

# Module 20 Medical Imaging case, CAFCR illustration

by *Gerrit Muller*      University of South-Eastern Norway-NISE

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

## Abstract

This module provides a complete illustration of the CAFCR based architecting method. The case is a Medical Imaging Workstation, created in the early nineties.

### Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

September 1, 2020  
status: finished  
version: 0

logo  
TBD

# Medical Imaging in Chronological Order

by *Gerrit Muller*      University of South-Eastern Norway-NISE

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

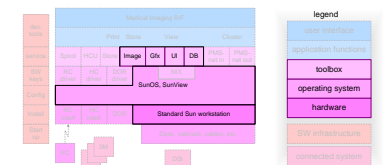
## Abstract

The chronological events of the product creation of the medical imaging workstation are discussed. The growth in functionality and size from prototype to product is shown. Typical problems in this period are explained.

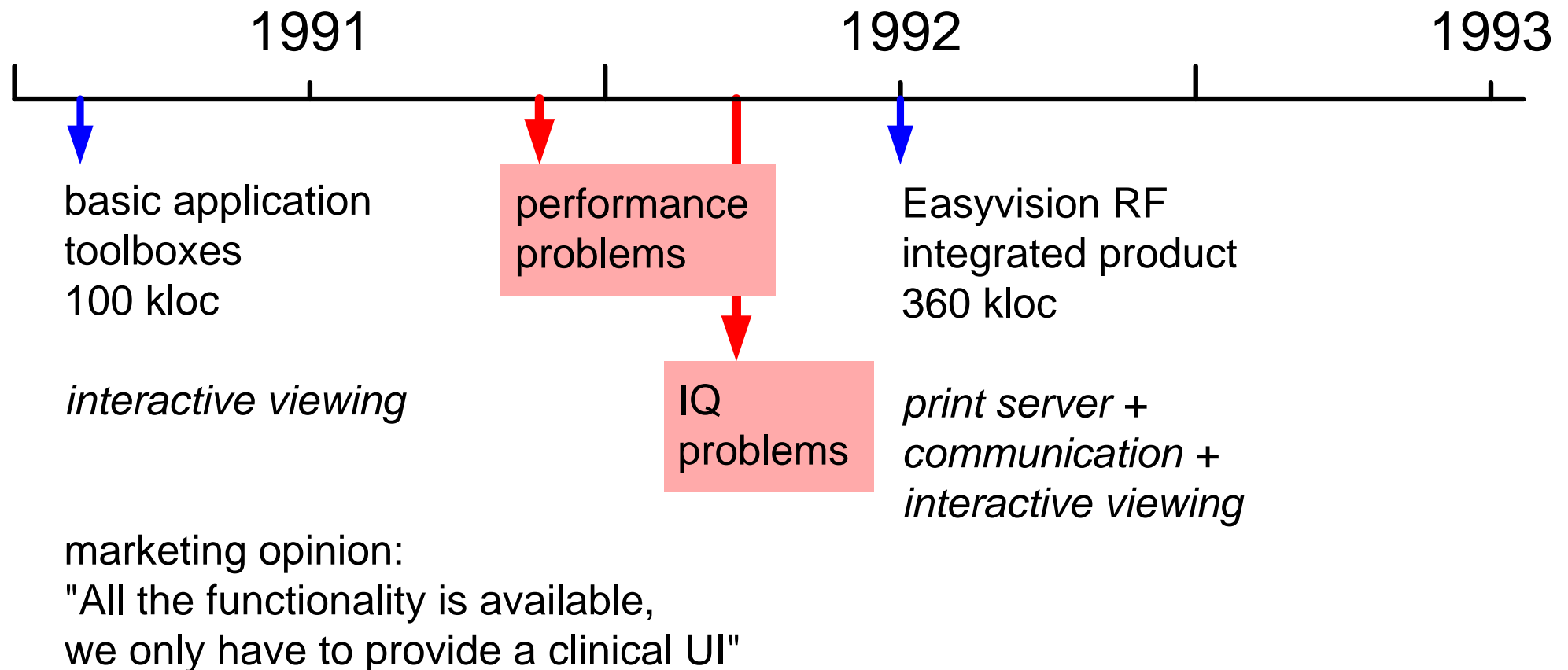
### Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

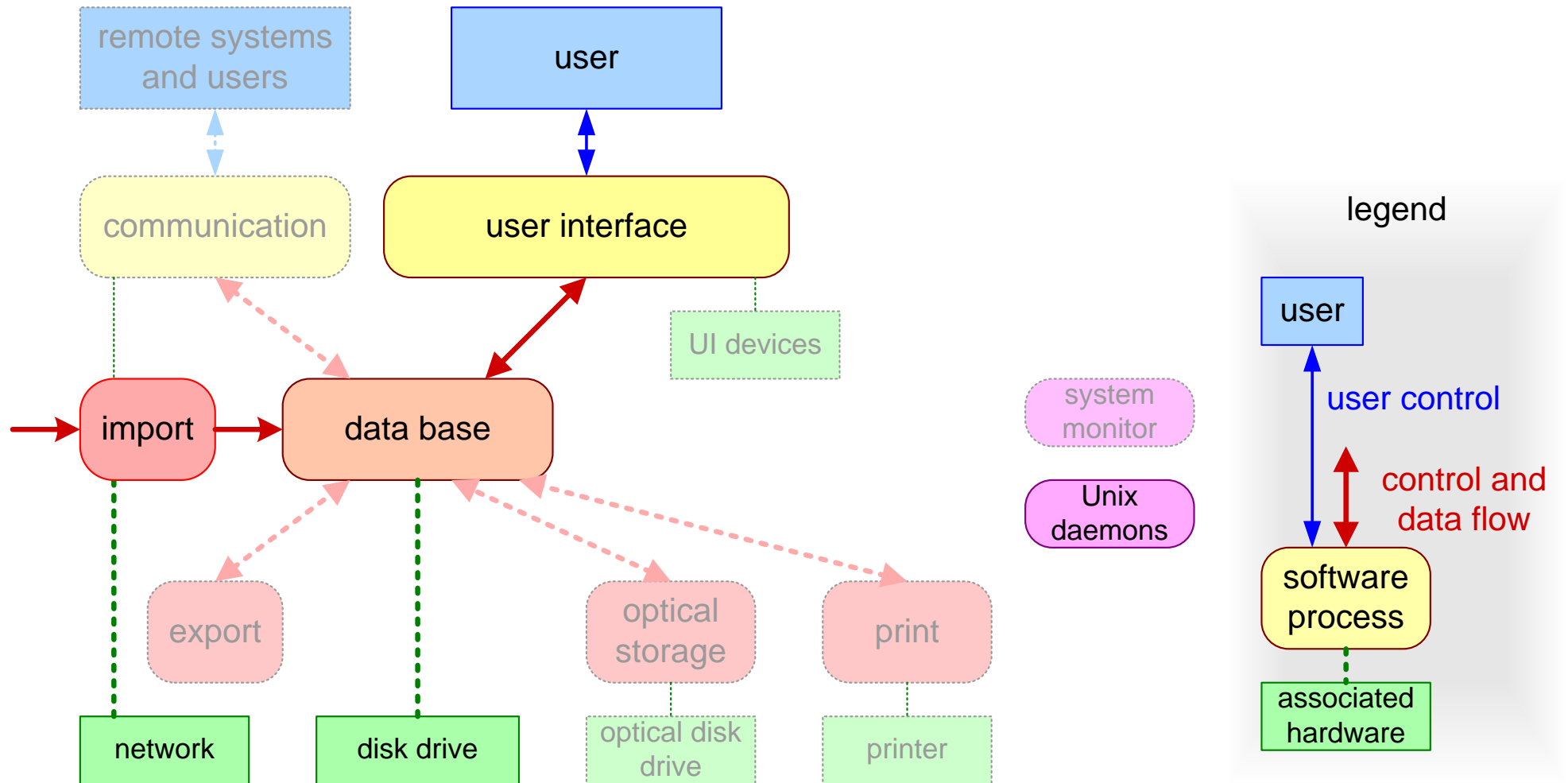
September 1, 2020  
status: finished  
version: 1.2



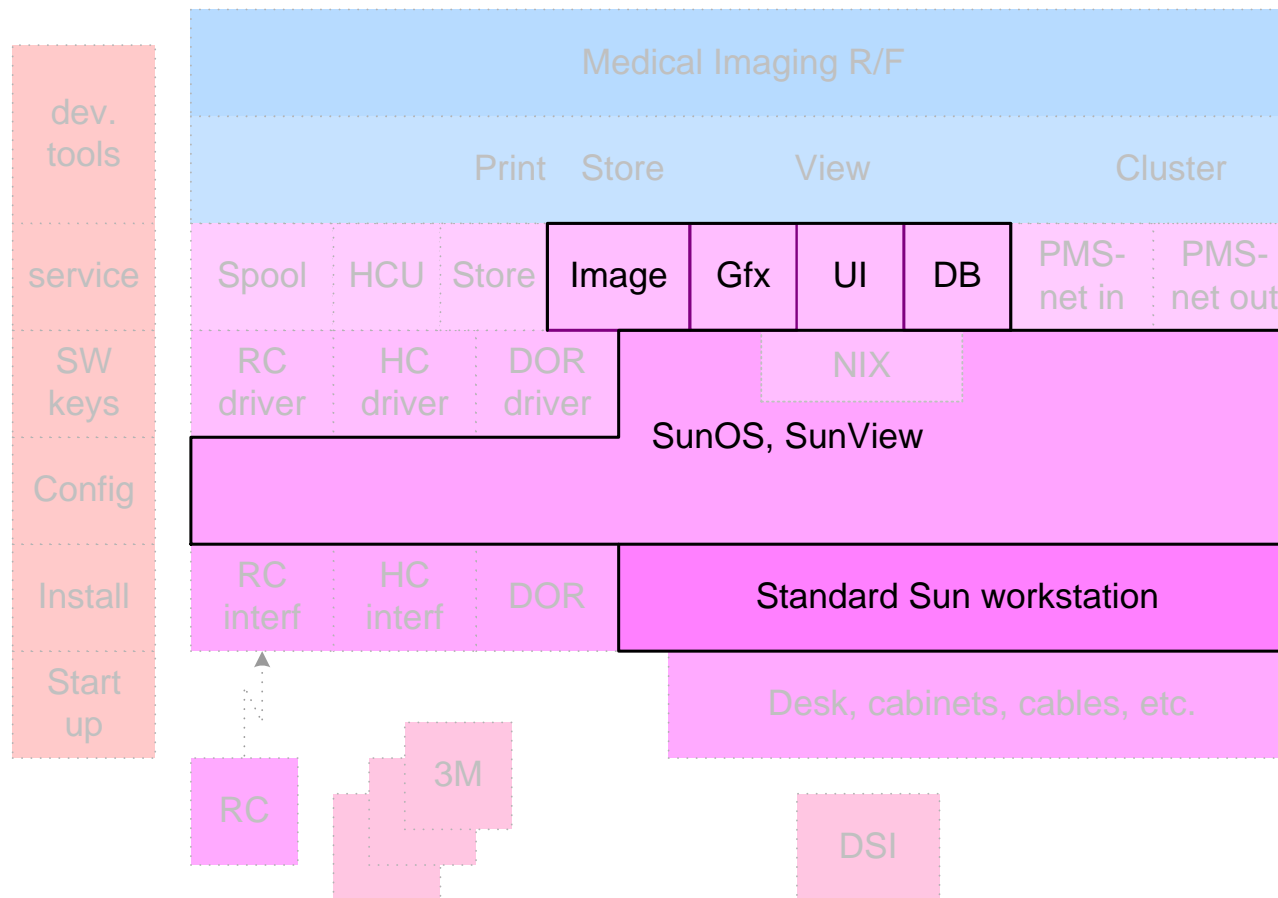
# Chronology of Easyvision RF R1 development



