

Submethods in the CR Views

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

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

Abstract

This chapter describes the *Conceptual* view and the *Realization* view. Both views are supported by a set of submethods to describe multi-disciplinary design, for example several decompositions and models are provided.

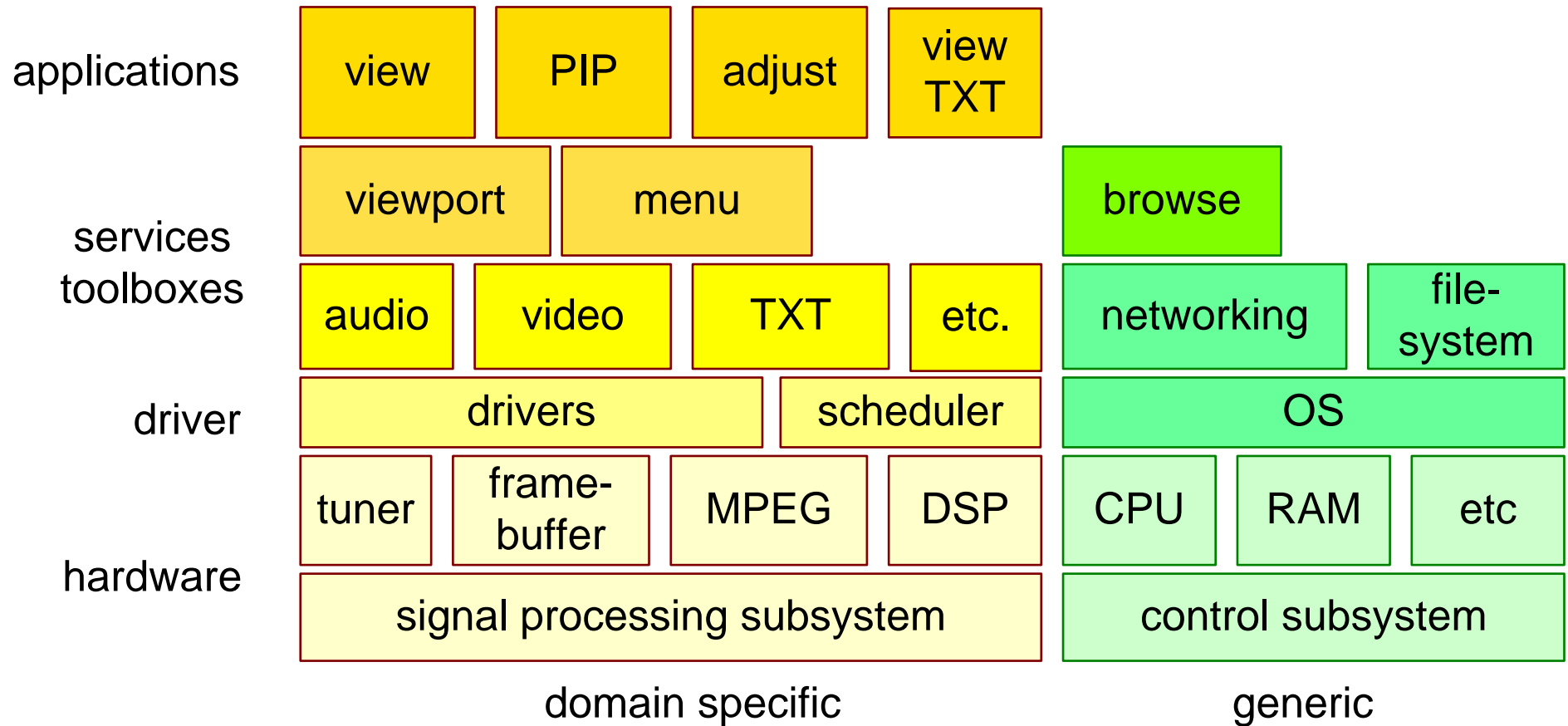
Distribution

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.

