transmitted by doctors. Usually, this imaging data is transferred and accessed using Digital Imaging and Communications in Medicine (DICOM).

Each modality has a worklist associated with it. The worklist stores the jobs that are scheduled for the specific modality.

### How to monitor modalities

Modalities should form part of a monitoring strategy. However, there is a challenge with monitoring them: by their nature, they have a “closed” architecture. They do not use standard protocols like SNMP or Netflow, and they generally do not provide APIs that can be used to access statistics about the devices. You could ping the devices, but this tells you very little about the status of the device, other than it is on and responding. In many cases, there is specific vendor software that lets you monitor the device, but the problem with this is that it is difficult to integrate this with your existing monitoring solution.



One way to bring these modalities into a unified monitoring picture is to use the mechanism that they communicate with: DICOM.

- **Modality availability and response times.** Use the **DICOM C-ECHO** request for this. Ideally, you should also ensure you get a notification if the response times of the DICOM devices become too high.
- **Count of worklist entries.** The number of worklist entries for a modality can show if there is a problem with the device. For example: if the number of worklist entries keeps growing, it means older entries are not being cleared. This could happen when there is a bottleneck or malfunction. If there are fewer entries than usual, it could mean that new requests are not being processed by the device. You can use the **DICOM C-FIND** request to query the number of worklist entries for modalities. Using it, you can count the number of worklist items associated either with a specific modality, or across all the modalities. Set alert thresholds so that you know when you are above or below your desired number of worklist entries.
- **Images stored locally on the device.** If images are stored locally on an imaging device, it might indicate an issue, since it could mean the modality is not able to store the images on the PACS or storage. Possible causes include a network outage or a configuration error. Use **DICOM C-FIND** to check if there are series or studies stored on the device. If there are, you might have a problem. This is useful if you have devices that connect using ethernet instead of WLAN, and which are moved from room to room. Checking for local files on these devices can help you ascertain if images are being archived correctly.

## PACS (Picture Archive and Communication System)

PACS is the central image archive and distribution platform of a hospital or a radiology department. This is where images from different modalities (medical devices that generate images) are stored and also accessed by doctors and specialists. Older images are archived and made available should the patient return in the future. Also known as “prefetch”.

In a typical hospital setup, the PACS connects to the modalities, the RIS, and the integration engine. It forms a central part of many radiology workflows, which subsequently means that problems with the PACS can have a big impact on the general functioning of the hospital. But what could these potential problems be?

A PACS needs a large amount of disc capacity in the storage it is attached to. It also needs very high transfer rates to guarantee stability. But this is just the start. There are many aspects of a PACS that should be monitored to ensure the overall stability of the radiology and other related workflows.

### How to monitor the PACS

- **Hardware that the PACS runs on.** This includes servers and any services required by the PACS. Monitor things like CPU load, network throughput, and available RAM.
- **Available disk space.** Although this is actually a part of hardware monitoring, it is so vital to PACS that it needs a separate mention. Depending on your PACS architecture, you should focus on the disks where the database, application data, and images are stored. And if these are provided by a storage like Dell EMC or NetApp you should monitor these volumes and luns as well.
- **Read/Write latency between PACS and the storage.** In every situation where files are transferred between the PACS and storage, a low latency is required. If no sufficient throughput can be provided for tasks like image pre-fetch, short-term to long-term transfer etc., the overall performance of the PACS could decrease.