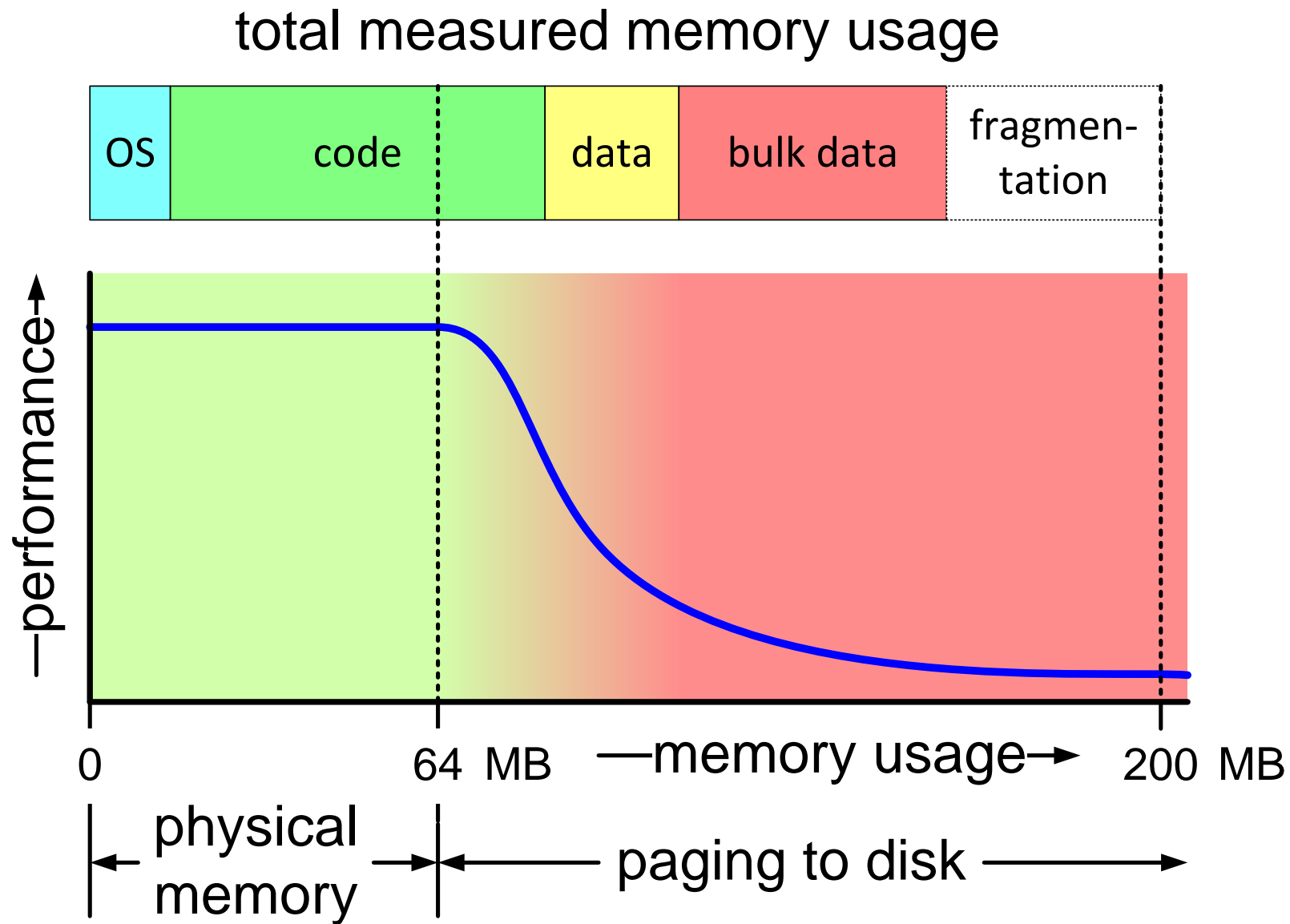
# SW Process structure 1991



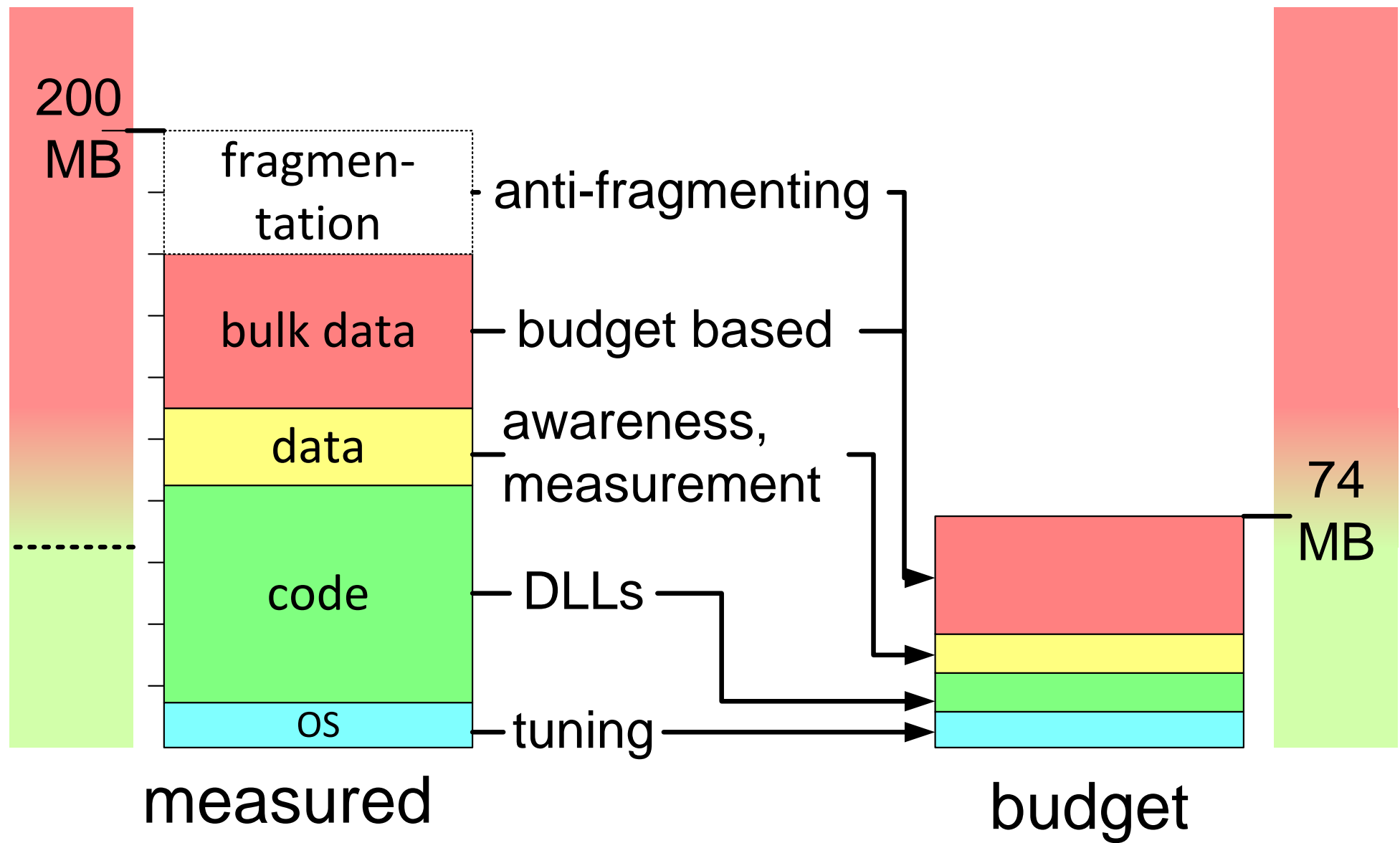
# SW layers 1991



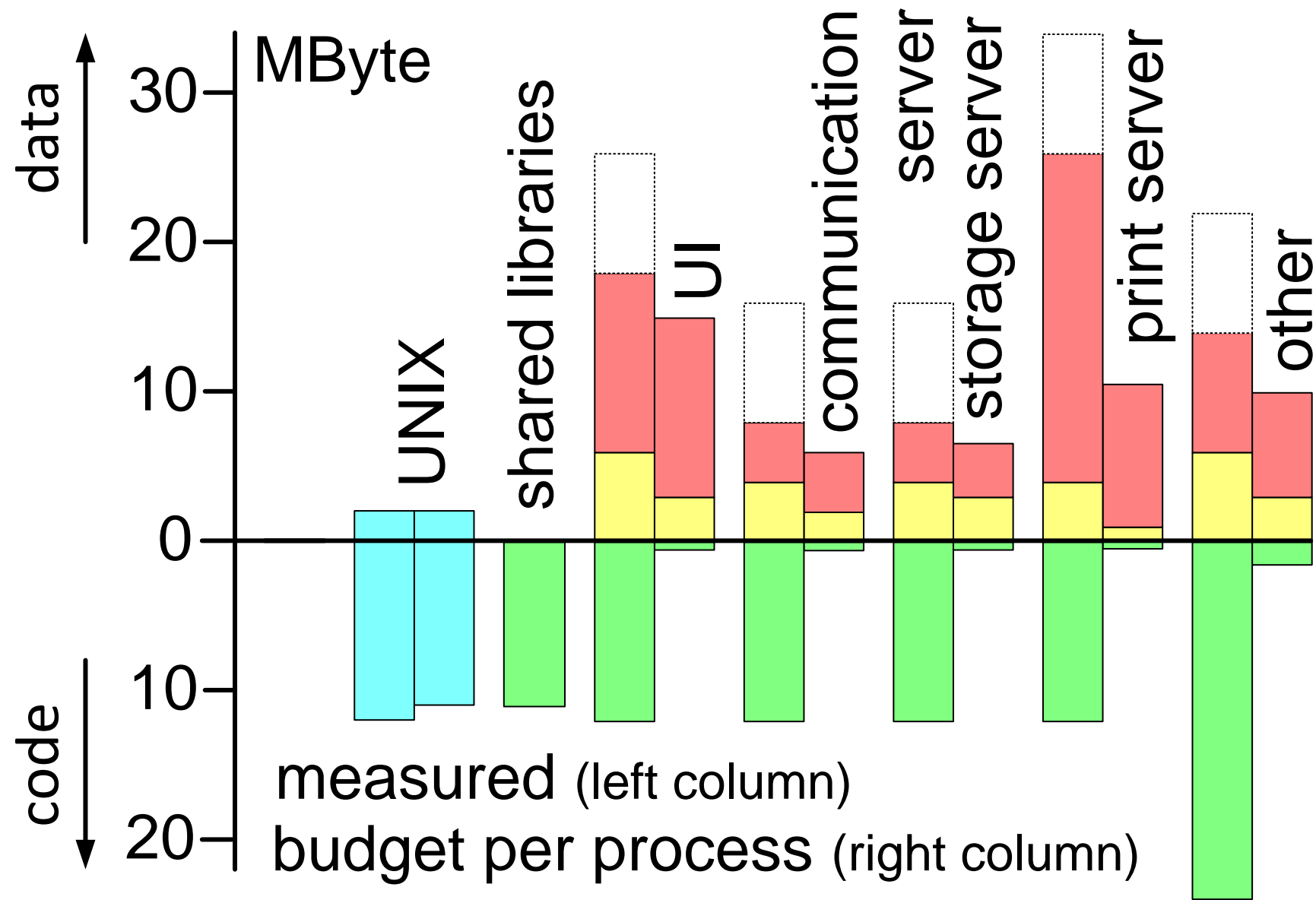
# Memory usage half way R1



# Solution of memory performance problem

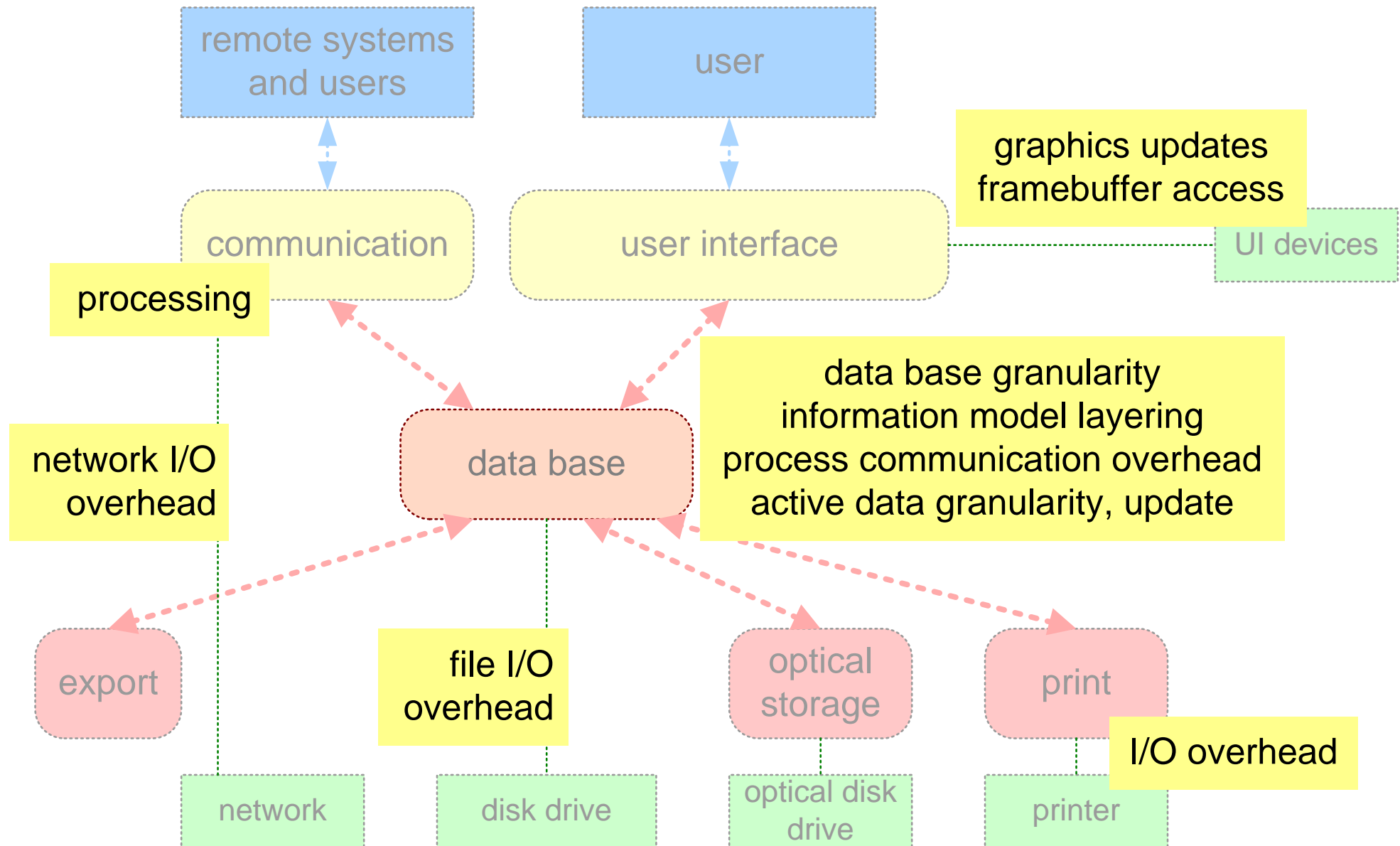


# Visualization memory use per process

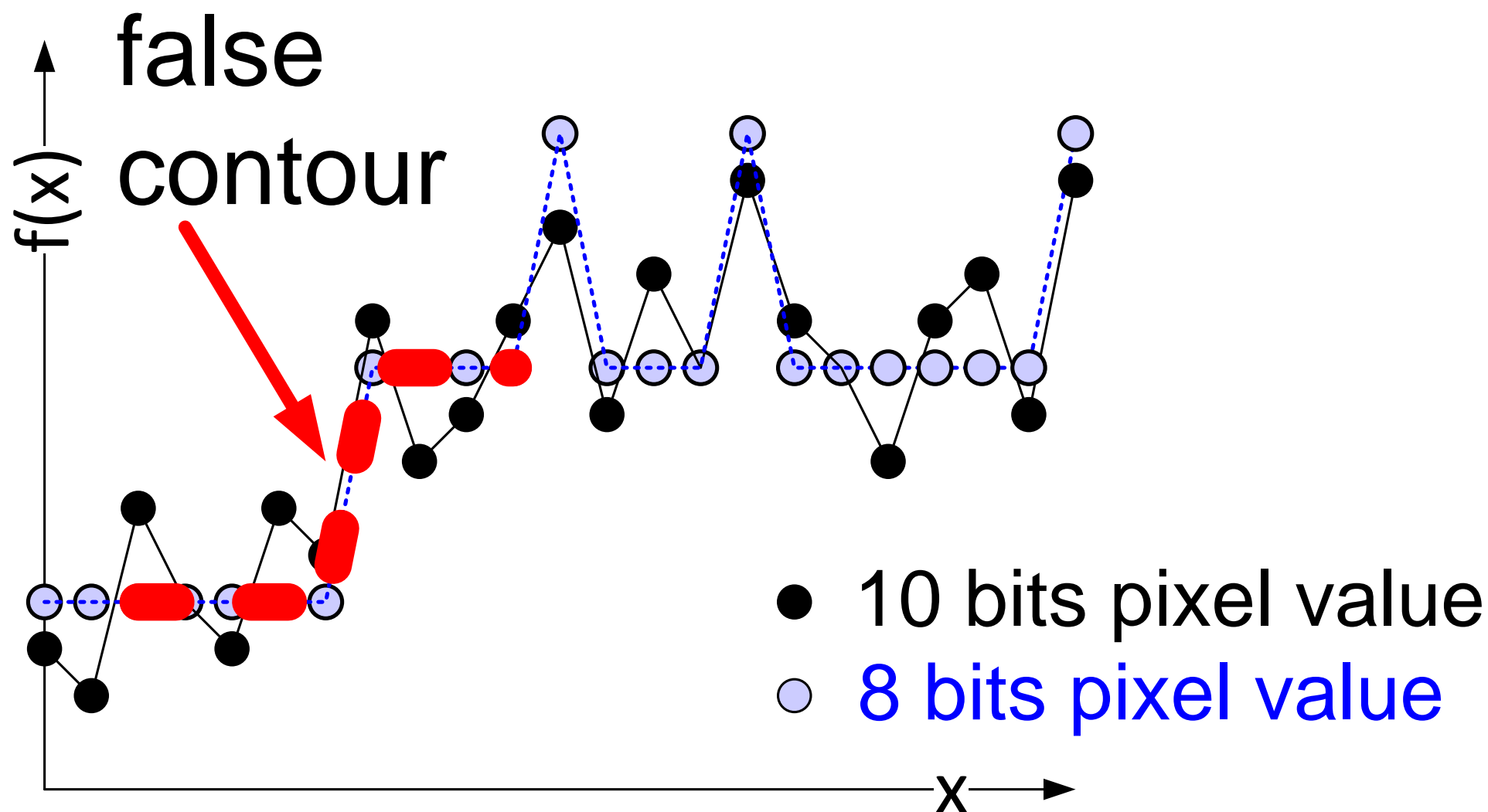




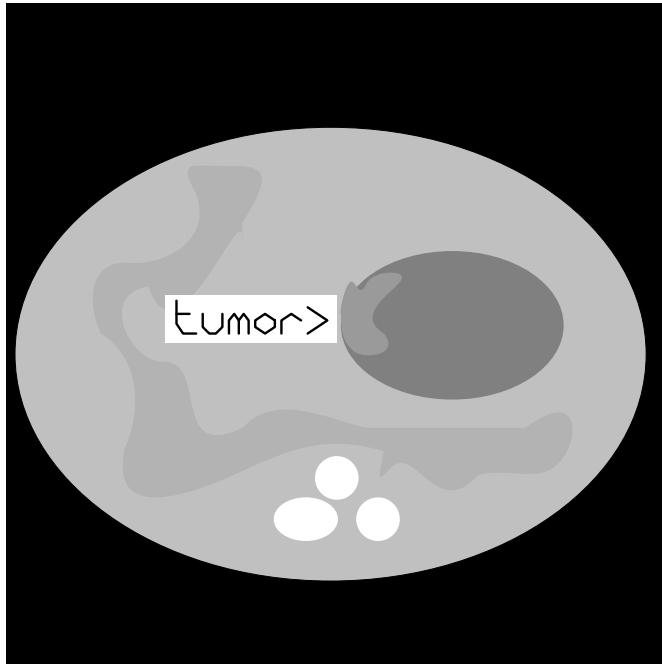
# Causes of performance problems other than memory use



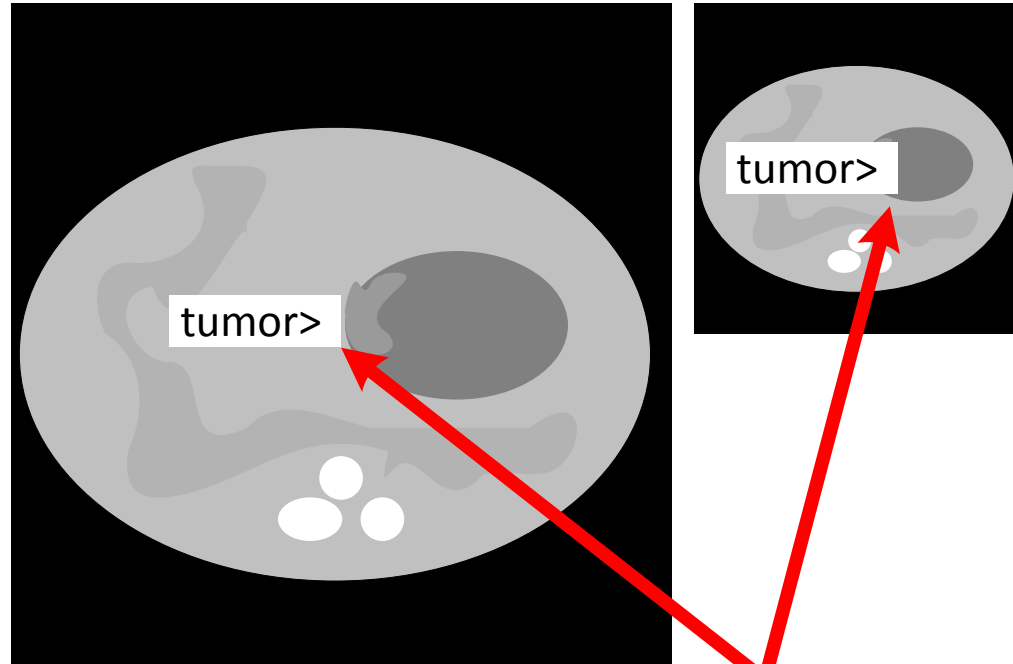
# Image quality and safety problem



# Safety problem



URF monitor output:  
fixed size letters at fixed grid



for user readability the font-size was  
determined "intelligently"; causing a dangerous  
mismatch between text and image

EV output: scaleable fonts in graphics overlay

# Medical Imaging Workstation: CAF Views

by *Gerrit Muller*      University of South-Eastern Norway-NISE

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

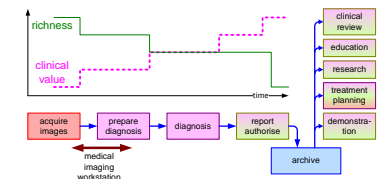
## Abstract

The Customer objectives, Application and Functional views are described. The radiology department and the radiologist are the main customer. The clinical and the financial context of the radiology department is shown. The medical imaging workstation is positioned in the field of IT products and in the clinical workflow. The market segmentation is shown. The typical URF examination is explained. Key drivers are linked to application drivers and to product requirements. The functionality development over time is shown and the role of the information model for interoperability is discussed.

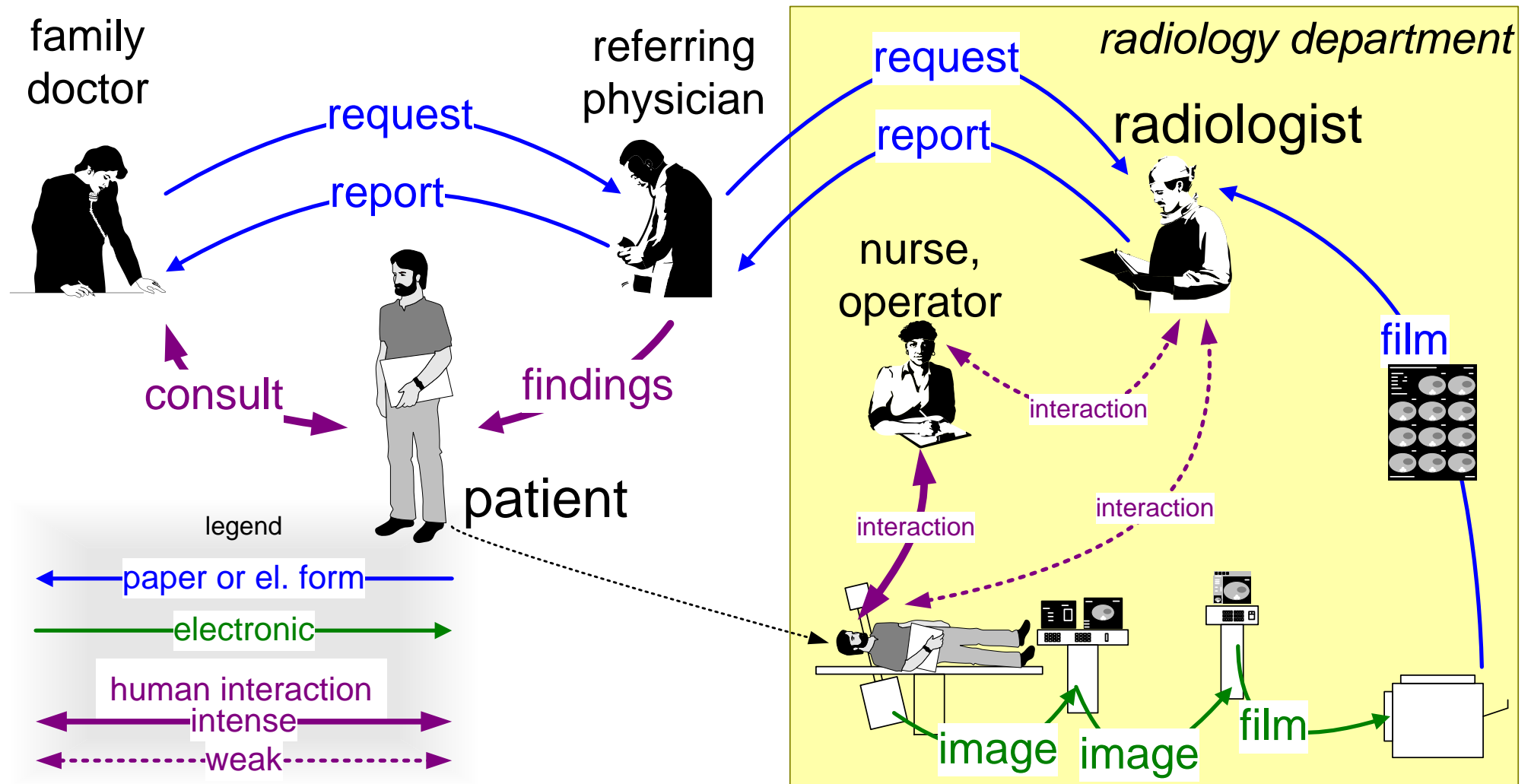
## Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

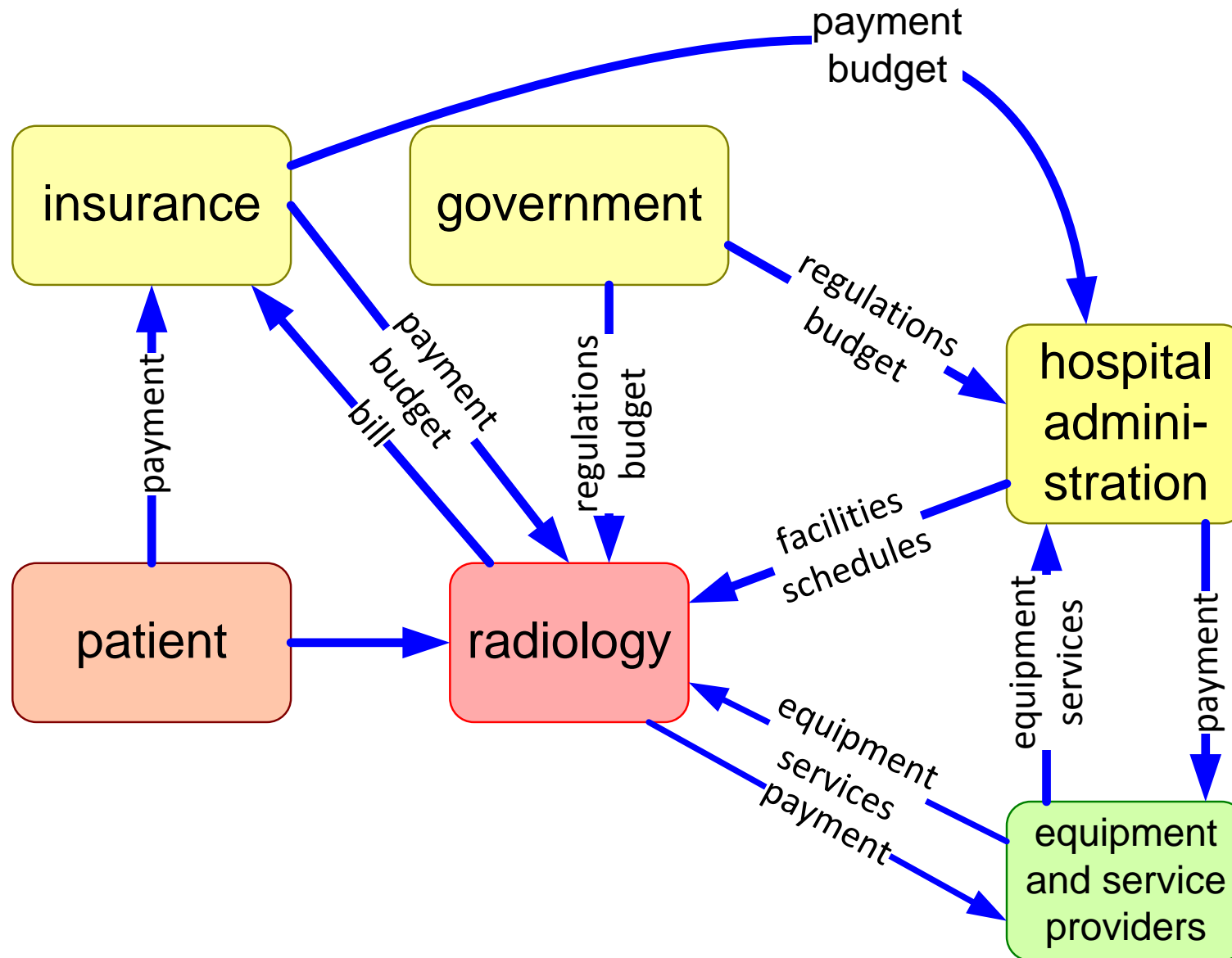
September 1, 2020  
status: finished  
version: 1.4