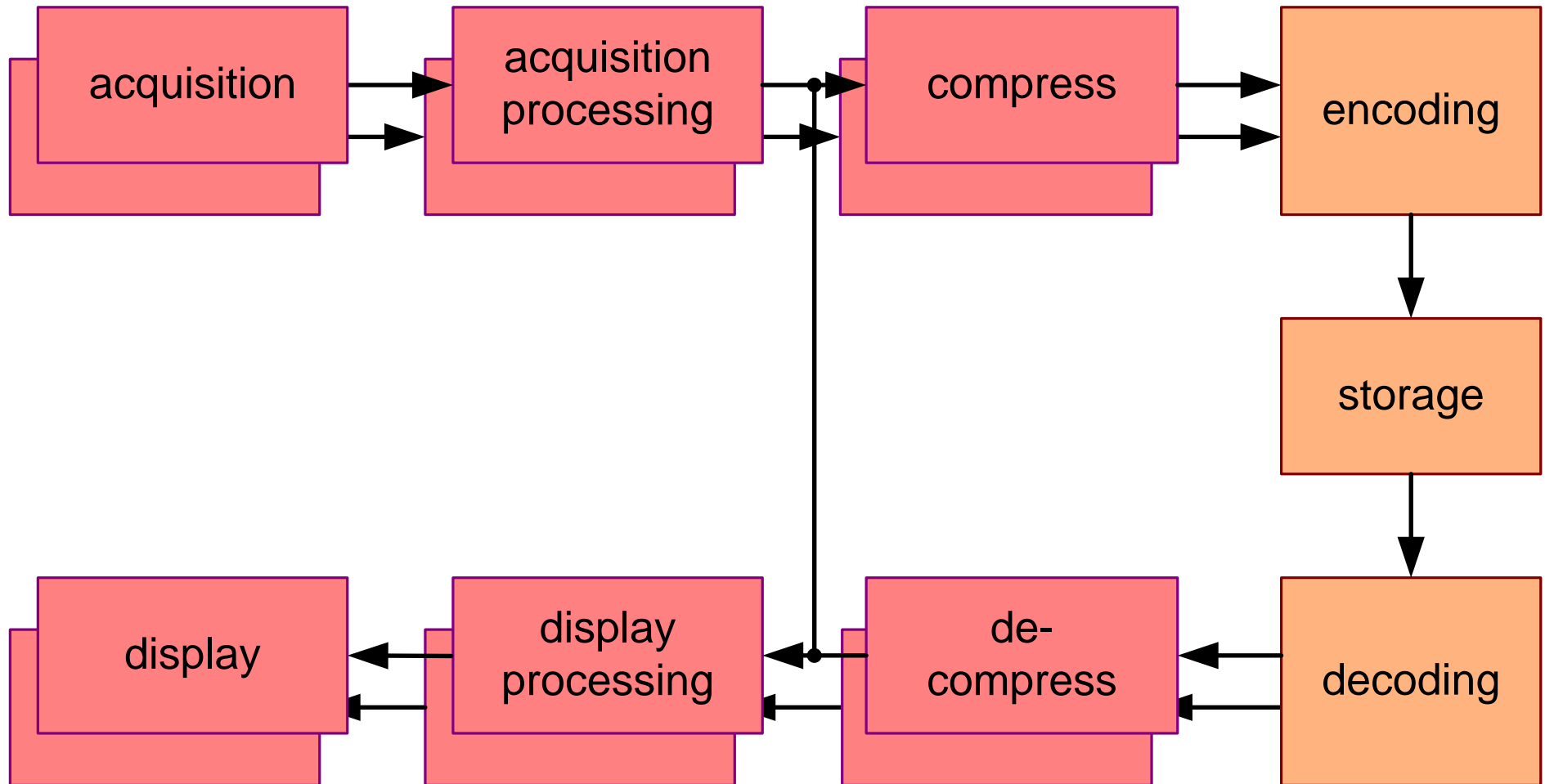
September 9, 2018
status: finished
version: 1.5