- **PACS API and Log Files:** Many PACS components provide insights with an API or in log files. It makes sense to monitor these with your monitoring solution, too.
  - API: Data from the API gives you statistics about the current application performance and the overall status, such as number of DICOM requests received, number of errors, and the status of internal processing queues.
  - Log files: Keep track of important system events, such as failed authentication attempts or internal PACS failures.
- **Application interfaces.** These interfaces in particular should be monitored:
  - DICOM interfaces: There are various ways to check that your DICOM interfaces are functioning correctly.
    - » **C-STORE** capability of DICOM interfaces to make sure it is possible to store images, and that there is enough bandwidth for it.
    - » **C-MOVE** and **C-FIND** requests to check the search and transfer of images.
    - » Monitor the basic connection and health of your DICOM interfaces using **C-ECHO** calls. The response time and returned values from your queries will help you to detect performance issues on certain scenarios e.g. high traffic periods in the afternoon.
    - » Check the **DICOM Worklist** to ensure you solve problems before they influence your workflows. For example: detecting old entries which aren't removed from the worklist improves the performance and reduces the risk of selecting incorrect entries.
  - **HL7 interfaces.** If HL7 messages are not transmitted correctly, or if they are incomplete, delays and other problems can occur. HL7 interfaces can be monitored either using APIs and log-files, or by sending dummy HL7 messages, which are flagged as test messages.
  - **User Interface.** Quite often, the PACS system is accessed using the Web interface for DICOM (called Web Access for DICOM Objects, or WADO). Monitoring the general availability and responsiveness of WADO can help detect poor user experience.
  - Monitor backup clusters to check on High Availability. If you have deployed High Availability in your environment (which is highly recommended), then you can also check on the backup cluster.

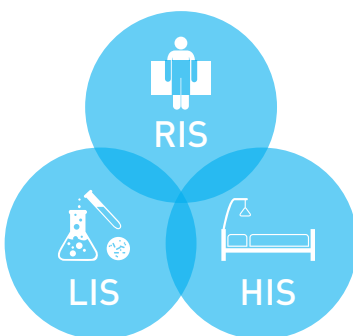
If you use a Vendor Neutral Archive (VNA) to archive certain images, you can use the above monitoring strategies for that, too.

## LIS, RIS, and HIS

There are several other systems that are usually found in a healthcare IT environment, namely:

- LIS, for ordering laboratory tests, managing the results, and generating reports
- HIS, for managing the administrative aspects of a hospital
- RIS, for managing radiological workflows

The nature of these systems differs depending on vendor and implementation, but the monitoring requirements remain similar across the board.



## How to monitor LIS, RIS, and HIS

- **Basic hardware monitoring.** The usual principles of monitoring apply for the hardware these systems run on, such as servers and storage. This includes monitoring things like available memory, CPU load, etc.
- **HL7 interfaces.** These systems usually all connect to the integration engine, and transmit their data using the HL7 protocol. You can use dummy HL7 messages to check on each interface. Essentially you can send a test HL7 message, get a response, and then compare the messages to ensure that everything is working correctly.
- **FHIR interfaces.** For FHIR interfaces, you can use the RESTful API to check that the interfaces are up and working.

## Network

Aside from the healthcare-specific systems and protocols, there is also the “traditional” IT infrastructure, which consists of firewalls, routers, switches, and so on. This requires network monitoring that is very similar to any other environment. Just some examples of what you can monitor:

- Network devices, such as routers, switches, firewalls, and so on
- Bandwidth usage in the network
- QoS for medical data in intensive care units to ensure real-time alerting
- Access points for tablets, medical cards, etc.



## Monitoring Software – What to Look For

The network monitoring solution for healthcare IT environments needs to monitor both traditional and healthcare IT-specific network elements. Here are some things to look out for when considering monitoring software:

- ✓ **Traditional Network Monitoring**  
This includes the ability to monitor routers, switches, firewalls, storage devices, and support for common protocols like SNMP, NetFlow, and others.
- ✓ **Healthcare IT Elements**  
Make sure the monitoring software can cover the solutions mentioned in the previous section (“Monitoring Healthcare IT”), which includes (among other things) the ability to monitor using the DICOM and HL7 protocols.
- ✓ **Dashboards**  
You need to be able to visualize the IT infrastructure, and preferably see it all on one screen. This means including both traditional elements and healthcare IT elements into the same overview, with the ability to drill down if you need to.  
**For example:** at the very top level, you could have the status of all your modalities shown on the overview dashboard. If there is a problem with one of the modalities, the status will show that there is an issue. You can then drill down to see which modality has the issue.
- ✓ **Alerts and Notifications**  
Since you can’t watch dashboards all the time, you need to be alerted when problems arise. The infrastructure monitoring solution you select should let you set thresholds for certain parameters, and notify you when they are exceeded.  
**For example:** a rising number of worklist entries for modalities could mean that worklist items are not being cleared—probably due to a network issue. In this case, when monitoring the number of worklist items, you would set a specific threshold. When this threshold is exceeded, the status is set something like “Warning” and a notification can be triggered.