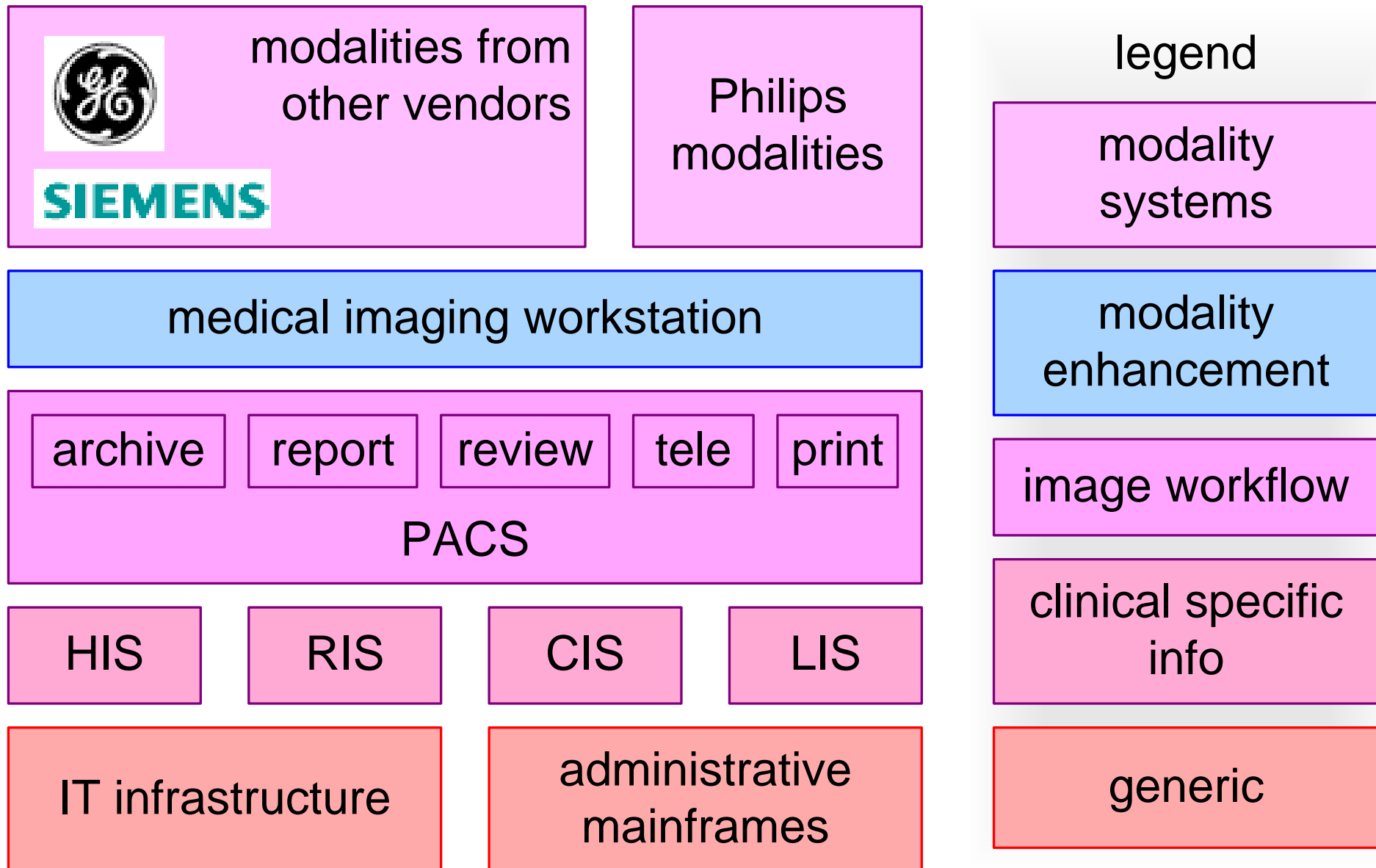
# The clinical context of the radiology department



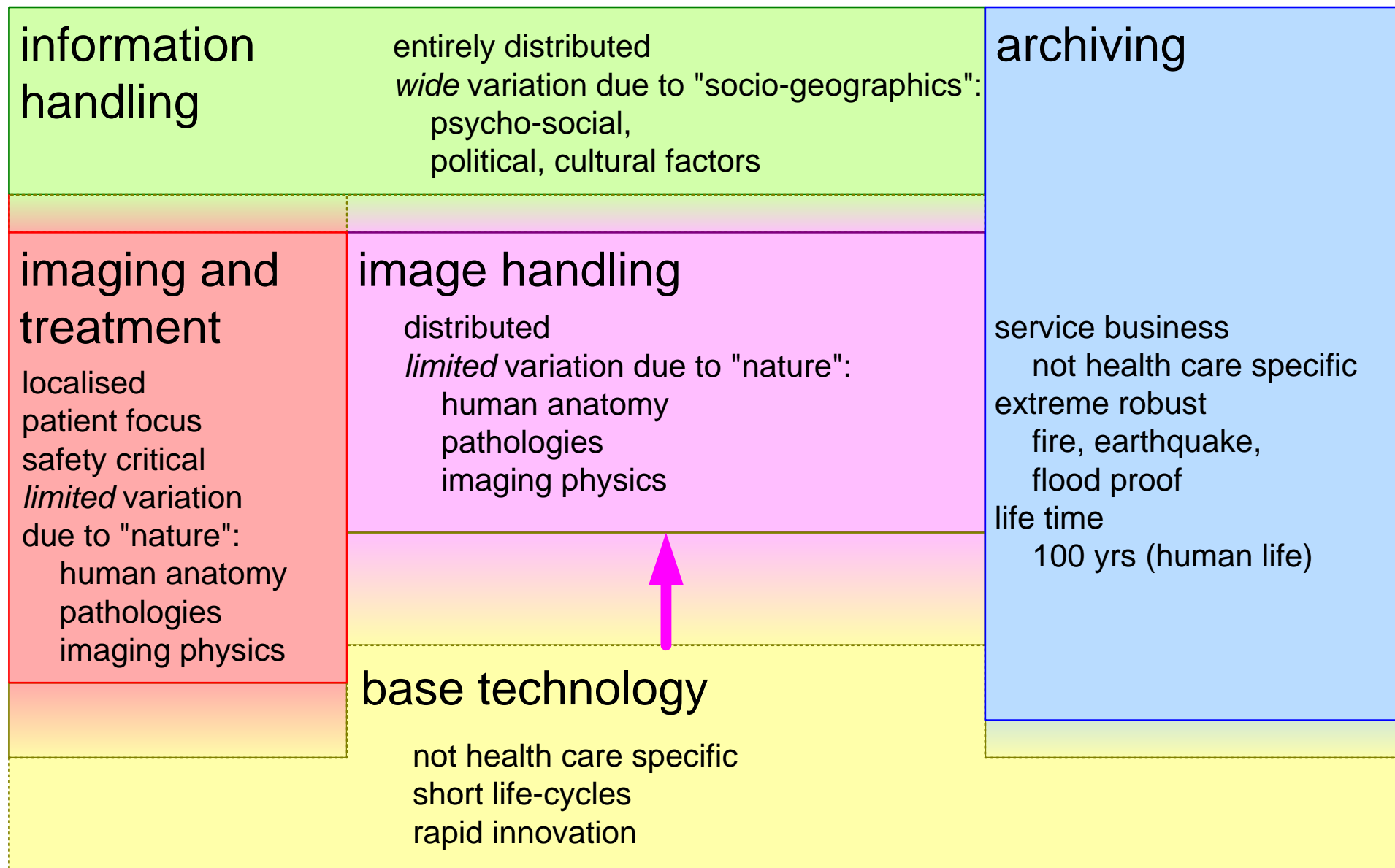
# The financial context of the radiology department



# Application layering of IT systems

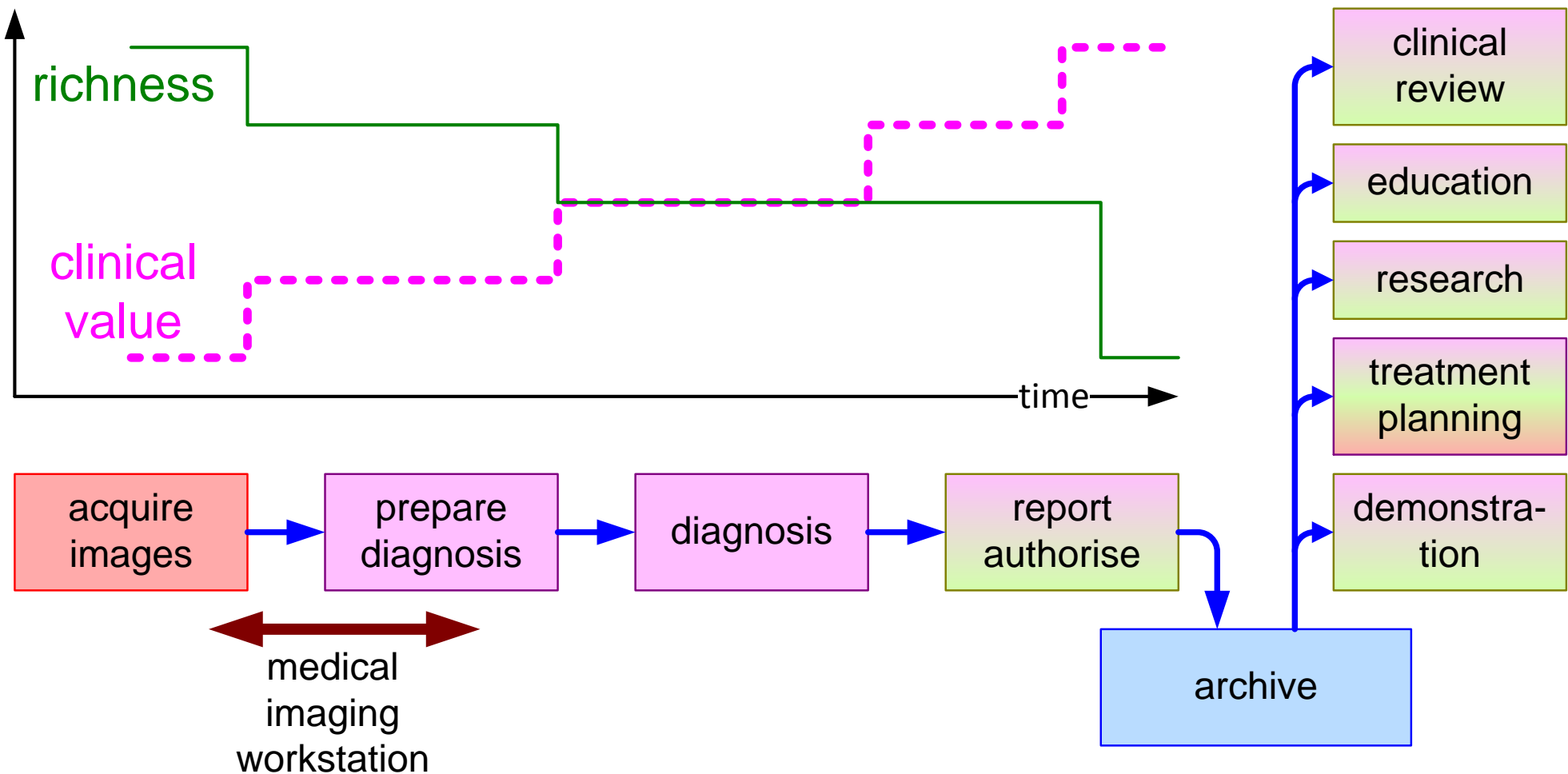


# Reference model for healthcare automation



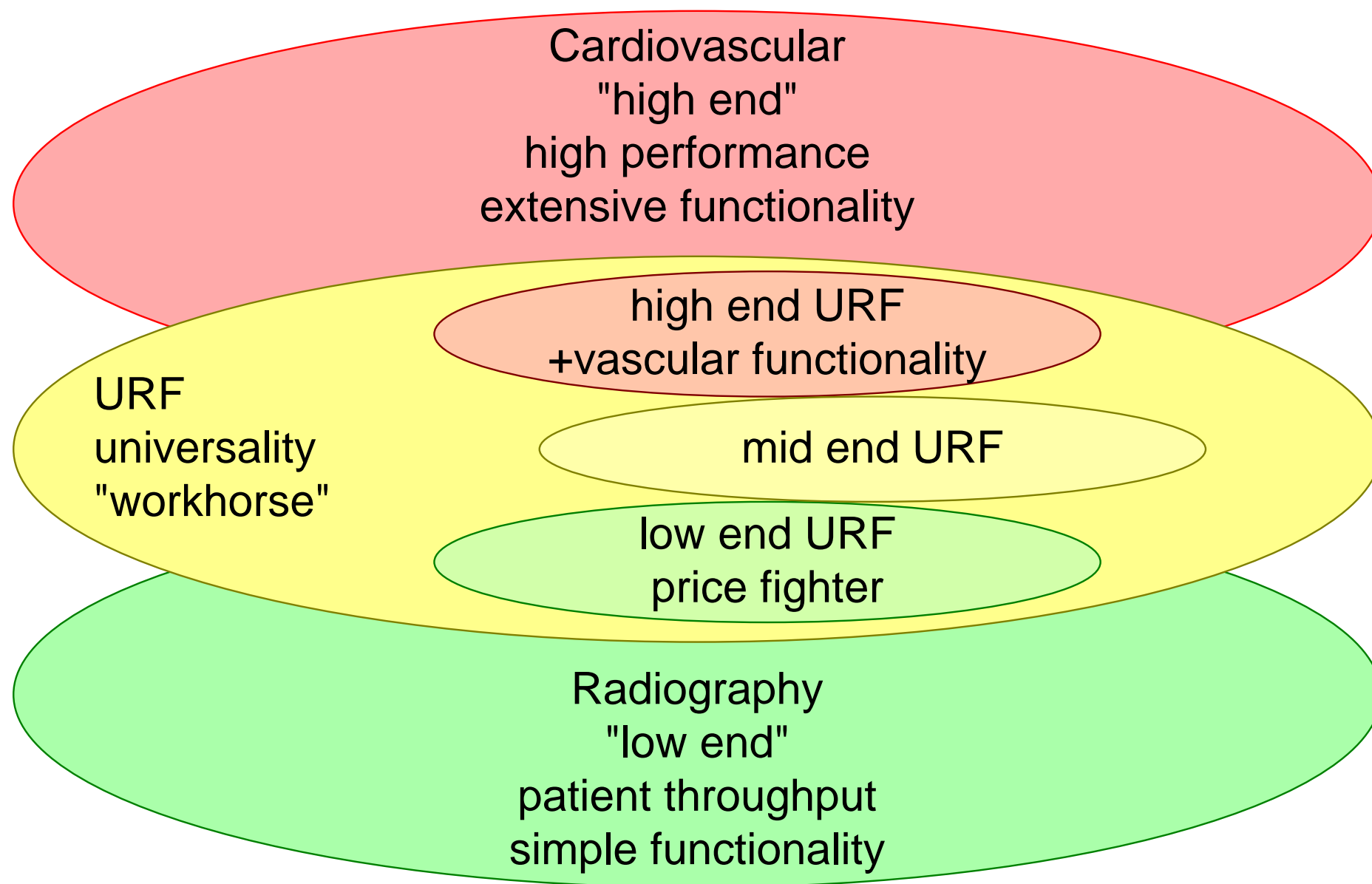


# Clinical information flow



# URF market segmentation

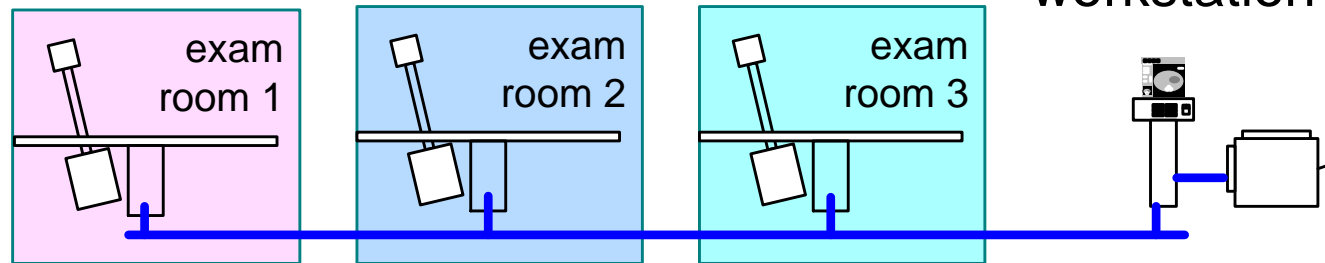
---



# Typical case URF examination

3 examination rooms connected to

1 medical imaging  
workstation + printer

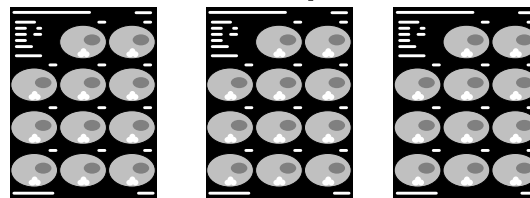


examination room: average 4 interleaved examinations / hour

image production: 20  $1024^2$  8 bit images per examination

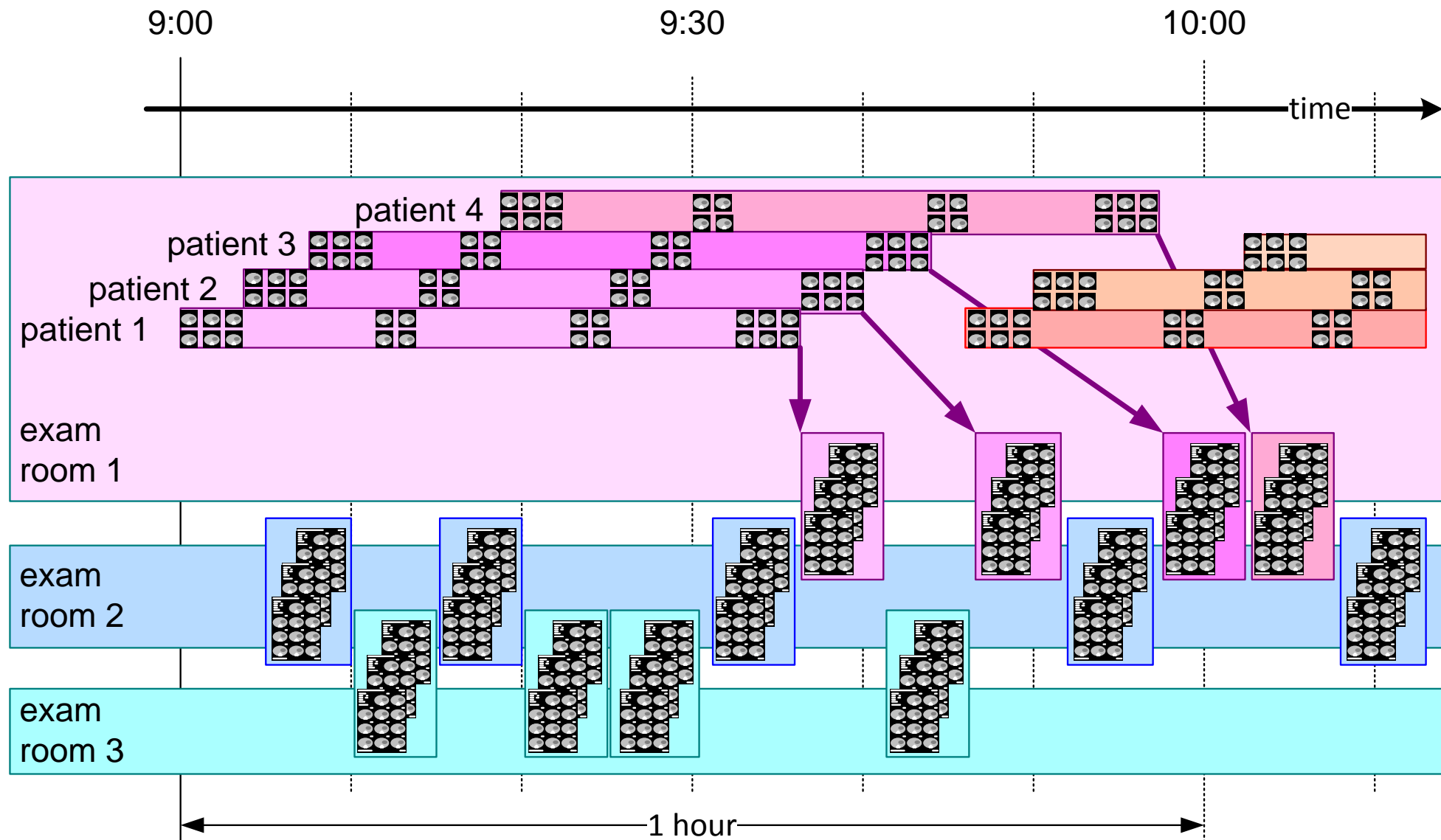


film production: 3 films of 4k\*5k pixels each

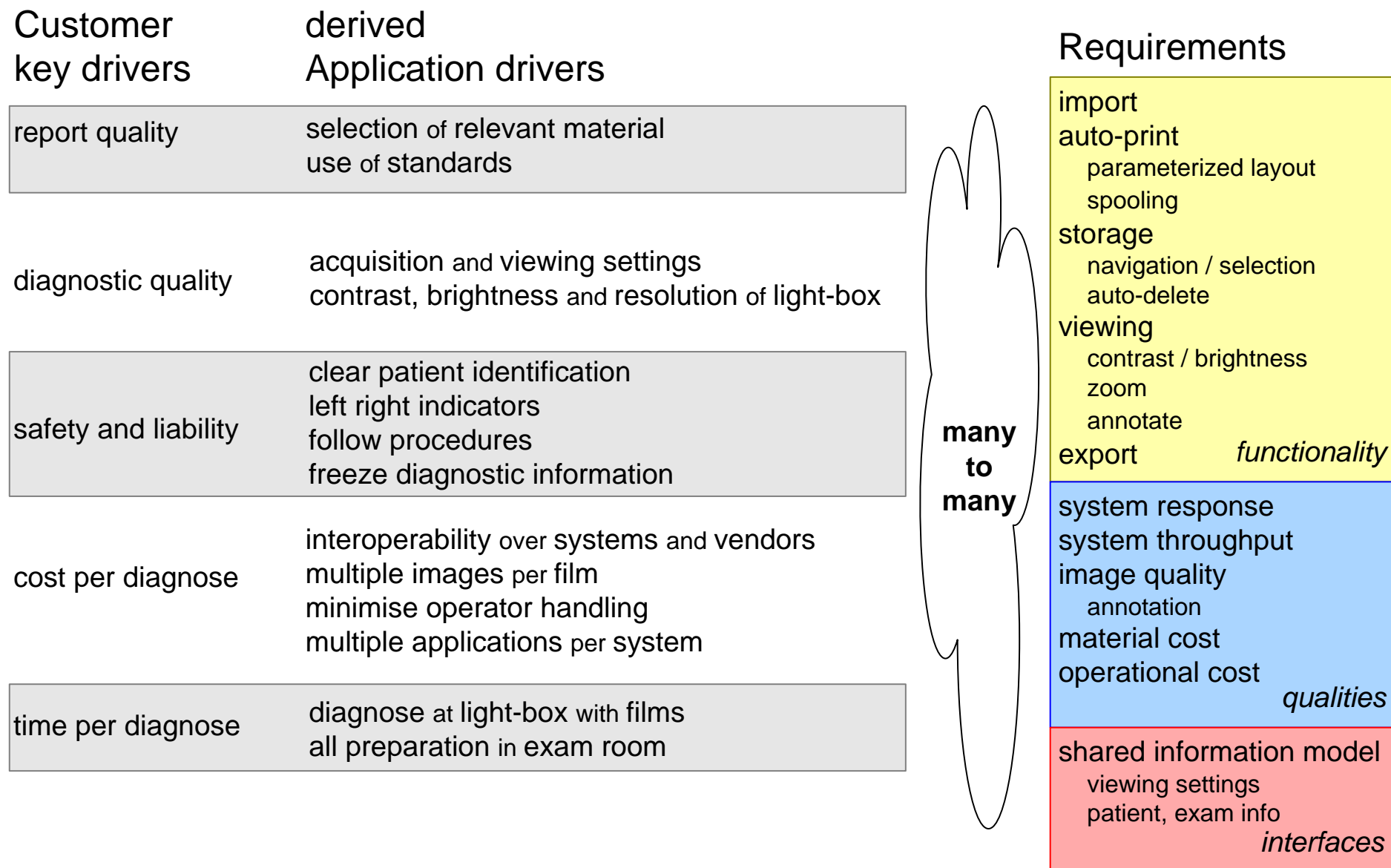


high quality output  
(bi-cubic interpolation)