logo
TBD

Construction Decomposition



Functional Decomposition

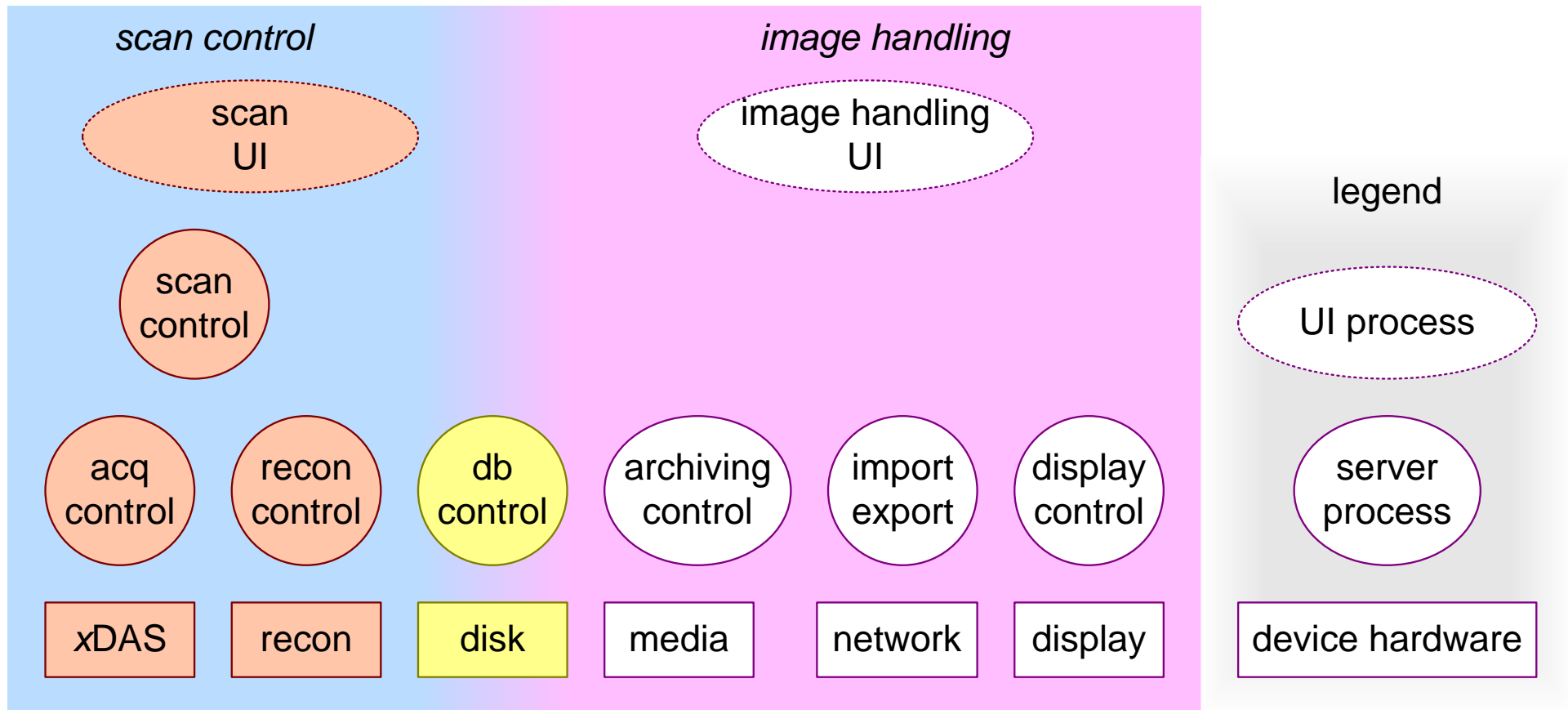


How about the **<characteristic>**
of the **<component>**
when performing **<function>**?