✓ **Workflow Monitoring**

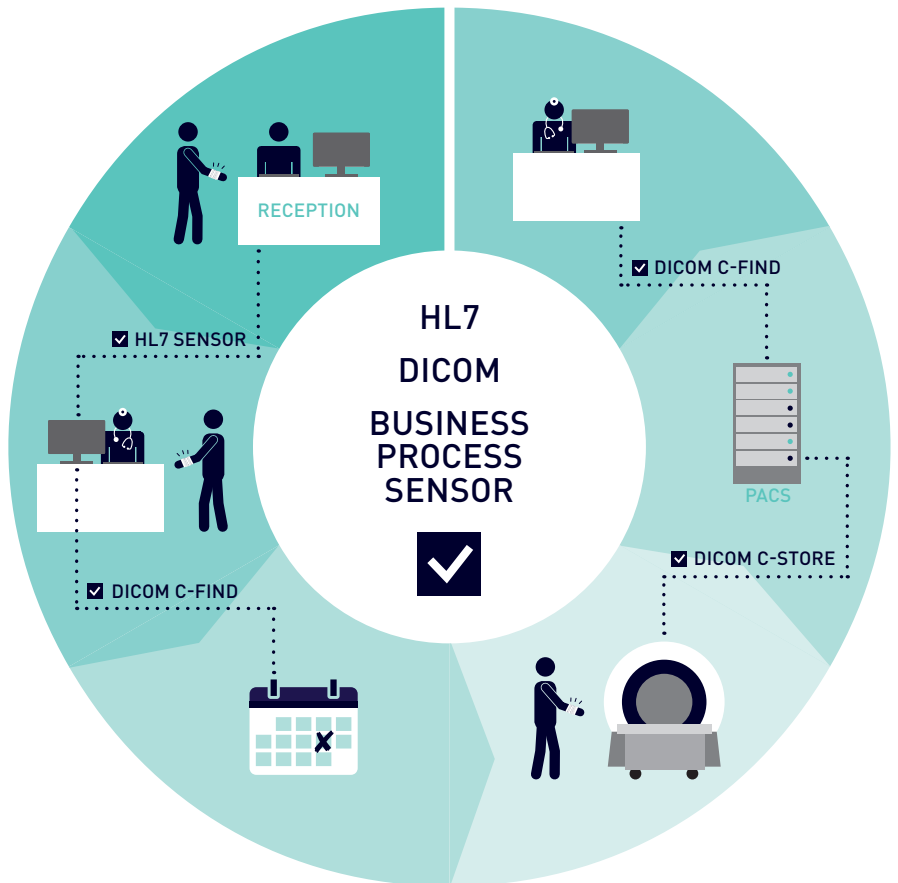
Realistically, the infrastructure in a hospital supports a series of workflows that consist of all kinds of devices and protocols. For example: a radiology workflow makes sure the patient goes from registration at reception, through their scans, to a diagnosis at the end. If any of the systems or interfaces fail, the workflow breaks down.

The monitoring solution you choose should let you build overviews of entire workflows, such as radiology workflows, laboratory workflows, and so on. For an example of how this works, take a look at the Workflow section below.

## Workflow Monitoring

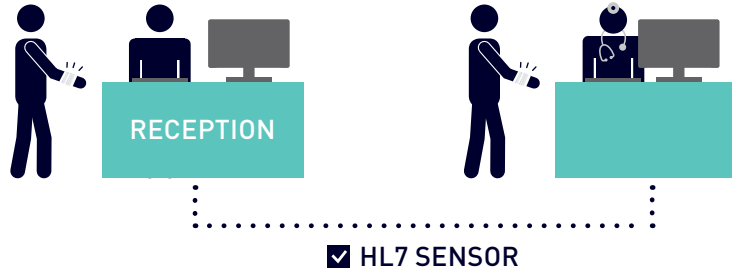
### Use Case – IHE Radiology Scheduled Workflow

IHE Radiology provides the technical framework for radiology workflows in a healthcare environment. To demonstrate how the monitoring concept discussed in this paper might be implemented, we take a radiology scheduled workflow as an example.