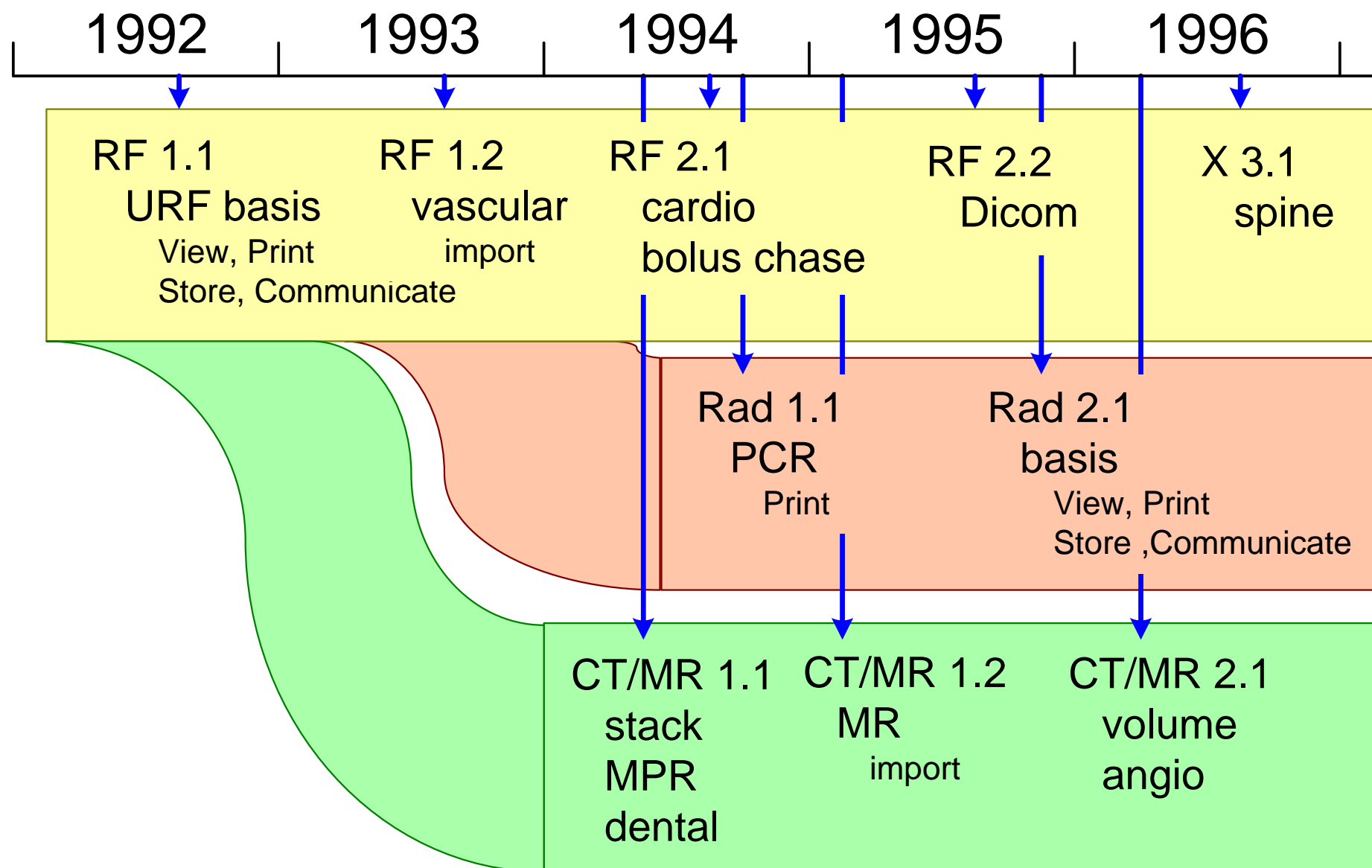
# Timing of typical URF examination rooms



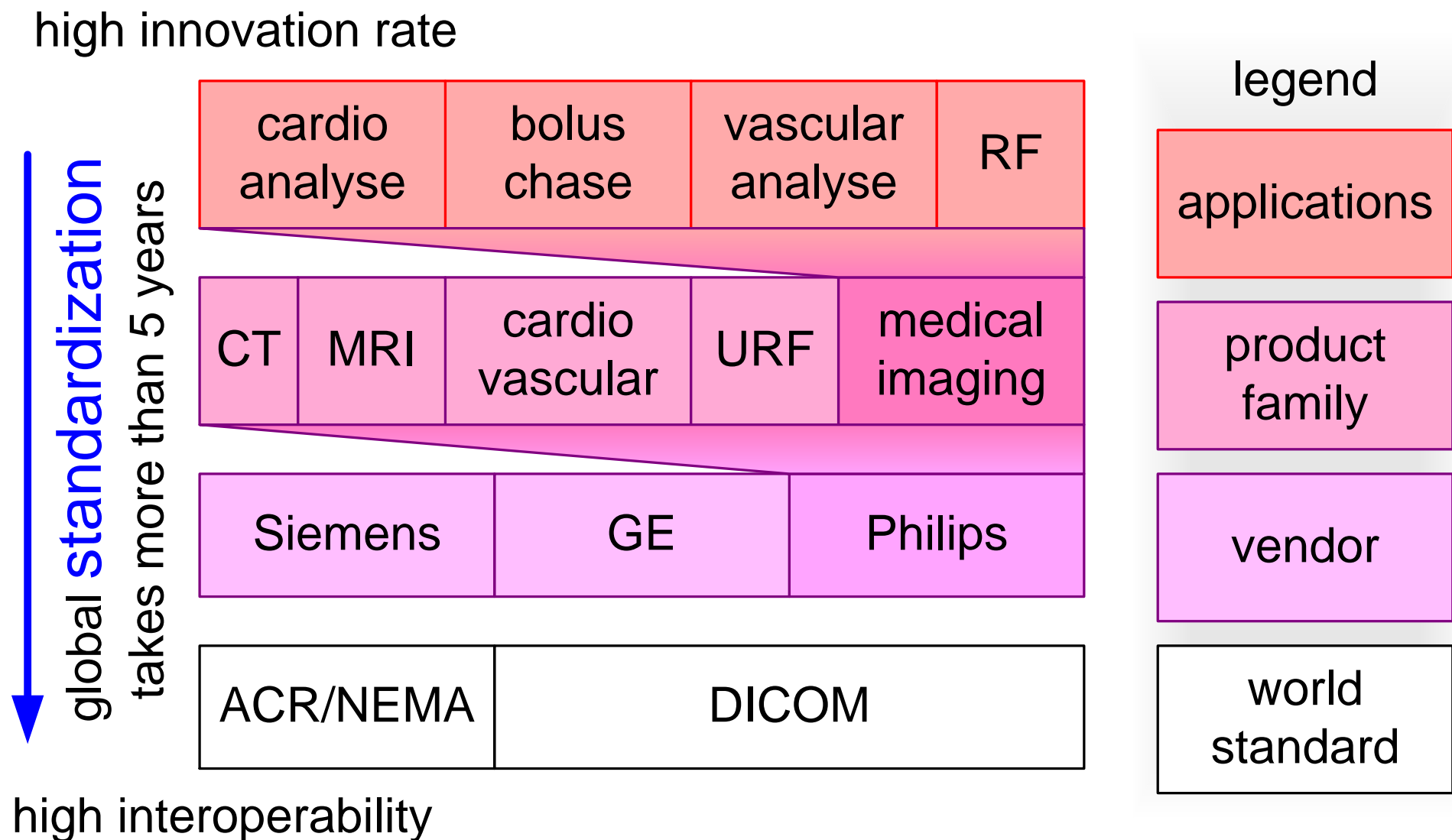
# Key drivers, application drivers and requirements



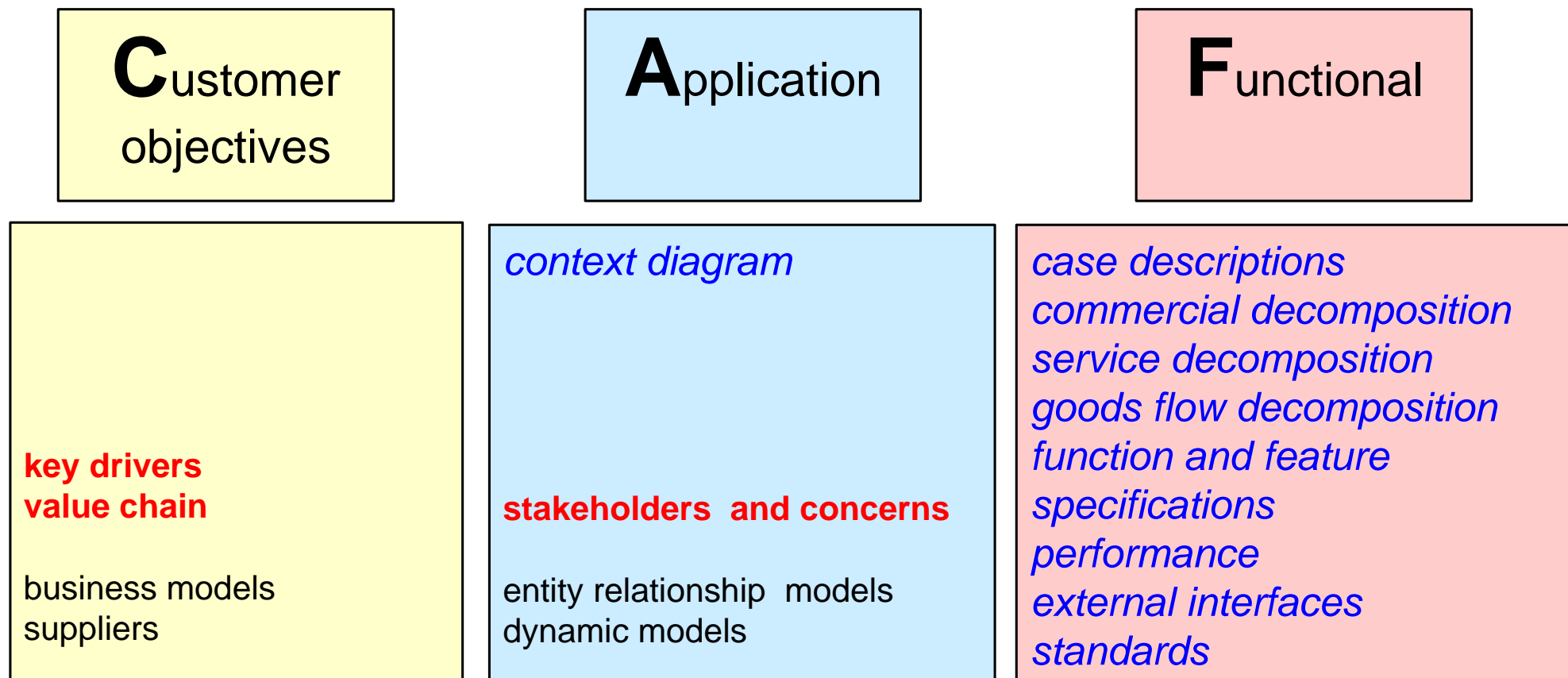
# Retrospective functionality roadmap



# Information model, standardization for interoperability



# Coverage of submethods of the CAF views



legend

*explicitly addressed*

*addressed only implicitly*

not addressed

coverage based on documentation status of first product release



# Medical Imaging Workstation: CR Views

by *Gerrit Muller*      University of South-Eastern Norway-NISE

e-mail: [gaudisite@gmail.com](mailto:gaudisite@gmail.com)

[www.gaudisite.nl](http://www.gaudisite.nl)

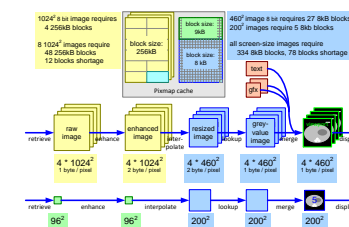
## Abstract

The concepts and realization of the medical imaging workstation are described. The following concepts are described: presentation and processing pipeline, resource management (CPU and memory), including caching and anti-fragmentation strategy, software process decomposition and decomposition rules. The actual realization figures serve as illustration for the justification of some of the concepts.

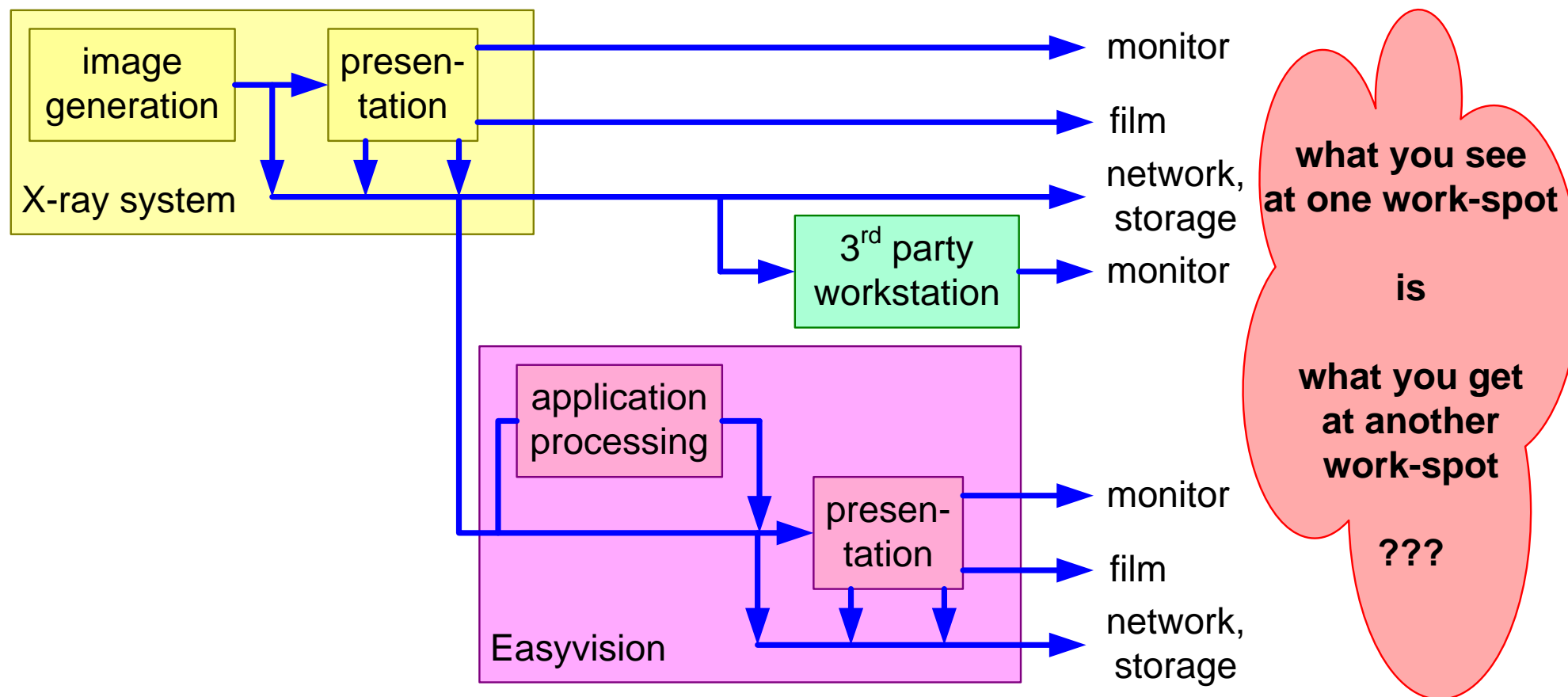
## Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

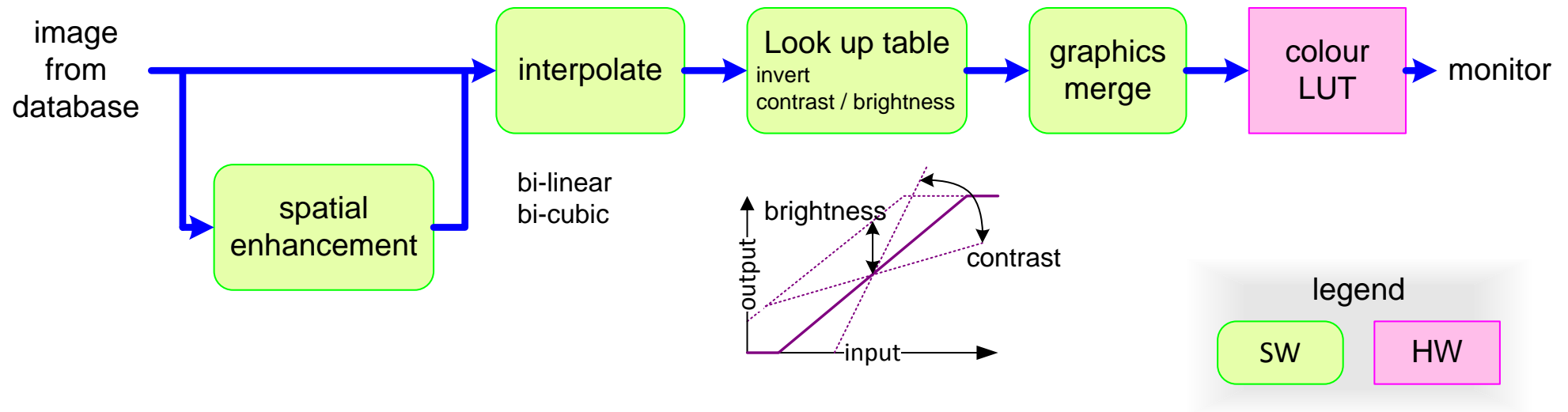
September 1, 2020  
status: finished  
version: 2.7