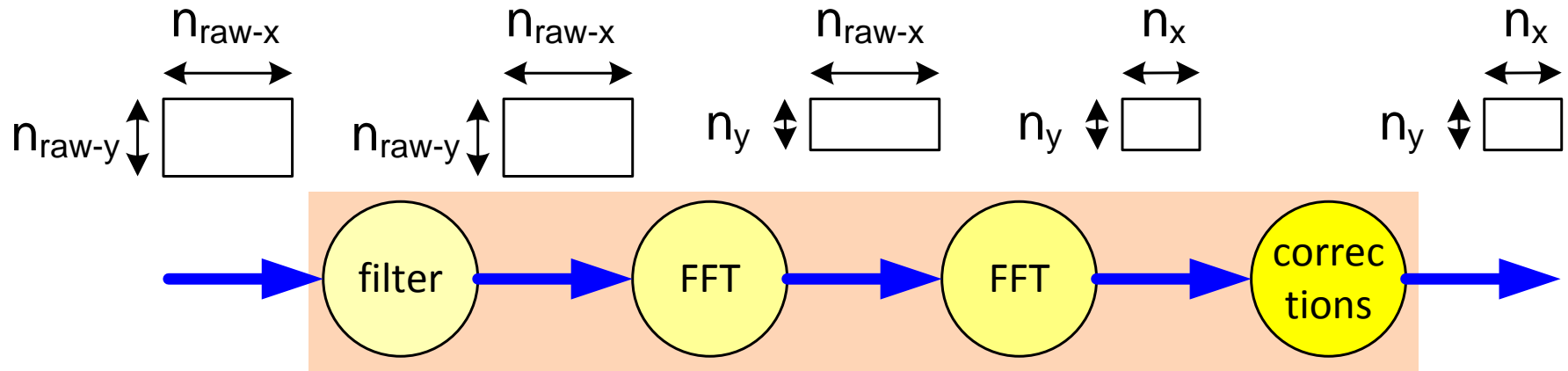
characteristics	SNR accuracy memory usage processing latency ...
components	import server user interface print server database server export server ...
functions	query DB render film play movie next brightness ...

What is the **memory usage** of
the **user interface**
when **querying the DB**

Process Decomposition



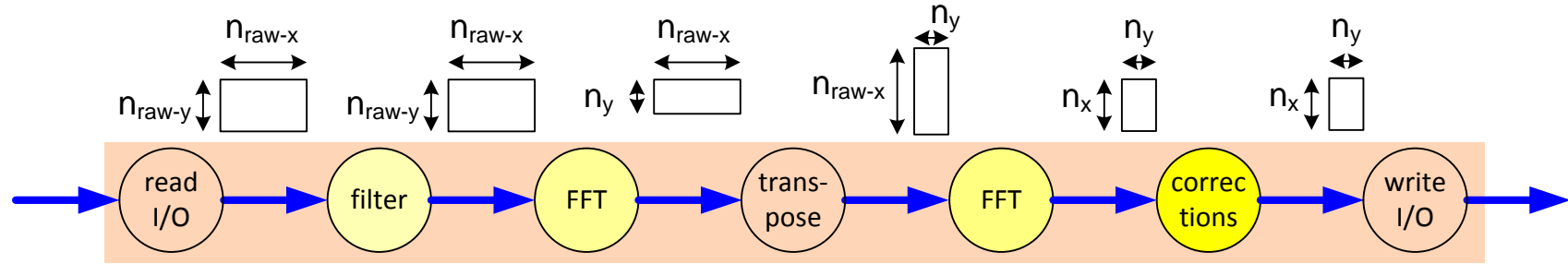
Conceptual Performance Model



$$t_{\text{recon}} = t_{\text{filter}}(n_{\text{raw-x}}, n_{\text{raw-y}}) + n_{\text{raw-x}} * (t_{\text{fft}}(n_{\text{raw-y}}) + t_{\text{col-overhead}}) + n_y * (t_{\text{fft}}(n_{\text{raw-x}}) + t_{\text{row-overhead}}) + t_{\text{corrections}}(n_x, n_y) + t_{\text{control-overhead}}$$

$$t_{\text{fft}}(n) = c_{\text{fft}} * n * \log(n)$$

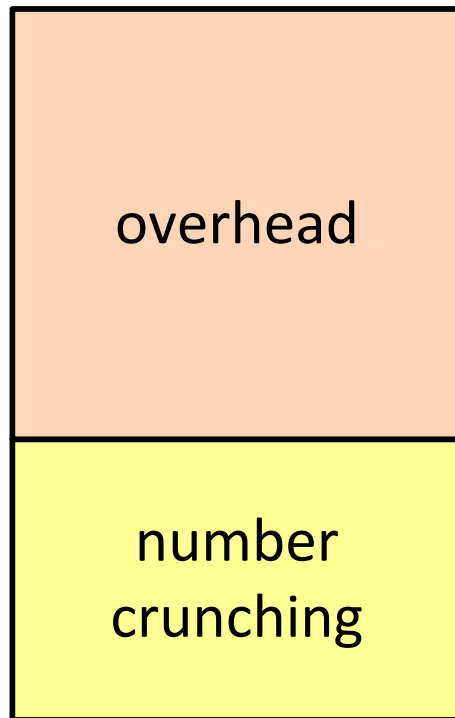
Model After More Detailed Performance Analysis