The radiology scheduled workflow is the technical framework covering the following aspects of radiology examinations:

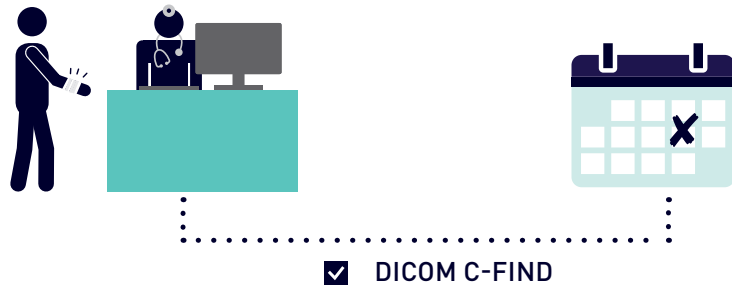
1. Register the patient and send the details to the attending doctor



The patient is registered in the HIS, and their details sent to the RIS, where the doctor can access them. The data is transferred using the HL7 protocol.

**How to monitor it:** Check that the HL7 interfaces are up and transferring information correctly.

2. Order and schedule the examination



The doctor places an Order Request with RIS to set up an appointment for the scan. This is also done using RIS over HL7.

**How to monitor it:** Check that the HL7 interfaces are up and transferring information correctly.

3. Storing the images



The images produced by the scan are stored to the PACS using a DICOM C-STORE request.

**How to monitor it:** Make sure that the images can be safely stored. Do this by checking that the DICOM devices (in this case, the modality and PACS) are up using the DICOM C-ECHO request, and by testing the response time and bandwidth used when executing a C-STORE request. You can also check on the modality's workload using DICOM C-FIND to make sure workload items are being cleared out properly.

4. Retrieve the images

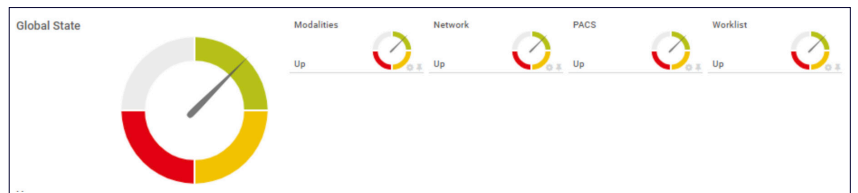


The doctor accesses the patient's images on the PACS using the DICOM C-FIND request.

**How to monitor it:** Use the C-ECHO request to regularly check on the PACS, and use C-FIND to test if images can be retrieved.

5. Monitor workflows by including all related devices, services etc. to ensure workflow is functional.

Include all the above monitoring metrics in one view, and create an overall status of the workflow. If there are problems with any individual step, the workflow status shows a warning. You can then drill down to each step to identify where the problem is.



In 1997 Paessler revolutionized IT monitoring with the introduction of PRTG Network Monitor. Today over 200,000 IT administrators, in more than 170 countries, rely on PRTG to monitor their business-critical systems, devices and network infrastructures. PRTG monitors the entire IT infrastructure 24/7 and helps IT professionals to seamlessly solve problems before they impact users.

Our mission is to empower technical teams to manage their infrastructure, ensuring maximum productivity. We build lasting partnerships and integrative, holistic solutions to achieve this. Thinking beyond IT networks, Paessler is actively developing solutions to support digital transformation strategies and the Internet of Things.

Learn more about Paessler and PRTG at [www.paessler.com](http://www.paessler.com).

Paessler AG · [www.paessler.com](http://www.paessler.com) · [info@paessler.com](mailto:info@paessler.com)

**NOTE:**  
All rights for trademarks and names are property of their respective owners.