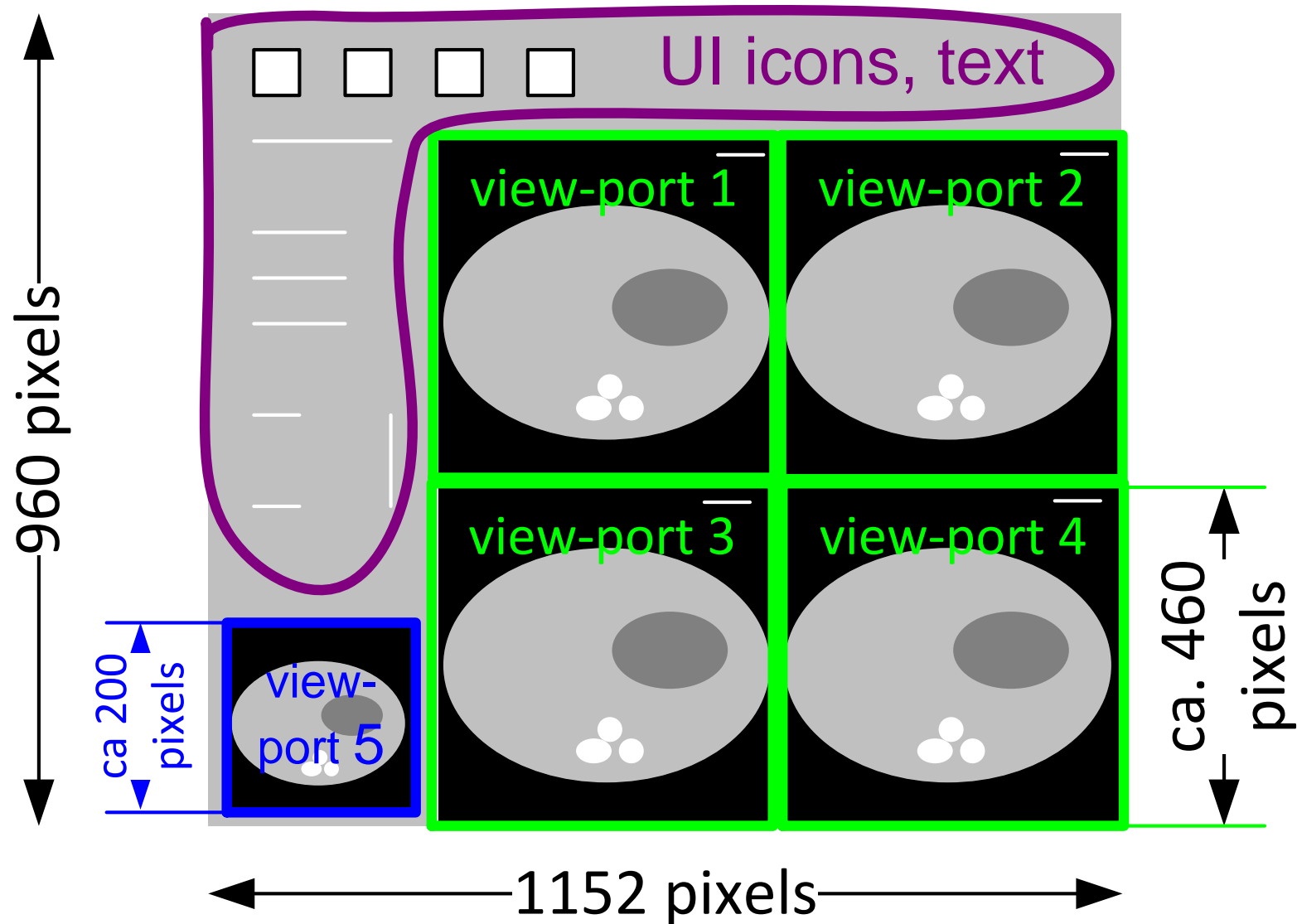
# Image Quality expectation WYSIWYG



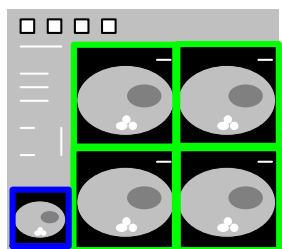
# Presentation pipeline for X-ray images



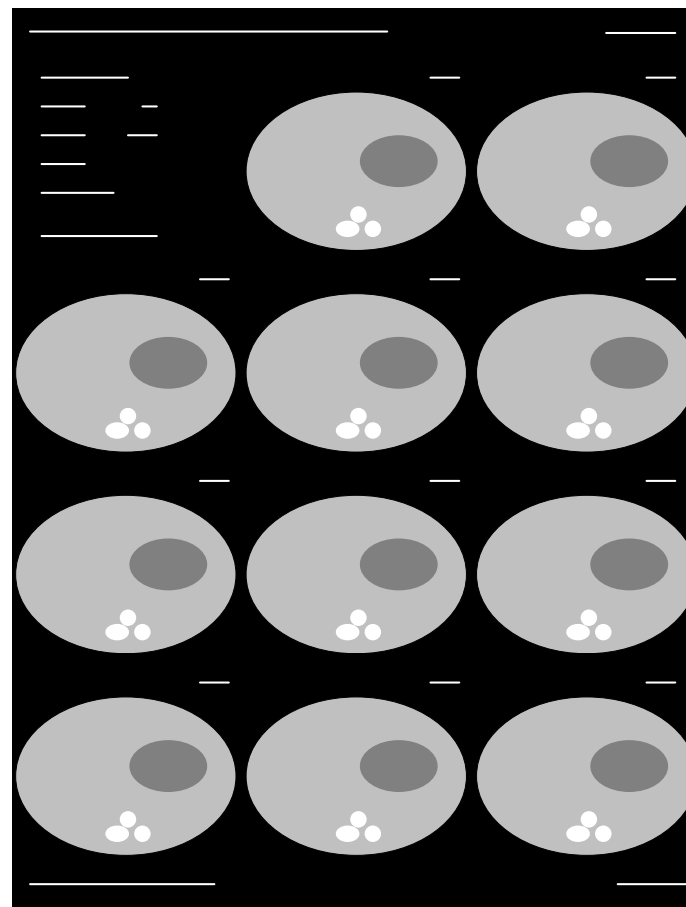
# Quadruple view-port screen layout



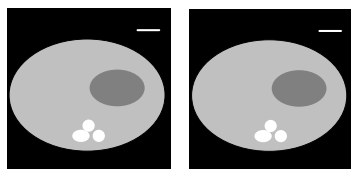
# Rendered images at different destinations



*Screen:*  
low resolution  
fast response

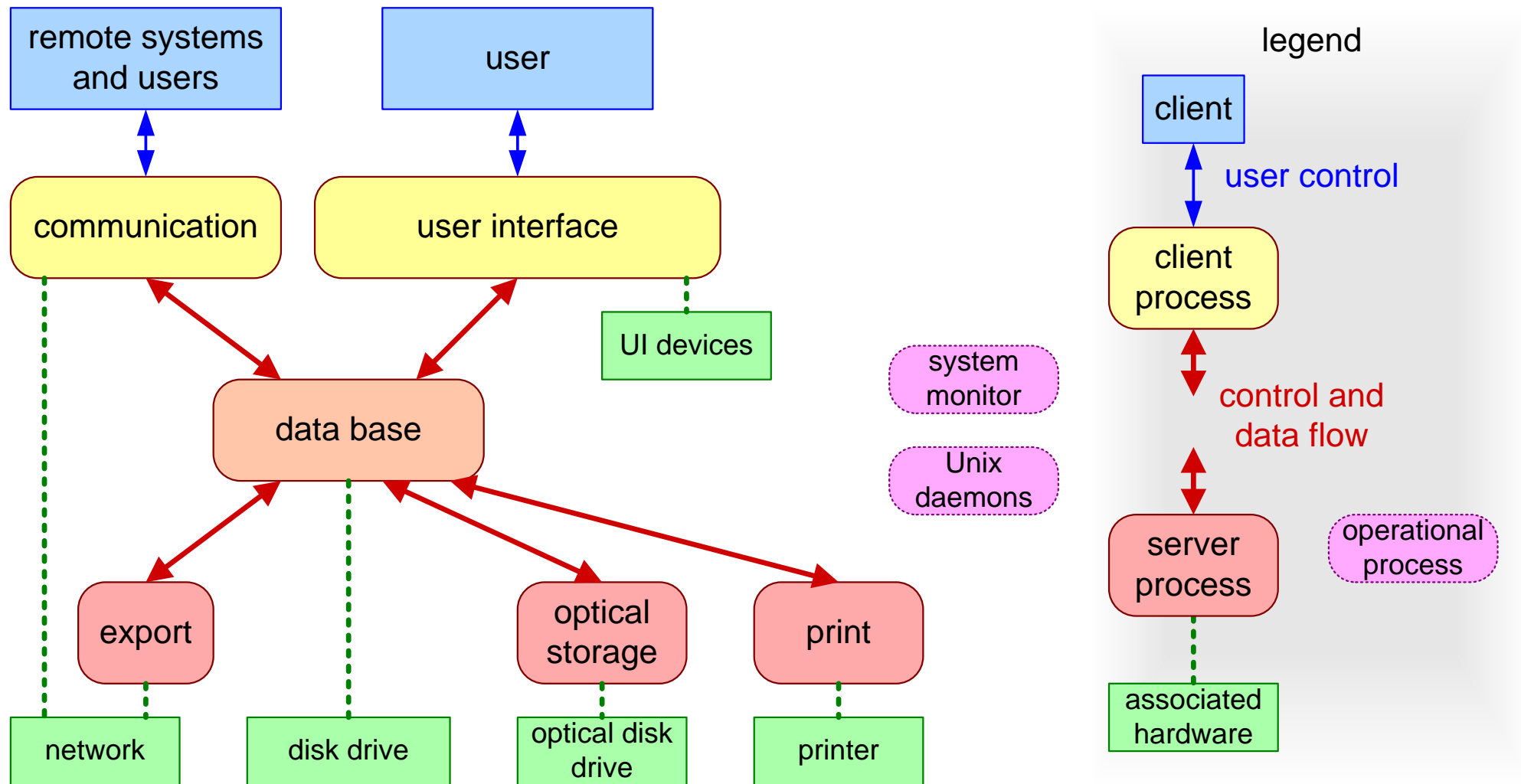


*Film:*  
high resolution  
high throughput



*Network:*  
medium resolution  
high throughput

# Concurrency via software processes

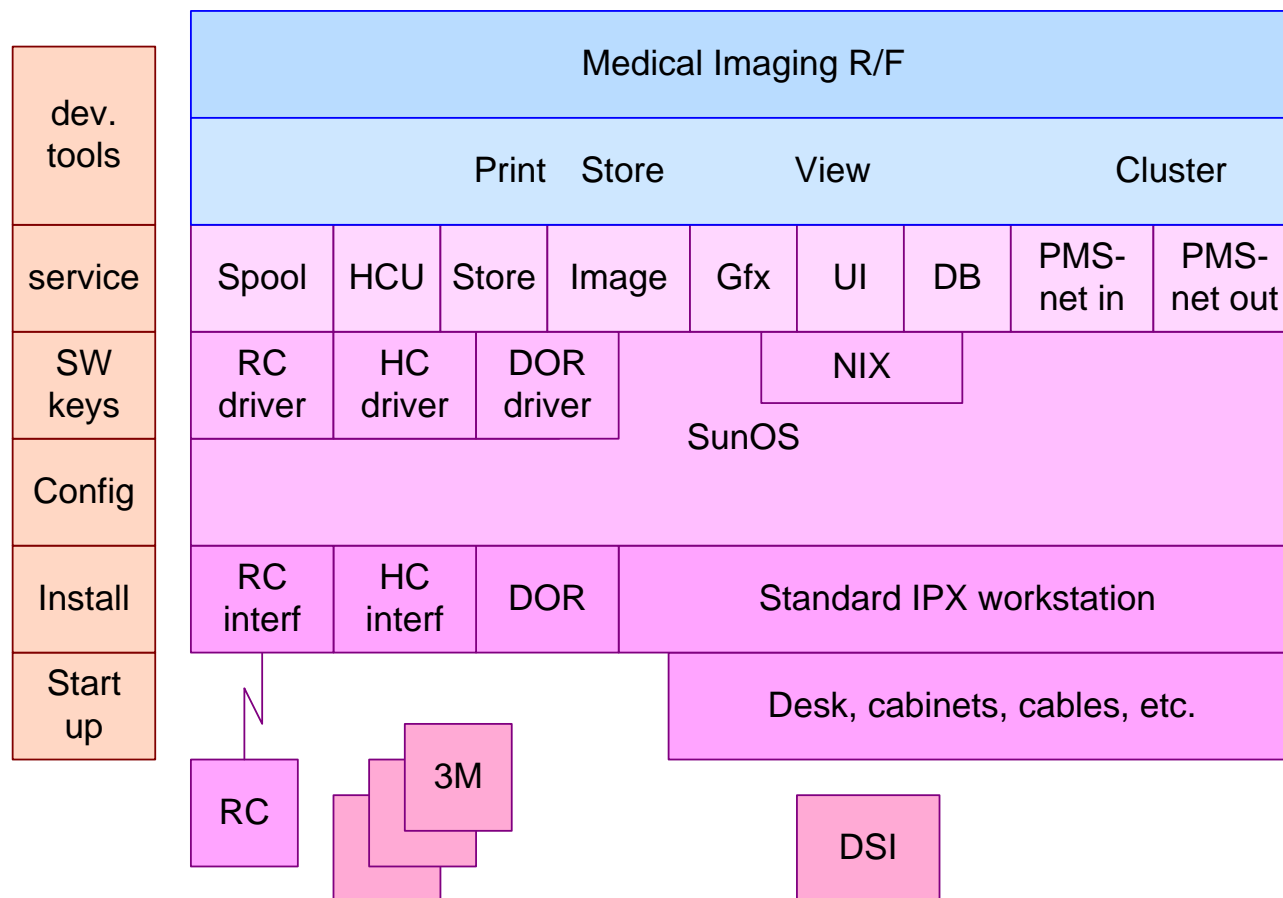


# Criteria for process decomposition

---

- management of concurrency
- management of shared devices
- unit of memory budget (easy measurement)
- enables distribution over multiple processors
- unit of exception handling: fault containment and watchdog monitor

# Simplified layering of the software

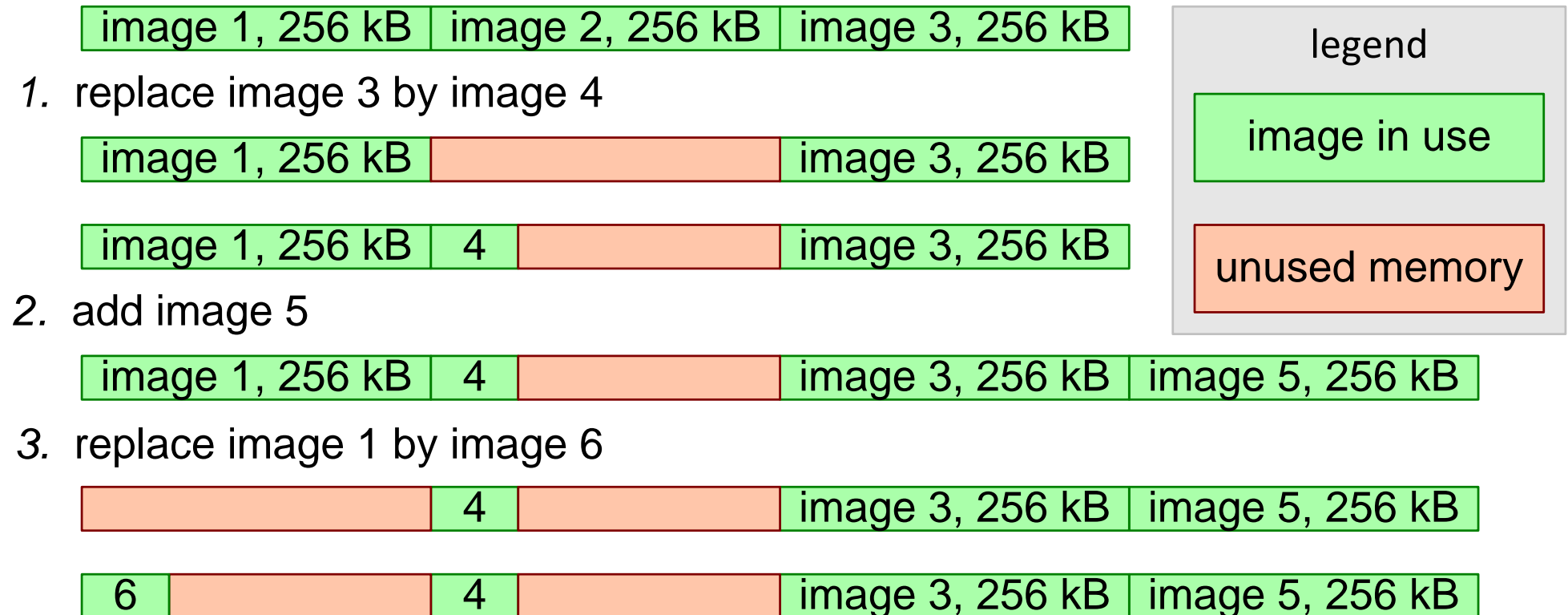




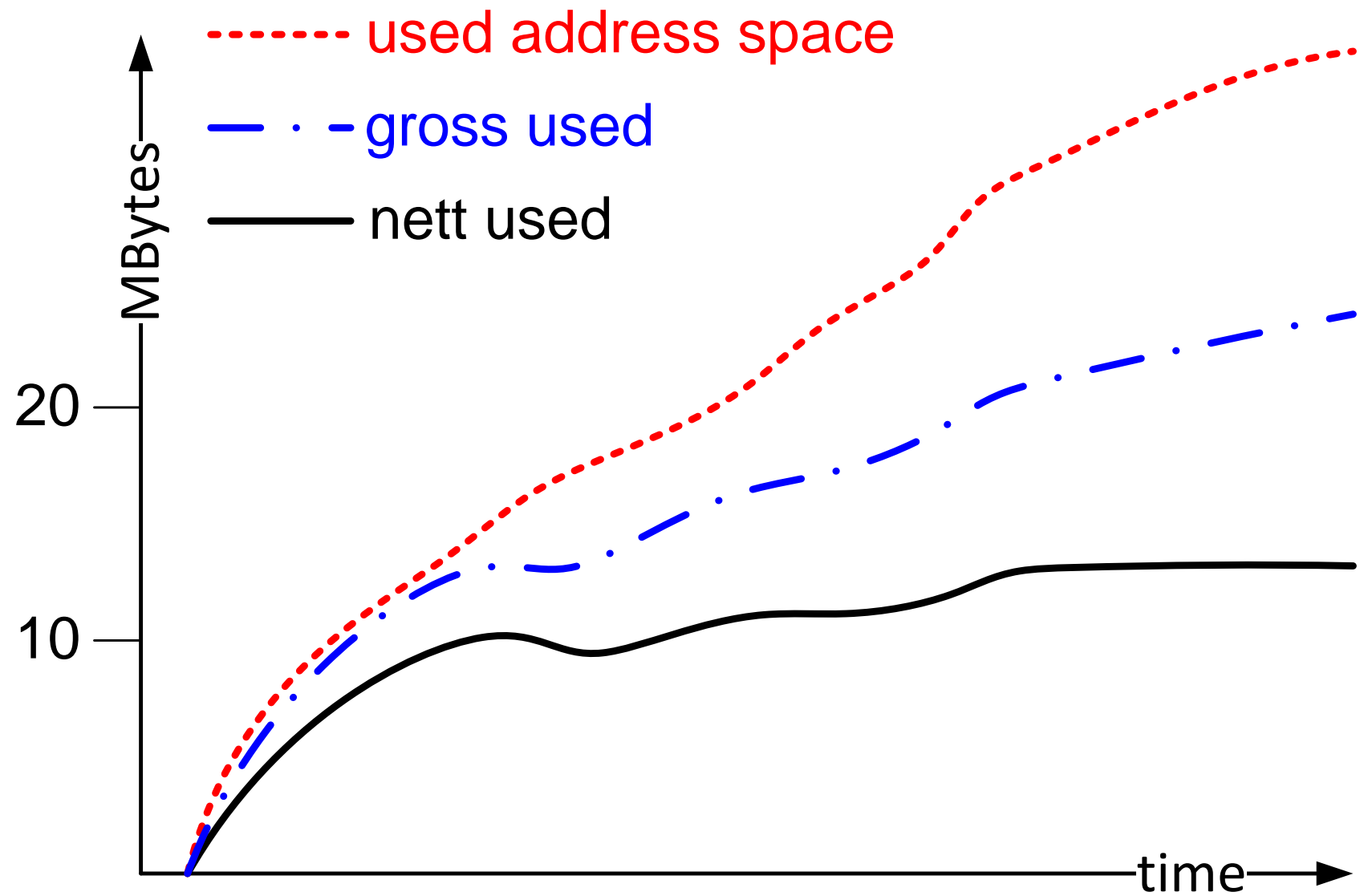
# Memory budget of Easyvision RF R1 and R2

	code		object data		bulk data		total	
<i>memory budget in Mbytes</i>	R1	R2	R1	R2	R1	R2	R1	R2
shared code	6.0	11.0					6.0	11.0
UI process	0.2	0.3	2.0	3.0	12.0	12.0	14.2	15.3
database server	0.2	0.3	4.2	3.2		3.0	4.4	6.5
print server	0.4	0.3	2.2	1.2	7.0	9.0	9.6	10.5
DOR server	0.4	0.3	4.2	2.0	2.0	1.0	6.6	3.3
communication server	1.2	0.3	15.4	2.0	10.0	4.0	26.6	6.3
UNIX commands	0.2	0.3	0.5	0.2			0.7	0.5
compute server		0.3		0.5		6.0		6.8
system monitor		0.3		0.5				0.8
application total	8.6	13.4	28.5	12.6	31.0	35.0	66.1	61.0
UNIX							7.0	10.0
file cache							3.0	3.0
total							76.1	74.0