$$t_{\text{recon}} = t_{\text{filter}}(n_{\text{raw-x}}, n_{\text{raw-y}}) + n_{\text{raw-x}} * (t_{\text{fft}}(n_{\text{raw-y}}) + t_{\text{col-overhead}}) + n_{\text{y}} * (t_{\text{fft}}(n_{\text{raw-x}}) + t_{\text{row-overhead}}) + t_{\text{corrections}}(n_{\text{x}}, n_{\text{y}}) + t_{\text{read I/O}} + t_{\text{transpose}} + t_{\text{write I/O}} + t_{\text{control-overhead}}$$

$t_{\text{fft}}(n) = c_{\text{fft}} * n * \log(n)$

bookkeeping
transpose
malloc, free
write I/O
read I/O
overhead
correction computations
row overhead
FFT computations
column overhead
FFT computations
overhead
filter computations



focus on overhead reduction

is more important

than faster algorithms

this is not an excuse for sloppy algorithms

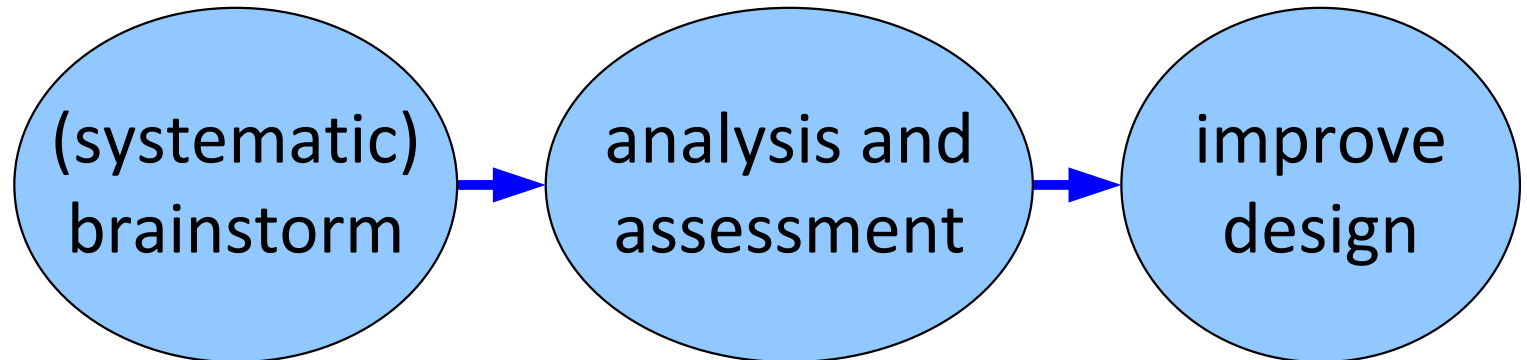
Micro Benchmarks

	<i>infrequent operations, often time-intensive</i>	<i>often repeated operations</i>
<i>database</i>	start session finish session	perform transaction query
<i>network, I/O</i>	open connection close connection	transfer data
<i>high level construction</i>	component creation component destruction	method invocation same scope other context
<i>low level construction</i>	object creation object destruction	method invocation
<i>basic programming</i>	memory allocation memory free	function call loop overhead basic operations (add, mul, load, store)
<i>OS</i>	task, thread creation	task switch interrupt response
<i>HW</i>	power up, power down boot	cache flush low level data transfer