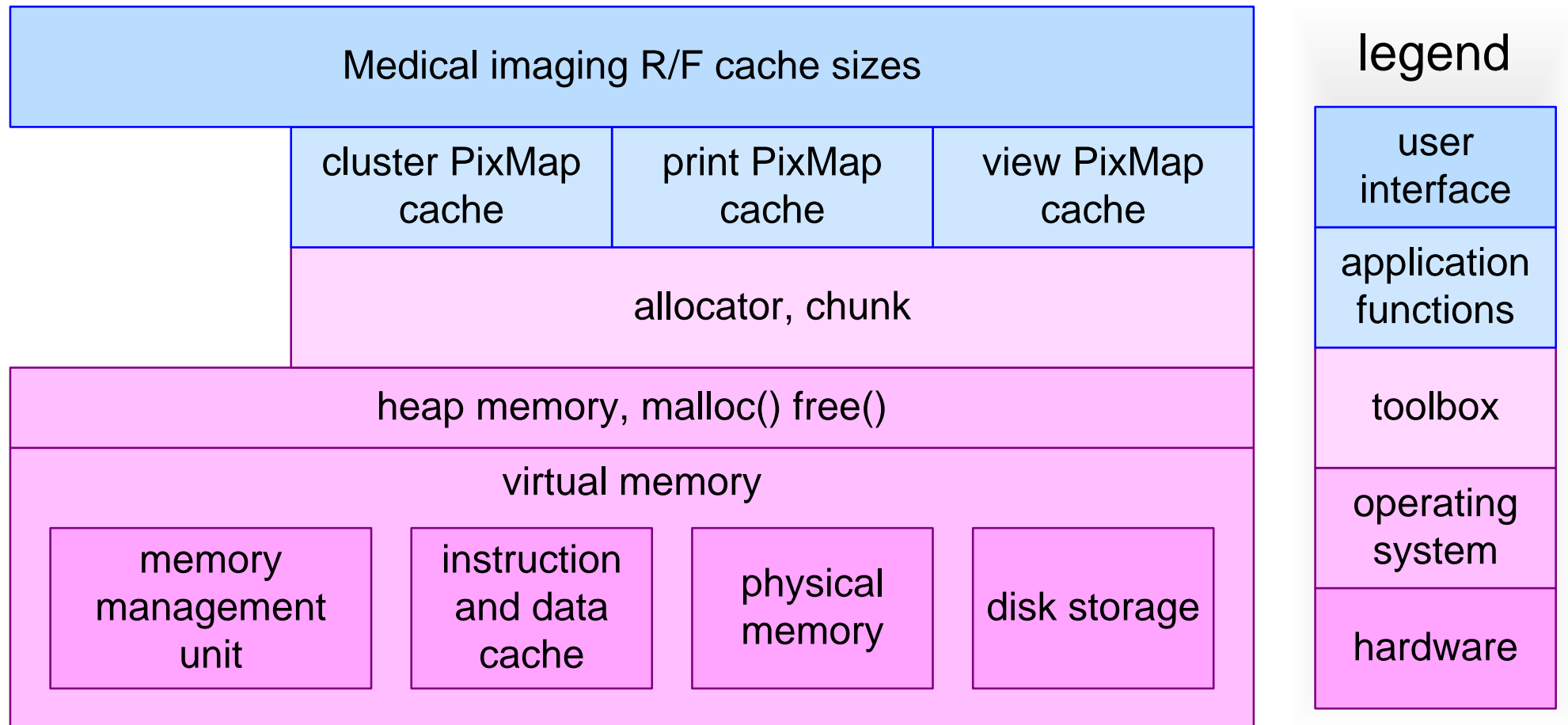
# Memory fragmentation



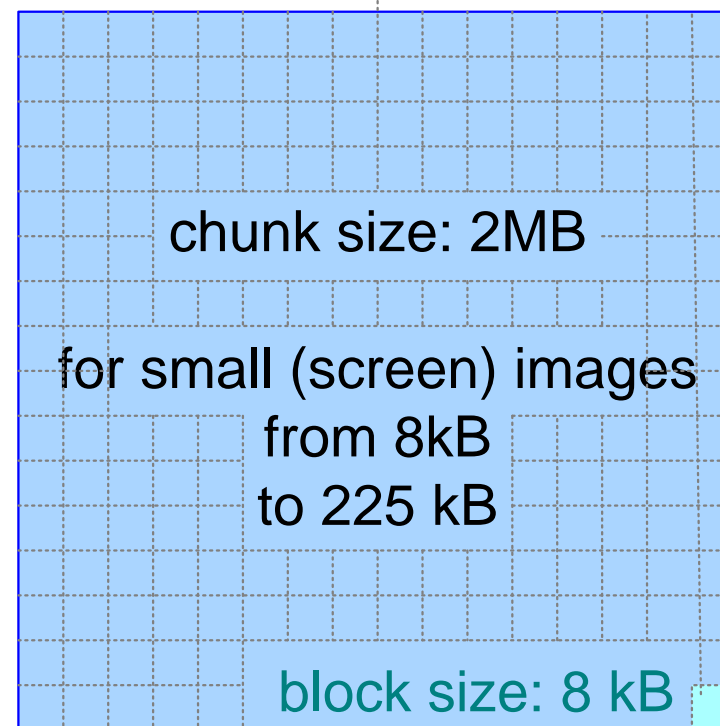
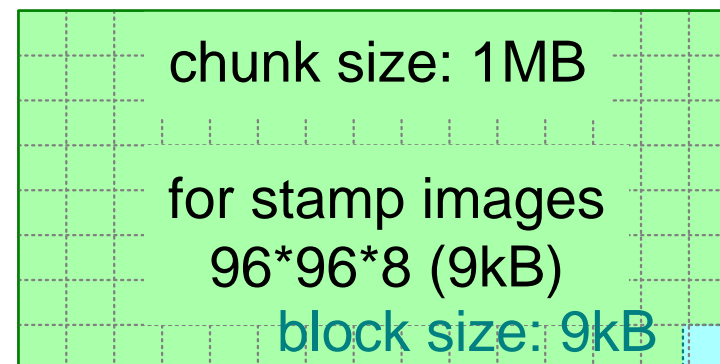
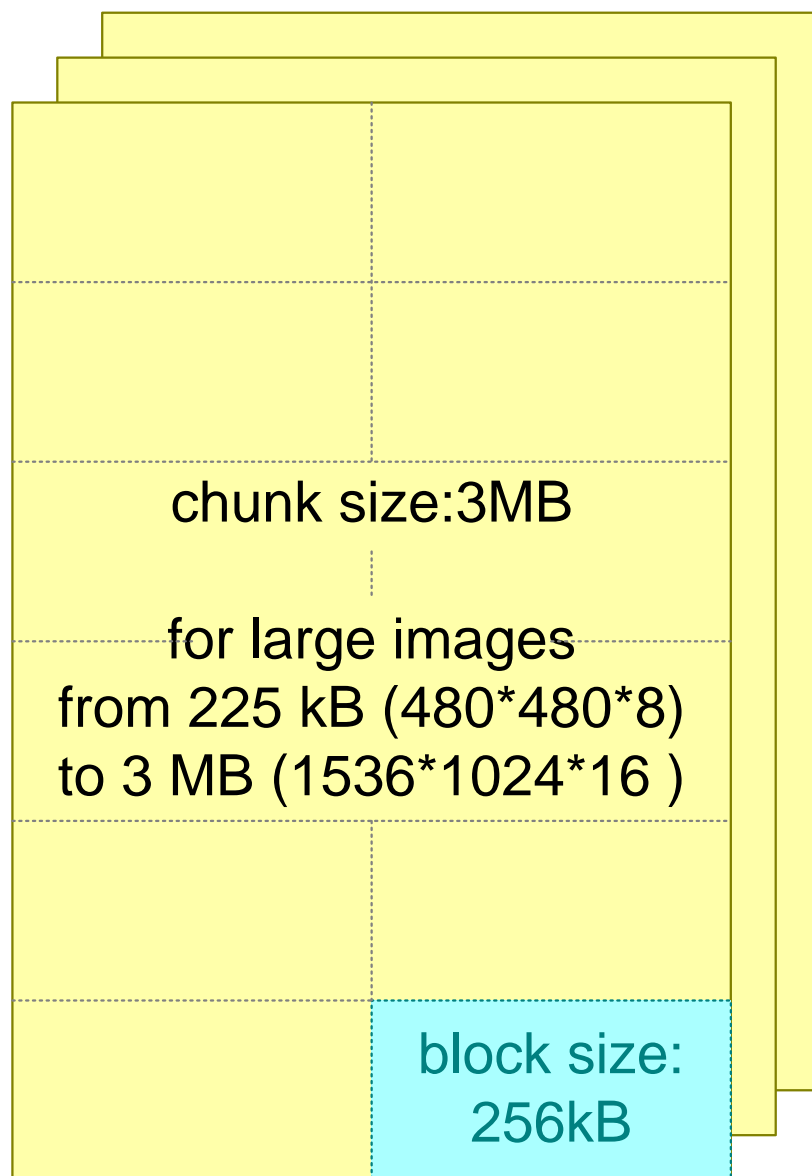
# Memory fragmentation increase



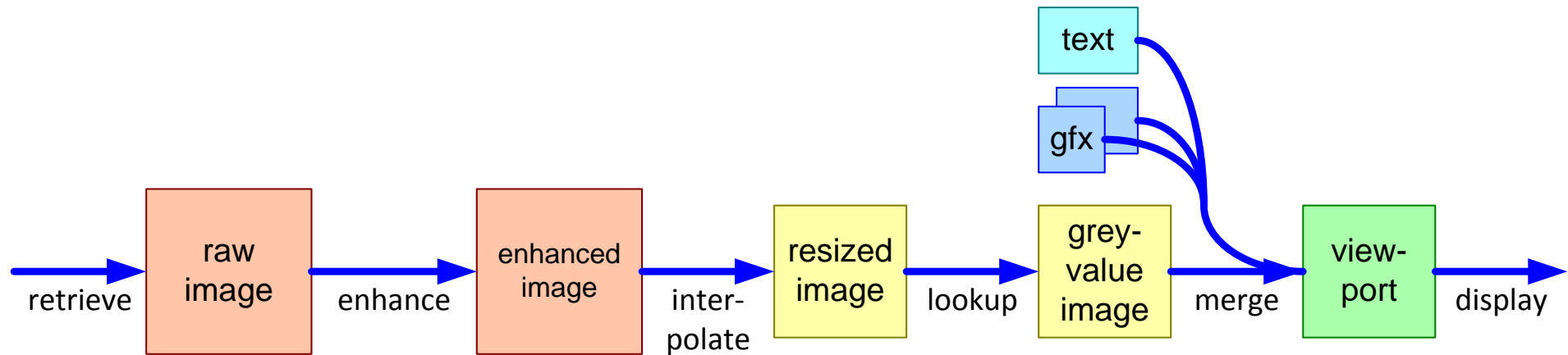
# Cache layers



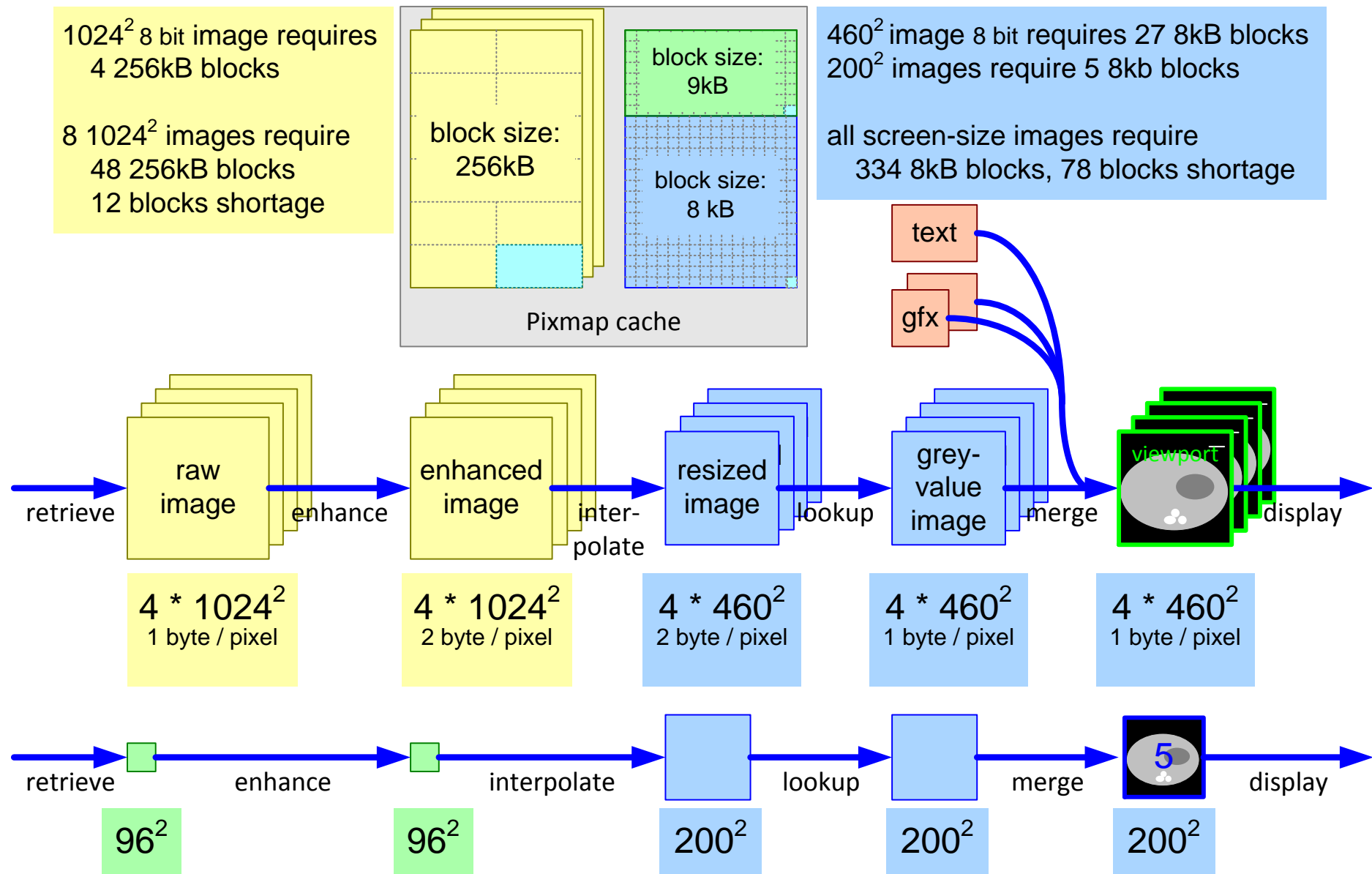
# Bulk data memory management memory allocators



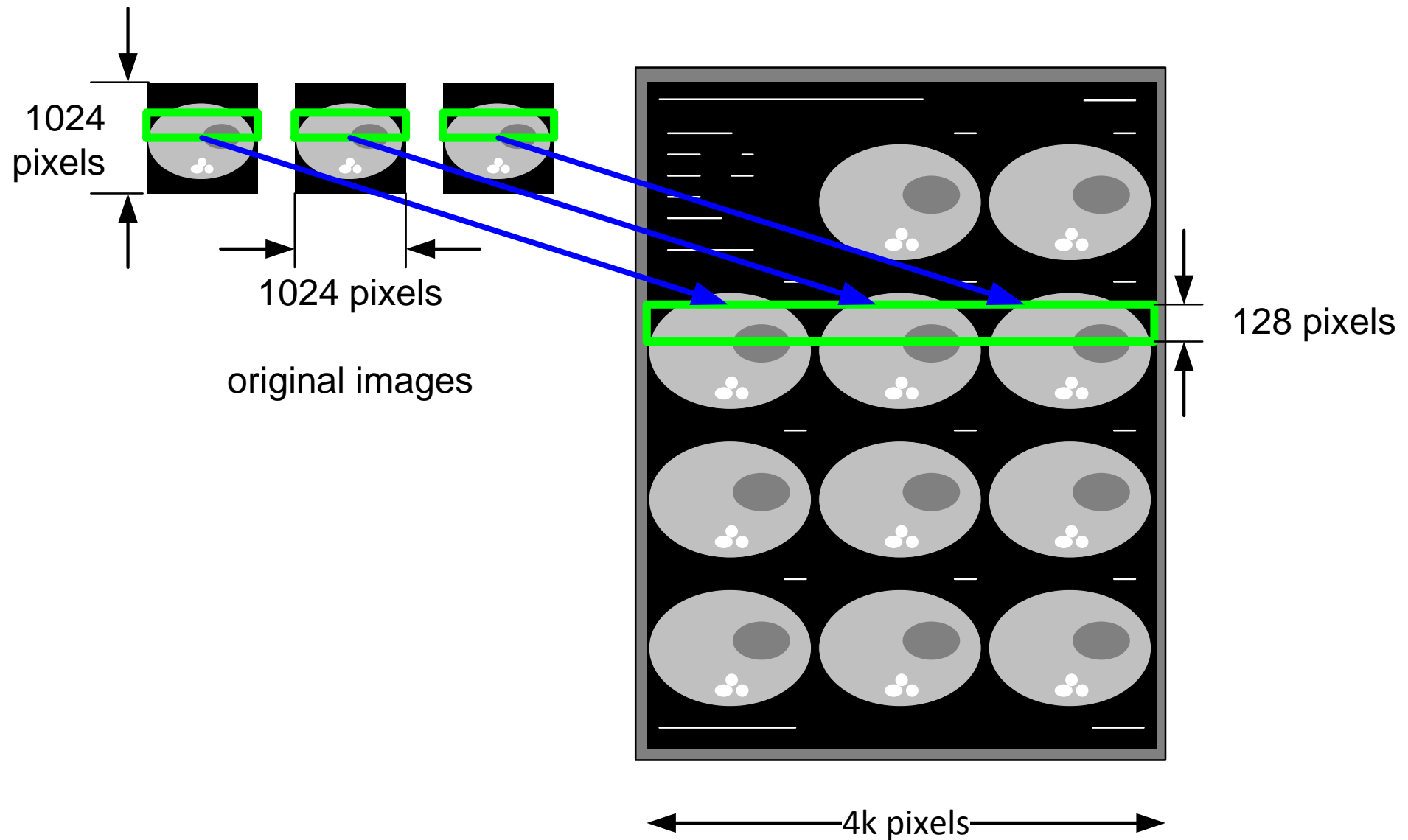
# Cached intermediate processing results



# Example of allocator and cache use

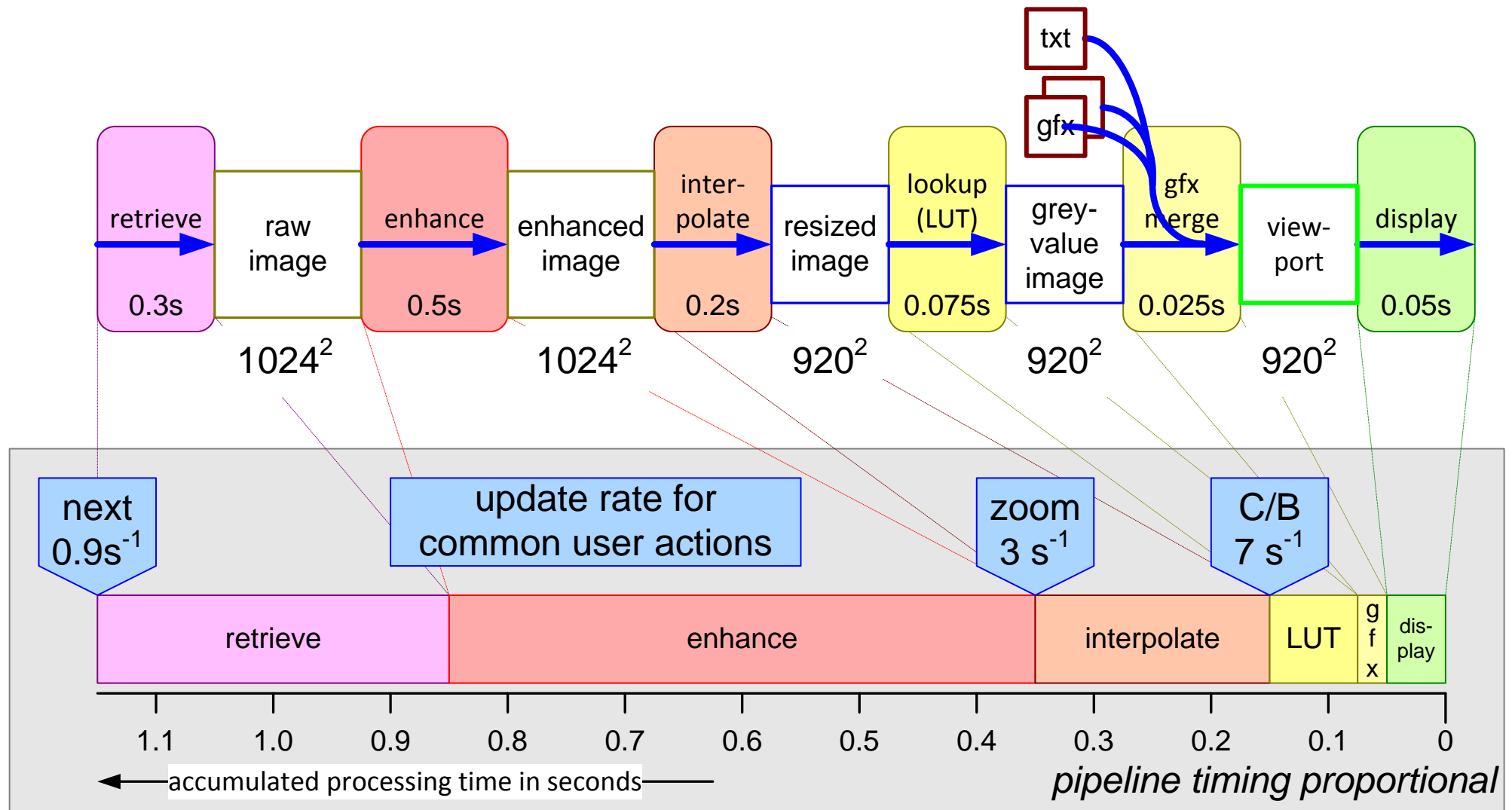


# Print server is based on banding

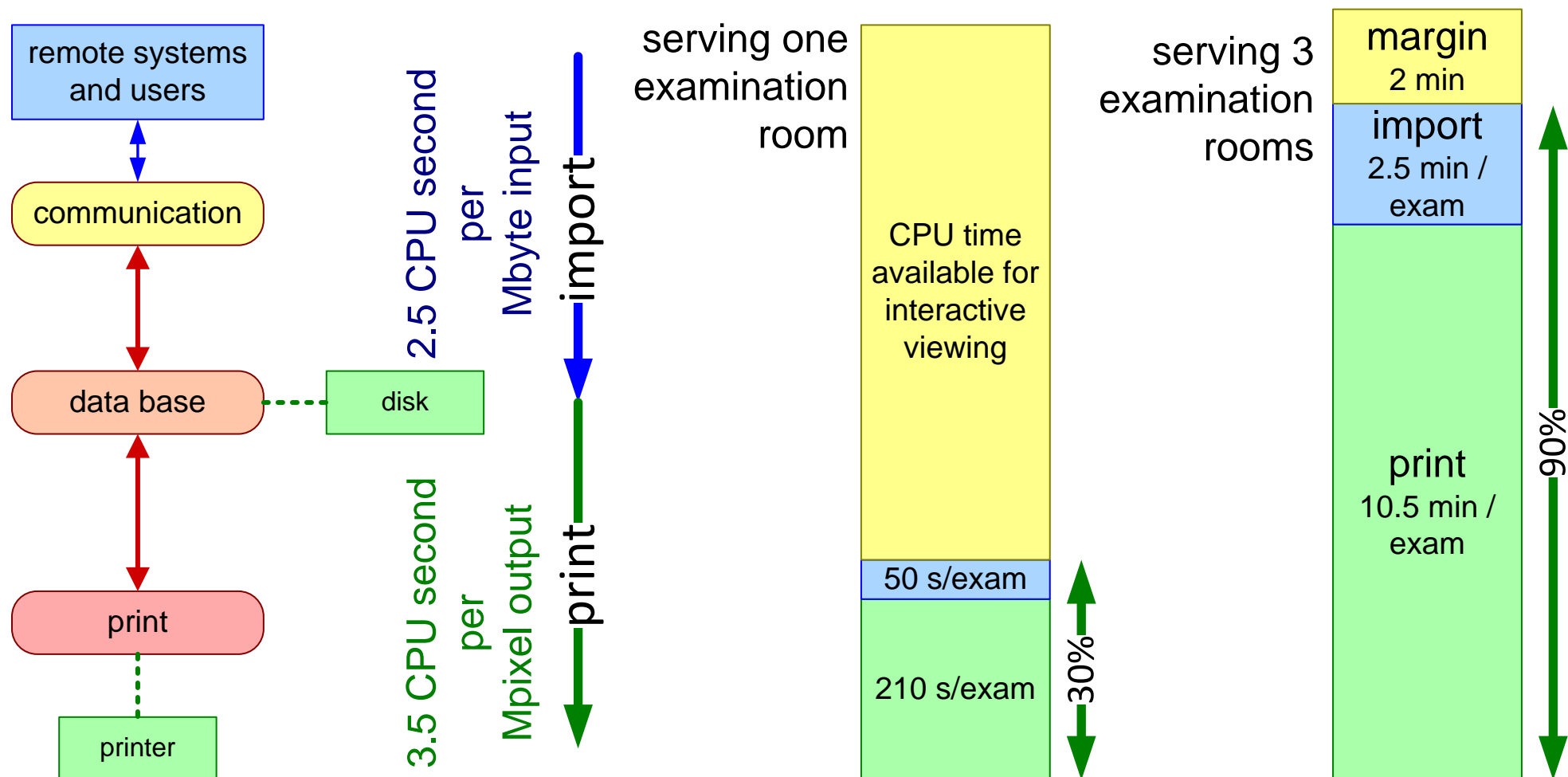




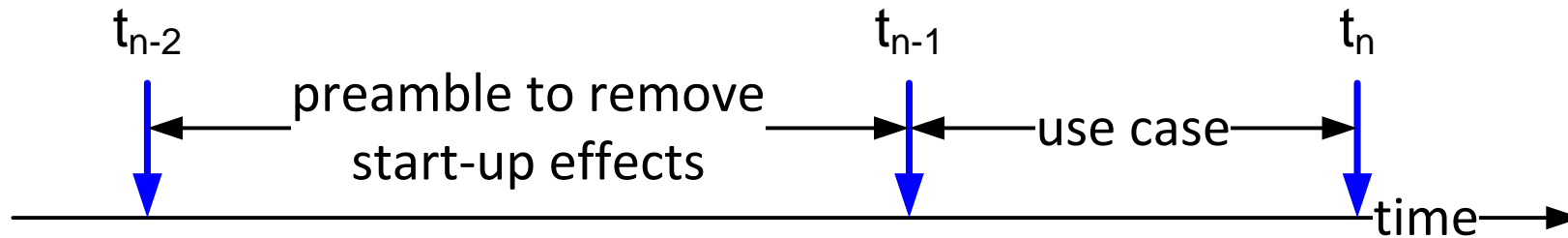
# CPU processing times and viewing responsiveness



# Server CPU load



# Resource measurement tools



oit

$\Delta$  object instantiations  
heap memory usage

ps  
vmstat  
kernel resource  
stats

kernel CPU time  
user CPU time  
code memory  
virtual memory  
paging

heapviewer (visualise fragmentation)

# Object Instantiation Tracing

class name	current nr of objects	deleted since $t_{n-1}$	created since $t_{n-1}$	heap memory usage
AsynchronousIO	0	-3	+3	[819200] [8388608]  [13252]
AttributeEntry	237	-1	+5	
BitMap	21	-4	+8	
BoundedFloatingPoint	1034	-3	+22	
BoundedInteger	684	-1	+9	
BtreeNode1	200	-3	+3	
BulkData	25	0	1	
ButtonGadget	34	0	2	
ButtonStack	12	0	1	
ByteArray	156	-4	+12	

# Overview of benchmarks and other measurement tools

	test / benchmark	what, why	accuracy	when
<i>public</i>	SpecInt (by suppliers)	CPU integer	coarse	new hardware
	Byte benchmark	computer platform performance OS, shell, file I/O	coarse	new hardware new OS release
<i>self made</i>	file I/O	file I/O throughput	medium	new hardware
	image processing	CPU, cache, memory as function of image, pixel size	accurate	new hardware
	Objective-C overhead	method call overhead memory overhead	accurate	initial
	socket, network	throughput CPU overhead	accurate	ad hoc
	data base	transaction overhead query behaviour	accurate	ad hoc
	load test	throughput, CPU, memory	accurate	regression

# Coverage of submethods of the CR views

C	R
<i>construction decomposition</i> <i>functional decomposition</i> <i>designing with multiple decompositions</i> <i>execution architecture</i> <i>internal interfaces</i> <i>performance</i> <i>start up</i> <i>shutdown</i> <i>integration plan</i>  <b>work breakdown</b> <b>safety</b>  reliability security	<i>budget</i> <i>benchmarking</i> <i>performance analysis</i> <i>granularity determination</i>          <b>value and cost</b>   safety analysis reliability analysis security analysis

legend    *explicitly addressed*    **addressed only implicitly**    not addressed

*coverage based on documentation status of first product release*

## disclaimer

The case material is based on actual data, from a complex context with large commercial interests. The material is ***simplified*** to increase the accessibility, while at the same time ***small changes*** have been made to remove commercial sensitivity. Commercial sensitivity is further reduced by using relatively ***old*** data (between 5 and 10 years in the past). Care has been taken that the illustrative value is maintained

# Story Telling in Medical Imaging

by *Gerrit Muller*     University of South-Eastern Norway-NISE

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

## Abstract

Story telling was not used explicit during the development of the medical imaging workstation. Two stories which did have a great impact of the development of the product are described: “The sales story” and “The radiologist at work”. The relation of the stories to the requirements and design is shown.

### Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

September 1, 2020  
status: finished  
version: 1.2





# Stories used during development

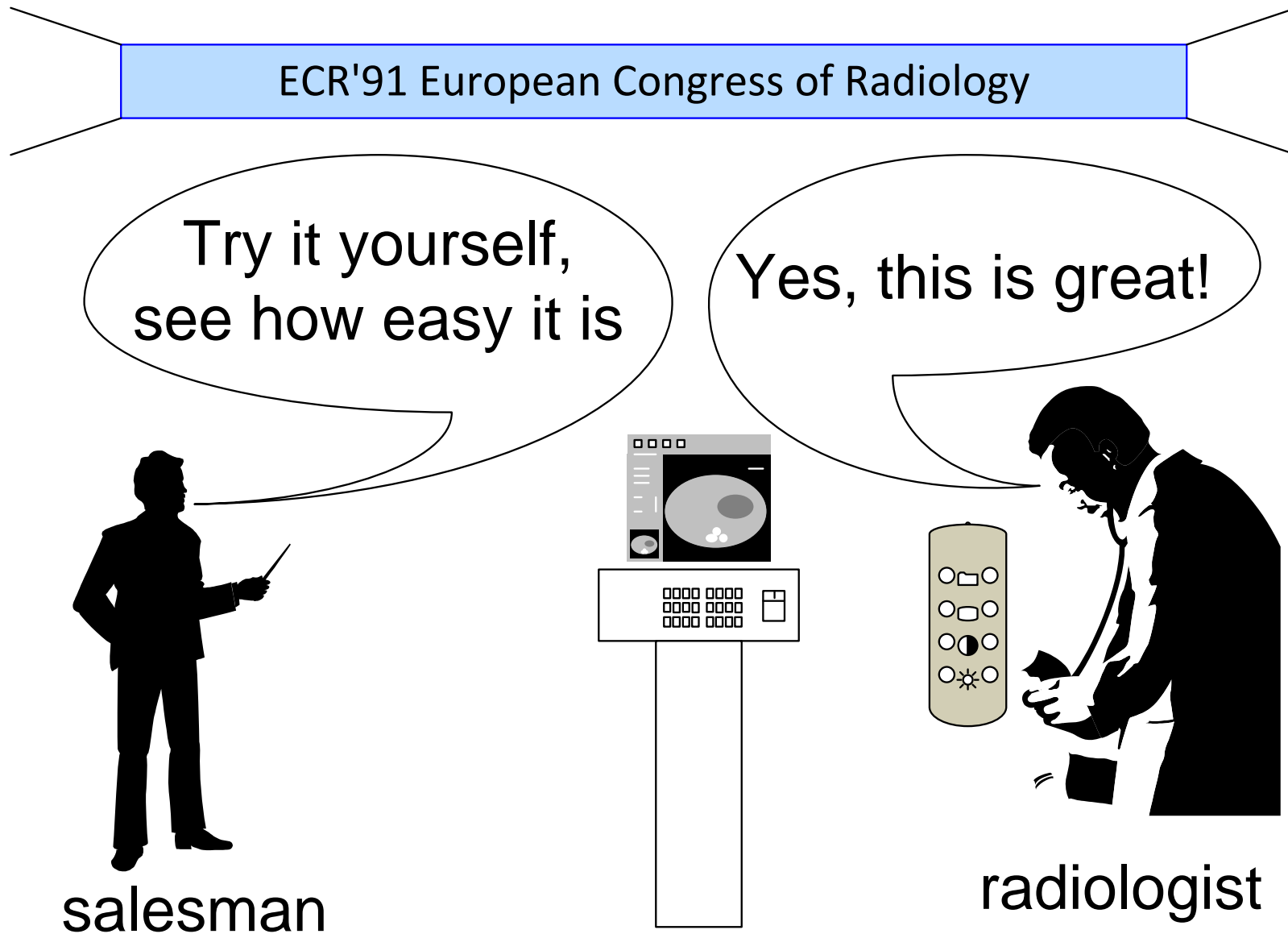
---

**The sales story** how to capture the interest of the radiologist for the product.

**The radiologist at work** describing the way a radiologist works, which explains why the radiologist is **not** interested in viewing, but very interested in films.

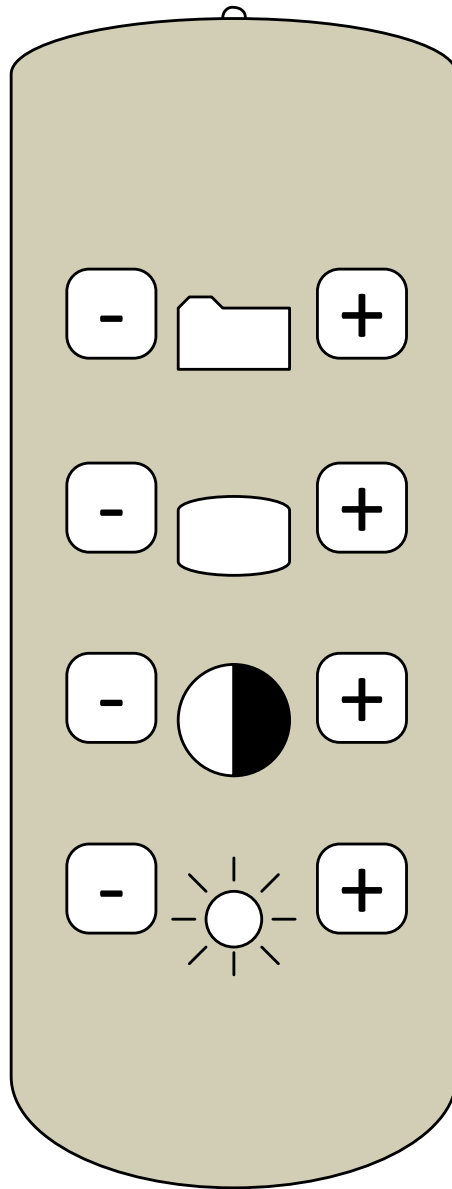
**The gastro intestinal examination** how the URF system is used to examine patients with gastro intestinal problems. This story is not described here, because it is outside the scope of the discussed thread of reasoning

# Main sales feature: easy viewing



# Remote control makes viewing easy

---



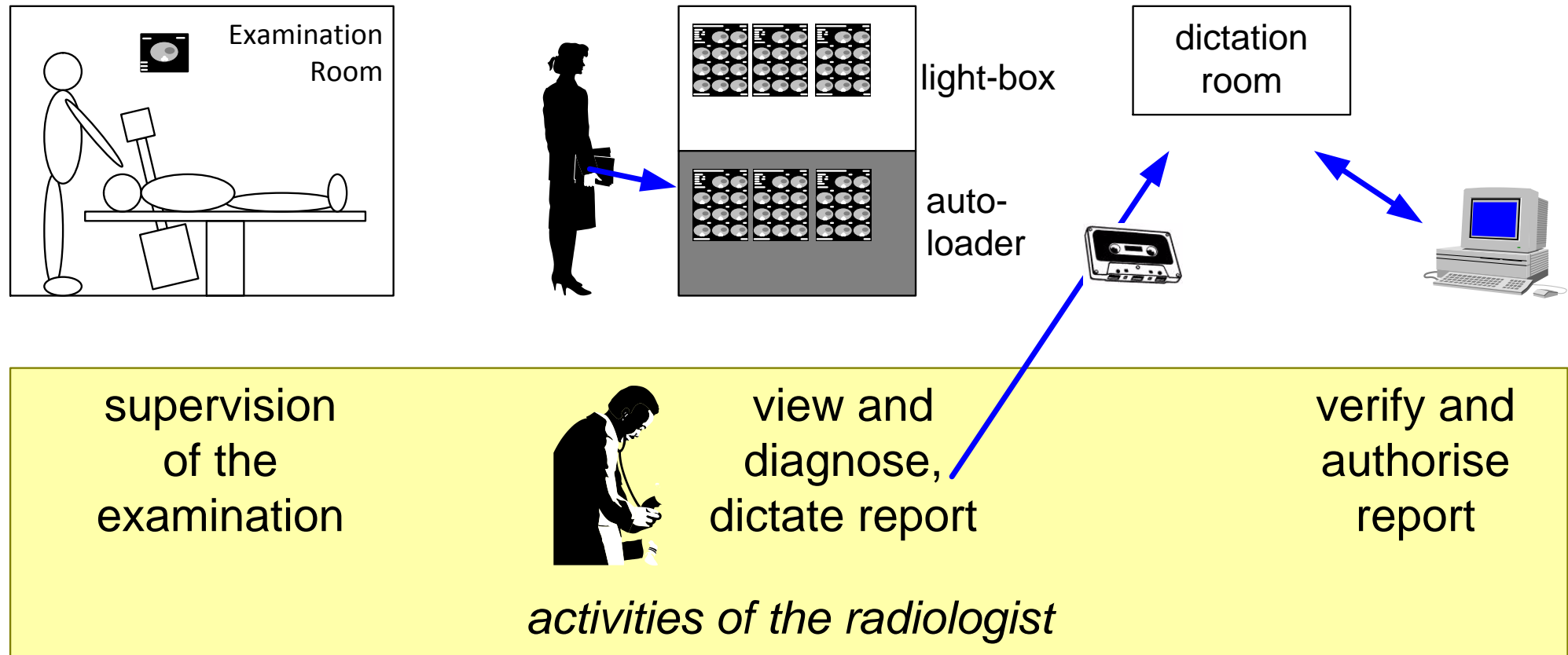
next / previous examination

next / previous image

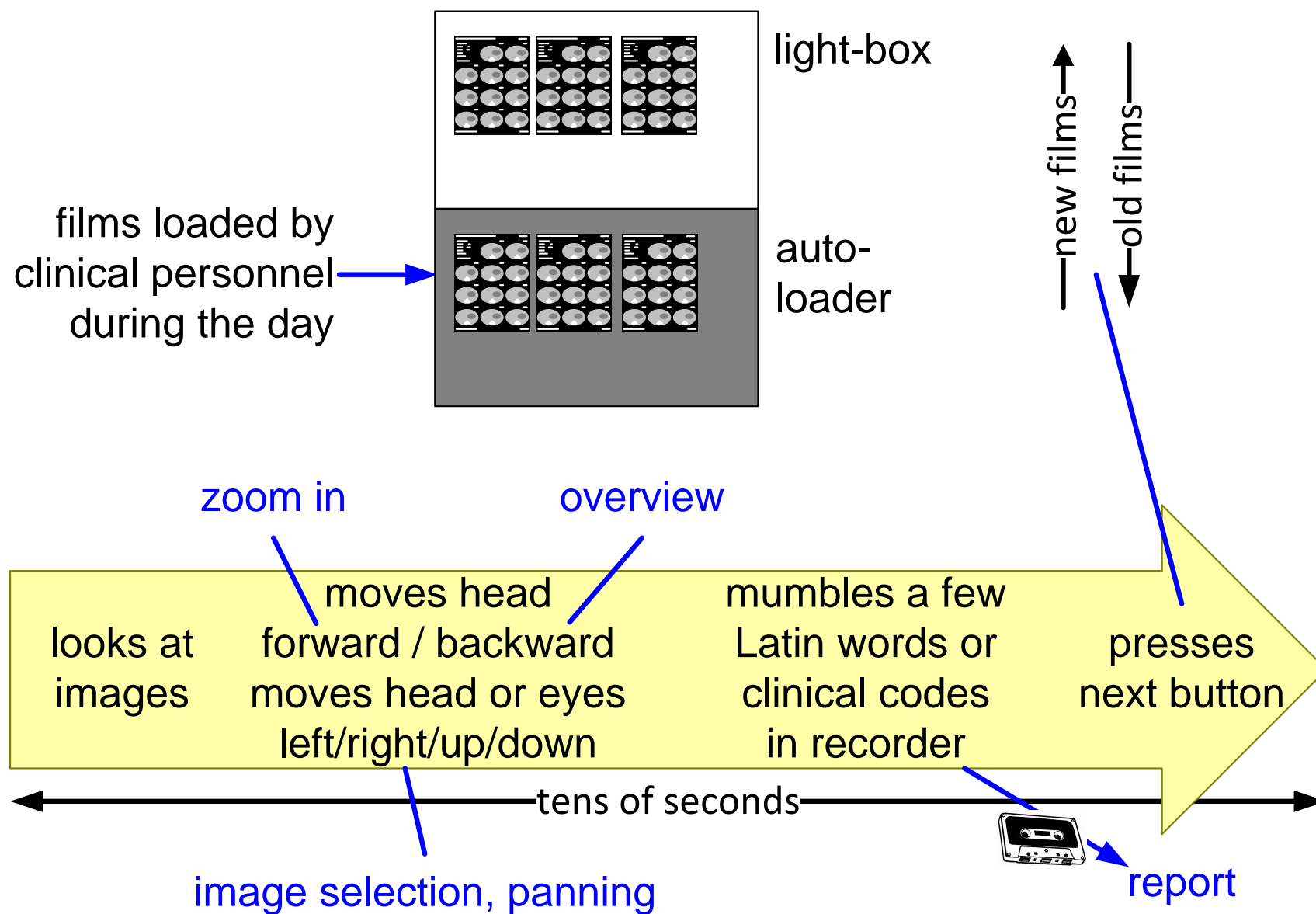
increase / decrease contrast

increase / decrease brightness

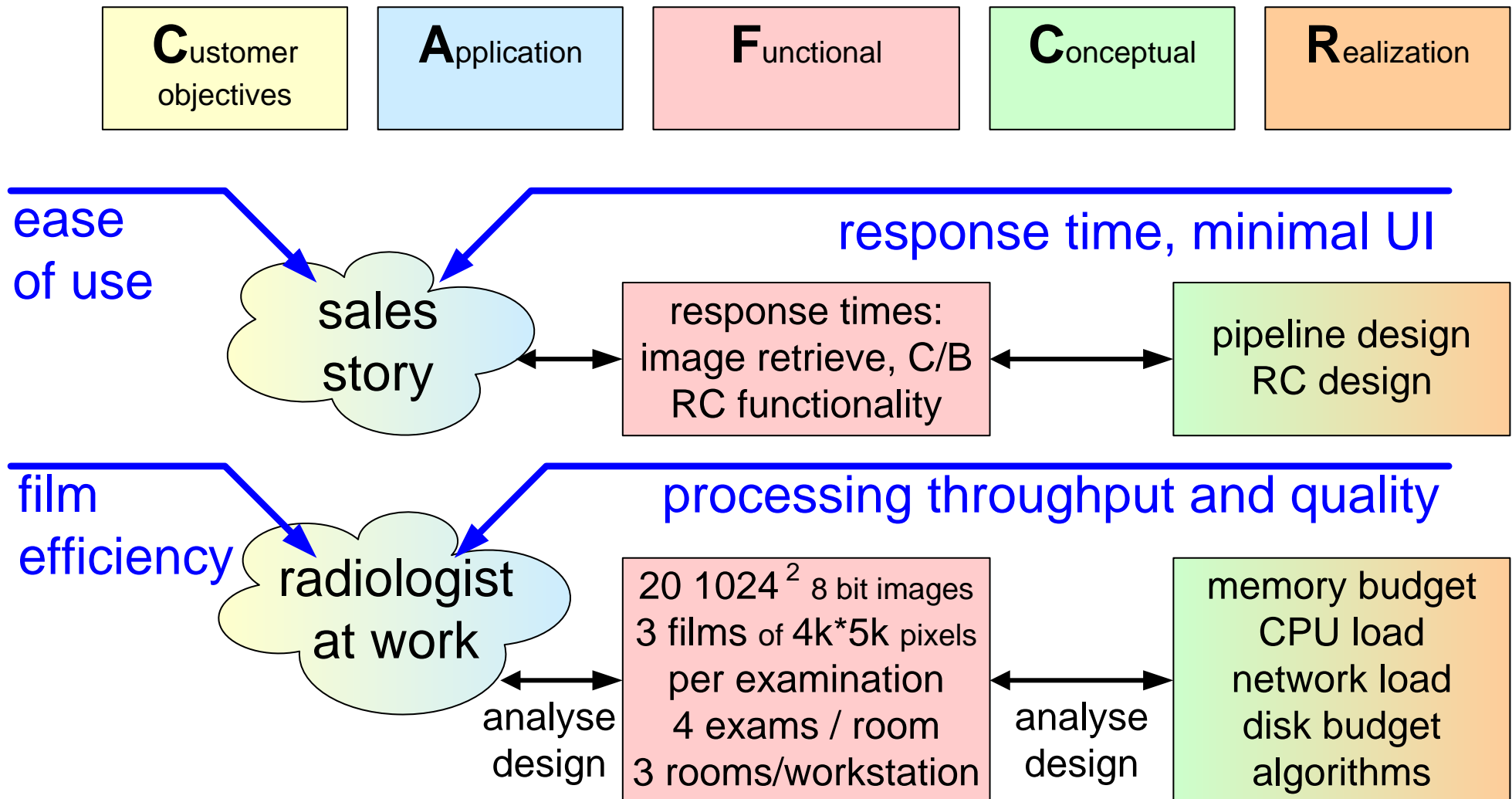
# Radiologist workspots and activities



# Diagnosis in tens of seconds



# From story to design



# Threads of Reasoning in the Medical Imaging Case

by *Gerrit Muller*      University of South-Eastern Norway-NISE

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

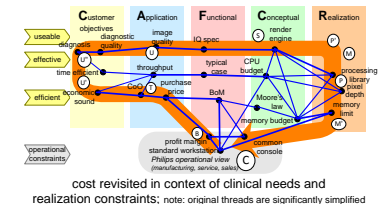
## Abstract

A thread of reasoning is build up in steps and the underlying reasoning is explained.