Budget Approach

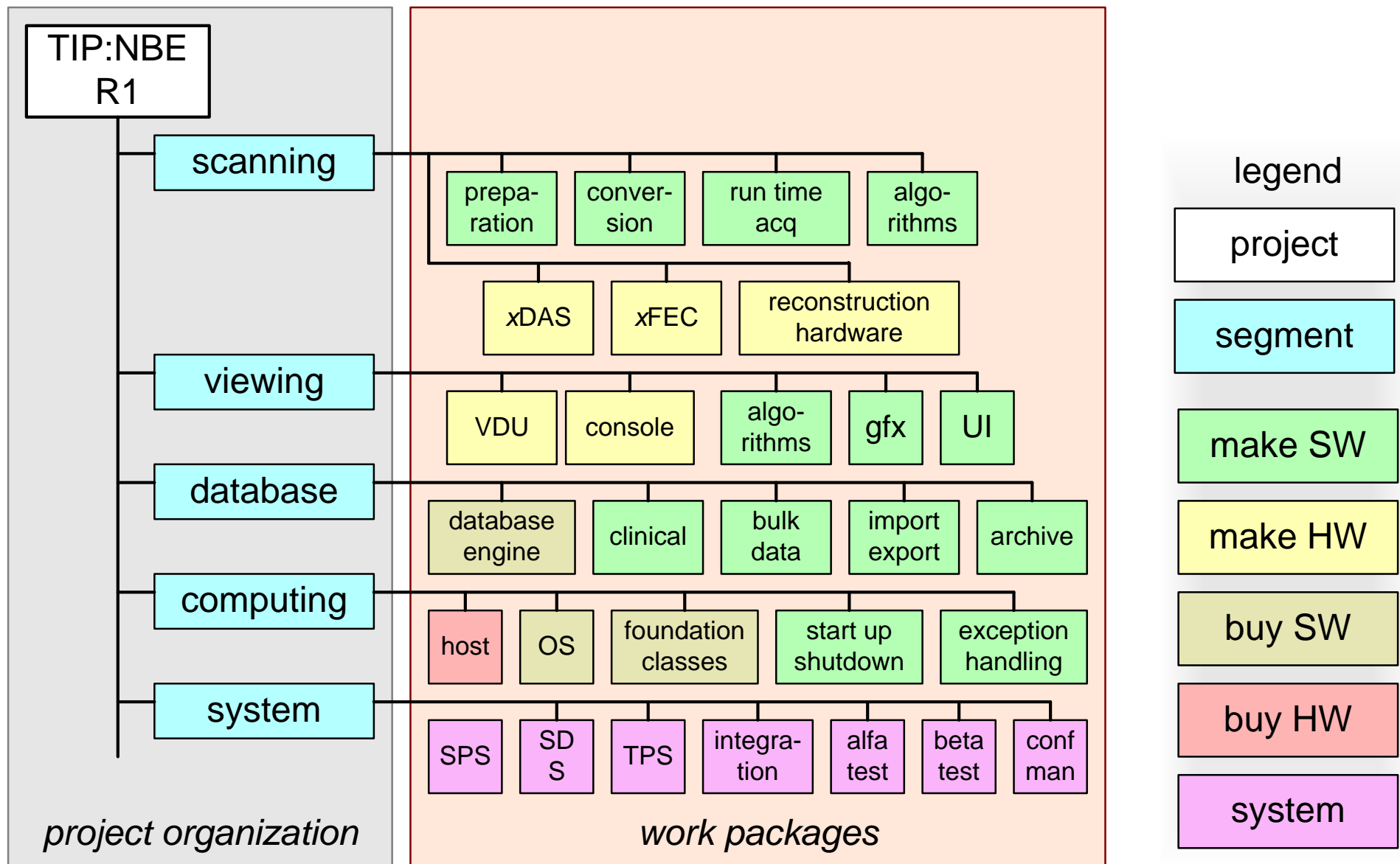
step	example
1A measure old systems	micro-benchmarks, aggregated functions, applications
1B model the performance starting with old systems	flow model and analytical model
1C determine requirements for new system	response time or throughput
2 make a design for the new system	explore design space, estimate and simulate
3 make a budget for the new system:	models provide the structure measurements and estimates provide initial numbers specification provides bottom line
4 measure prototypes and new system	micro-benchmarks, aggregated functions, applications profiles, traces
5 Iterate steps 1B to 4	