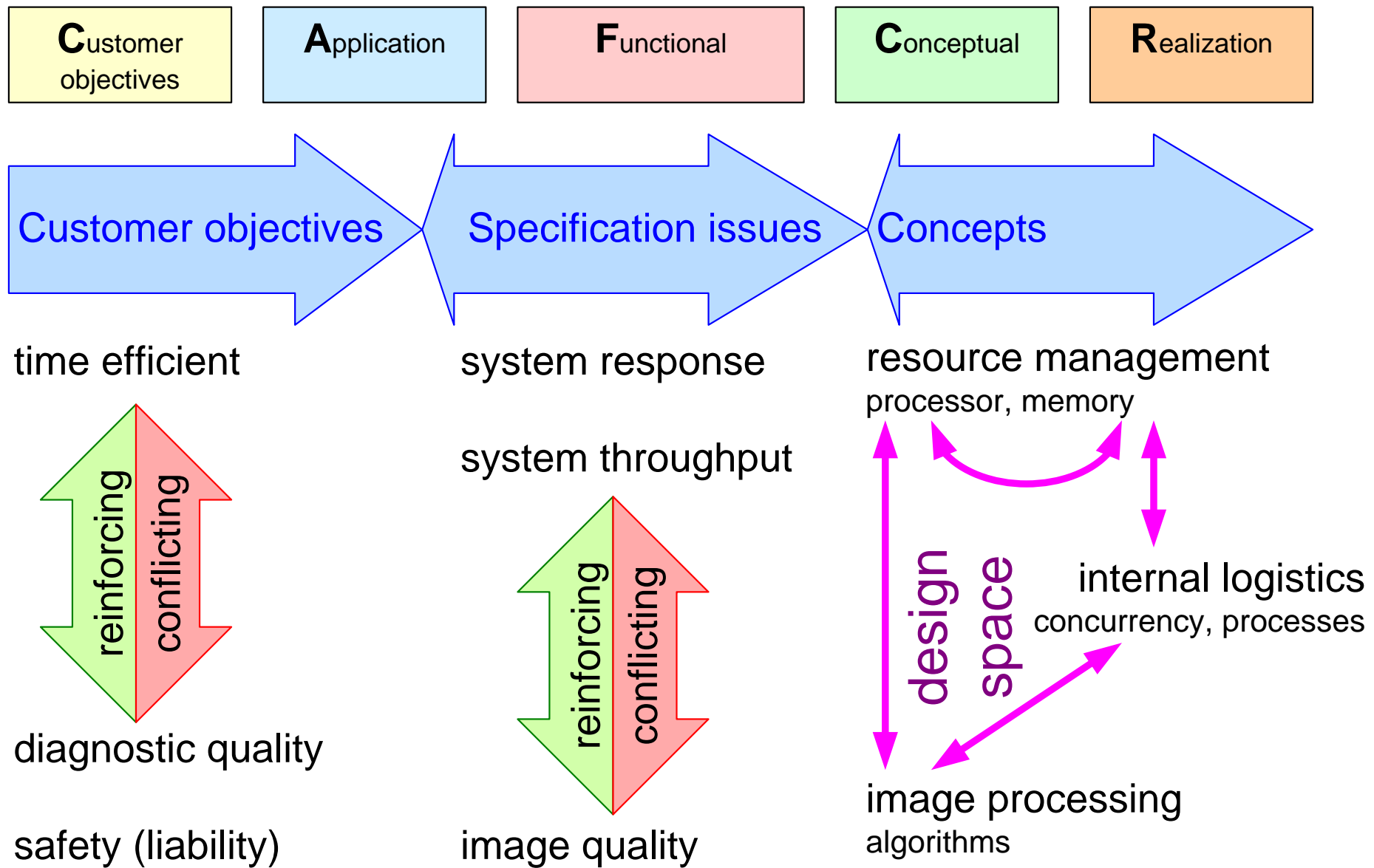
## Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

September 1, 2020  
status: finsihed  
version: 2.4

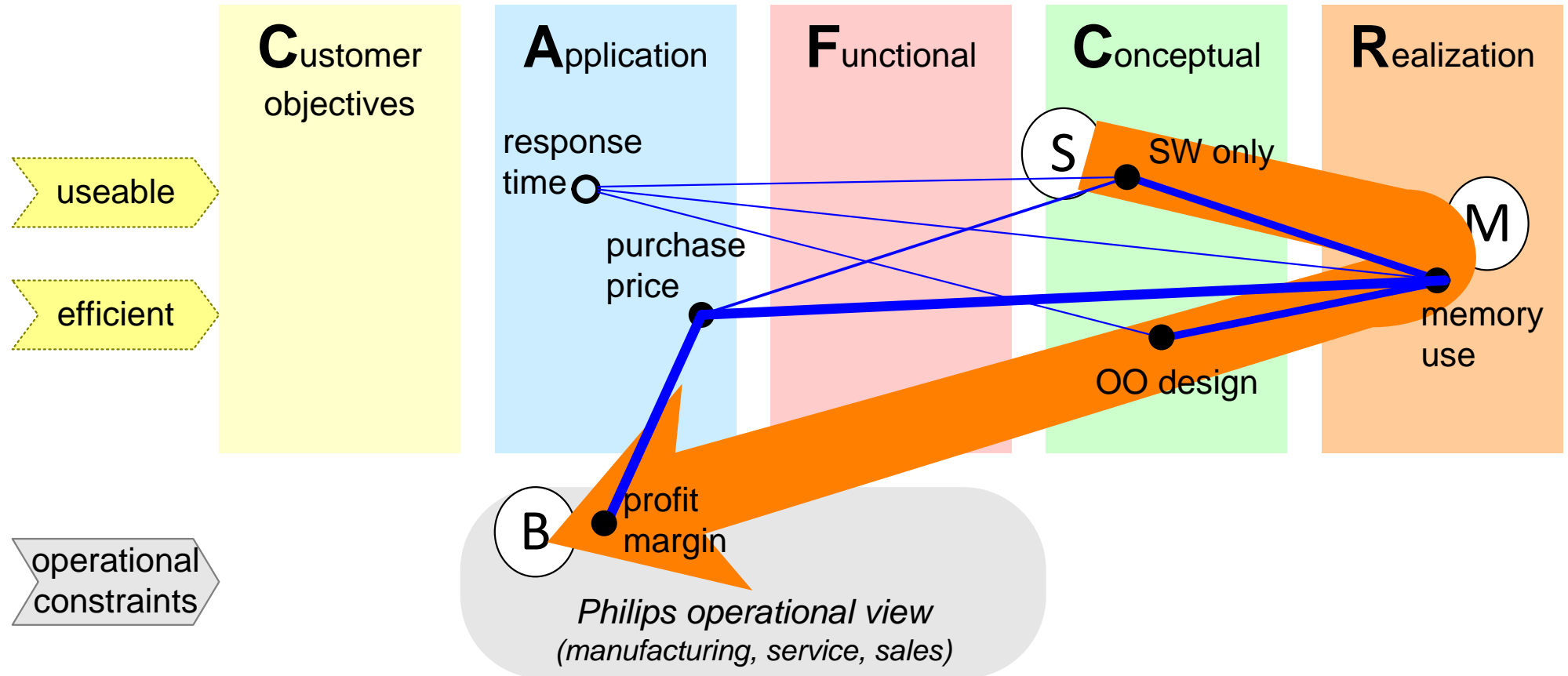


# Thread of reasoning based on efficiency-quality tension



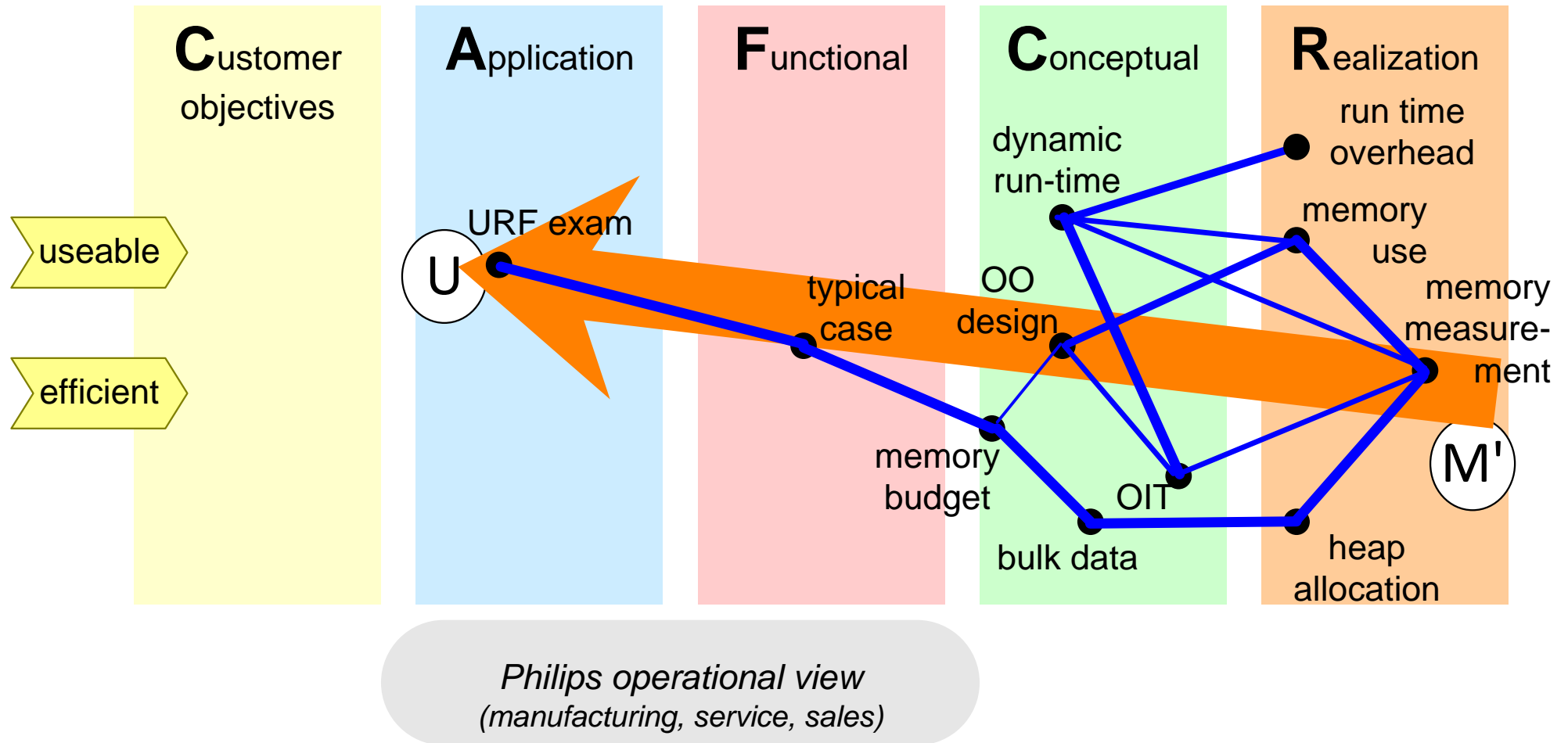


# Thread of reasoning; introvert phase



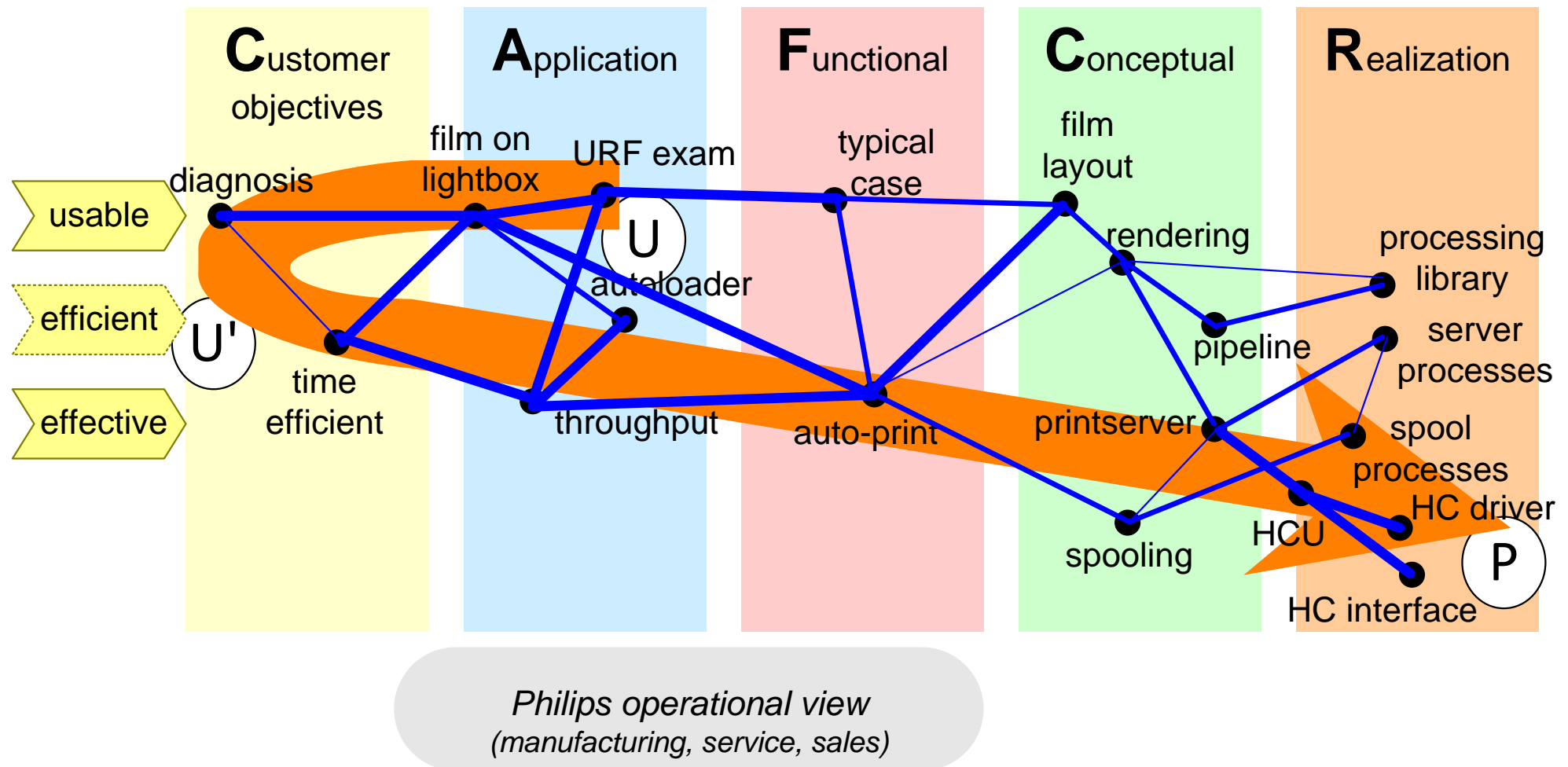
Introvert view: cost and impact of new technologies

# Thread of reasoning; phase 2



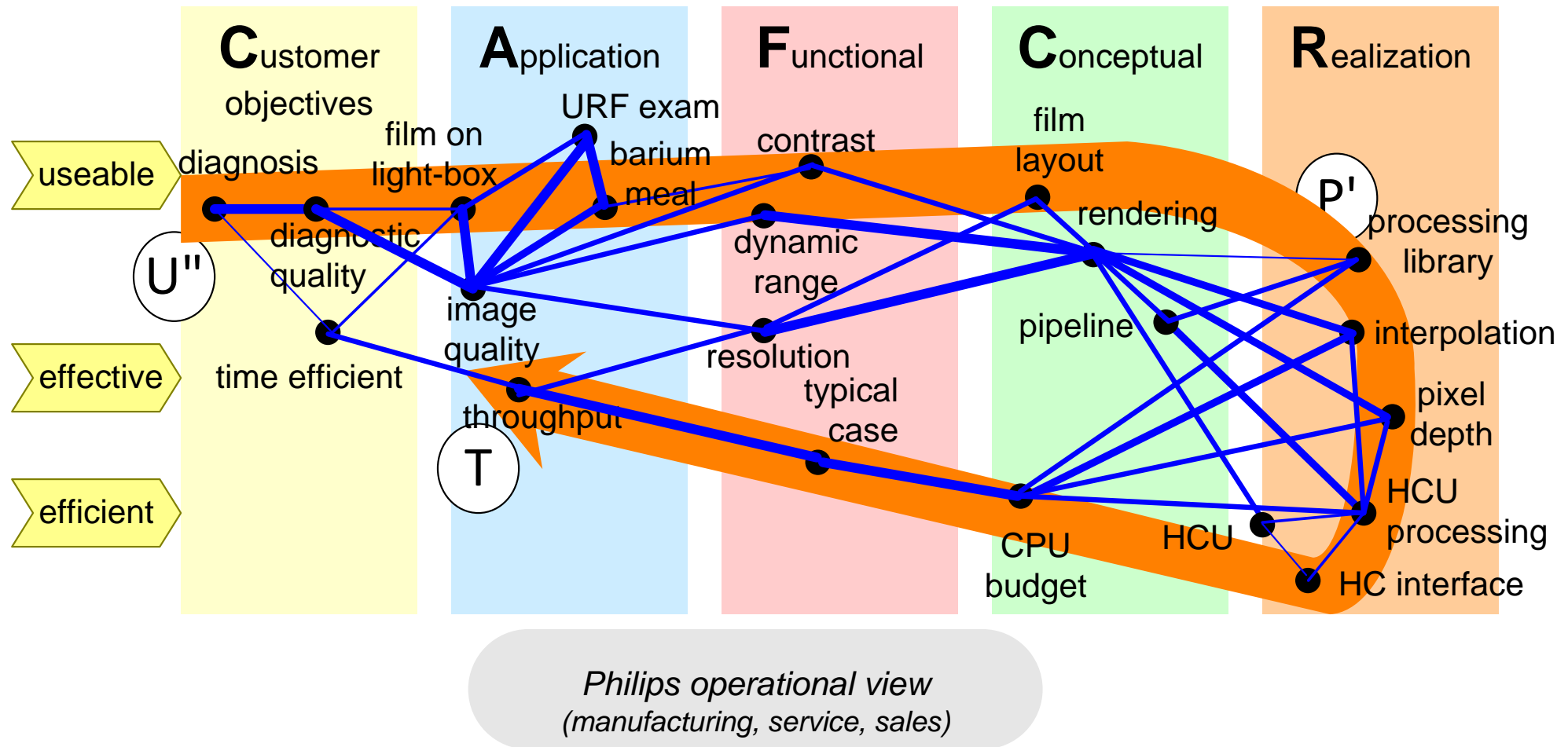
How to measure memory, how much is needed?  
from introvert to extrovert

# Thread of reasoning; phase 3



Radiologists diagnose from film, throughput is important  
Extrovert view shows conceptual and realization gaps!

# Thread of reasoning; phase 4



from extrovert diagnostic quality, via image quality, algorithms and load, to extrovert throughput

Threads of Reasoning in the Medical Imaging Case  
61 Gerrit Muller