Safety, Reliability and Security Analysis

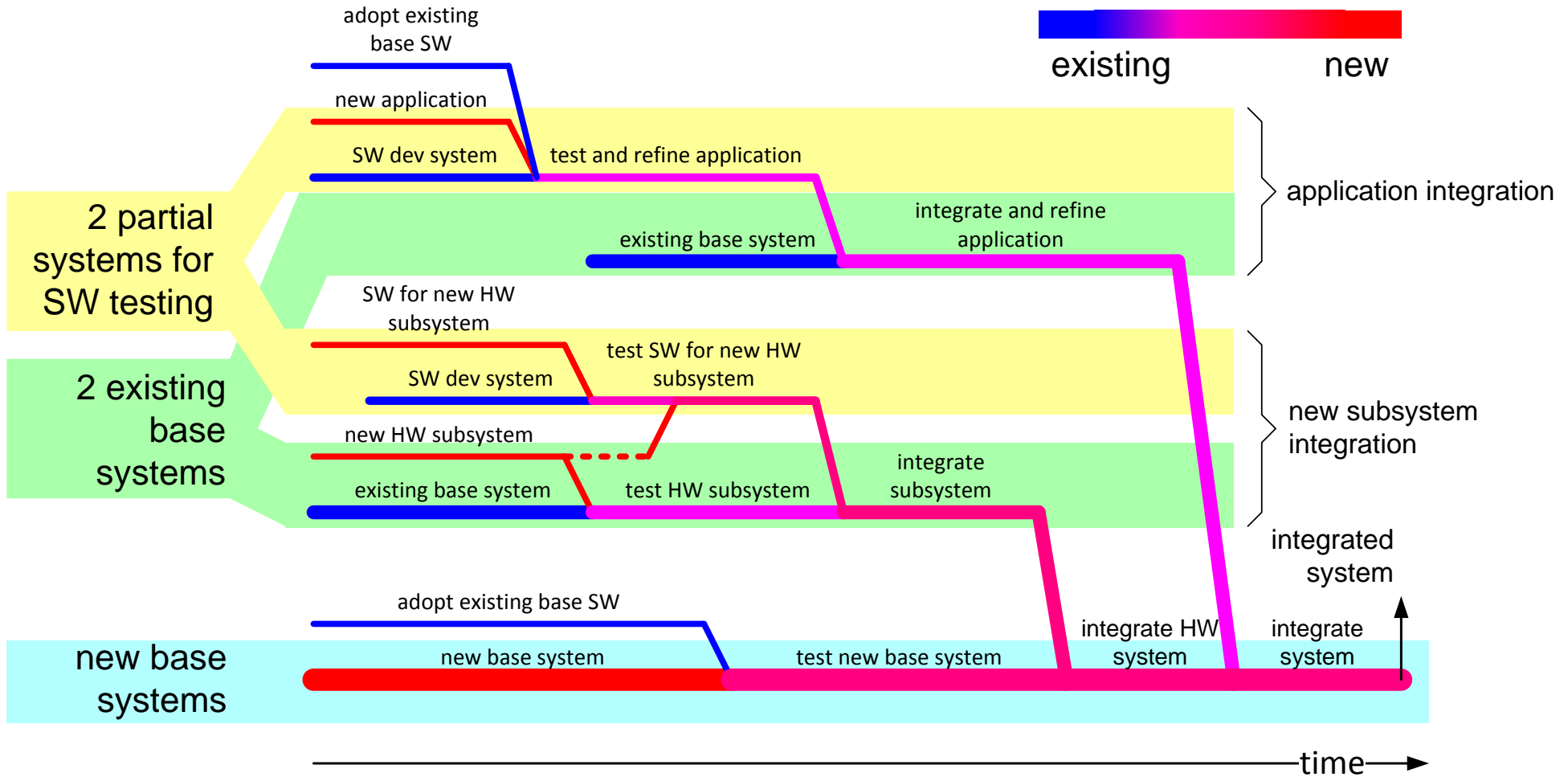


safety hazard analysis	potential hazards	probability severity	measures
reliability FMEA	failure modes	effects	measures
security	vulnerability risks	consequences	measures

Work Break Down



Integration Plan



Overview CR Submethods

Conceptual

construction decomposition
functional decomposition
designing with multiple decompositions
execution architecture
internal interfaces
performance
start up
shutdown
integration plan
work breakdown
safety
reliability
security

Realization

budget
benchmarking
performance analysis
value and cost
safety analysis
reliability analysis
security analysis
granularity determination