# Submethods in the CR Views

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

## Abstract

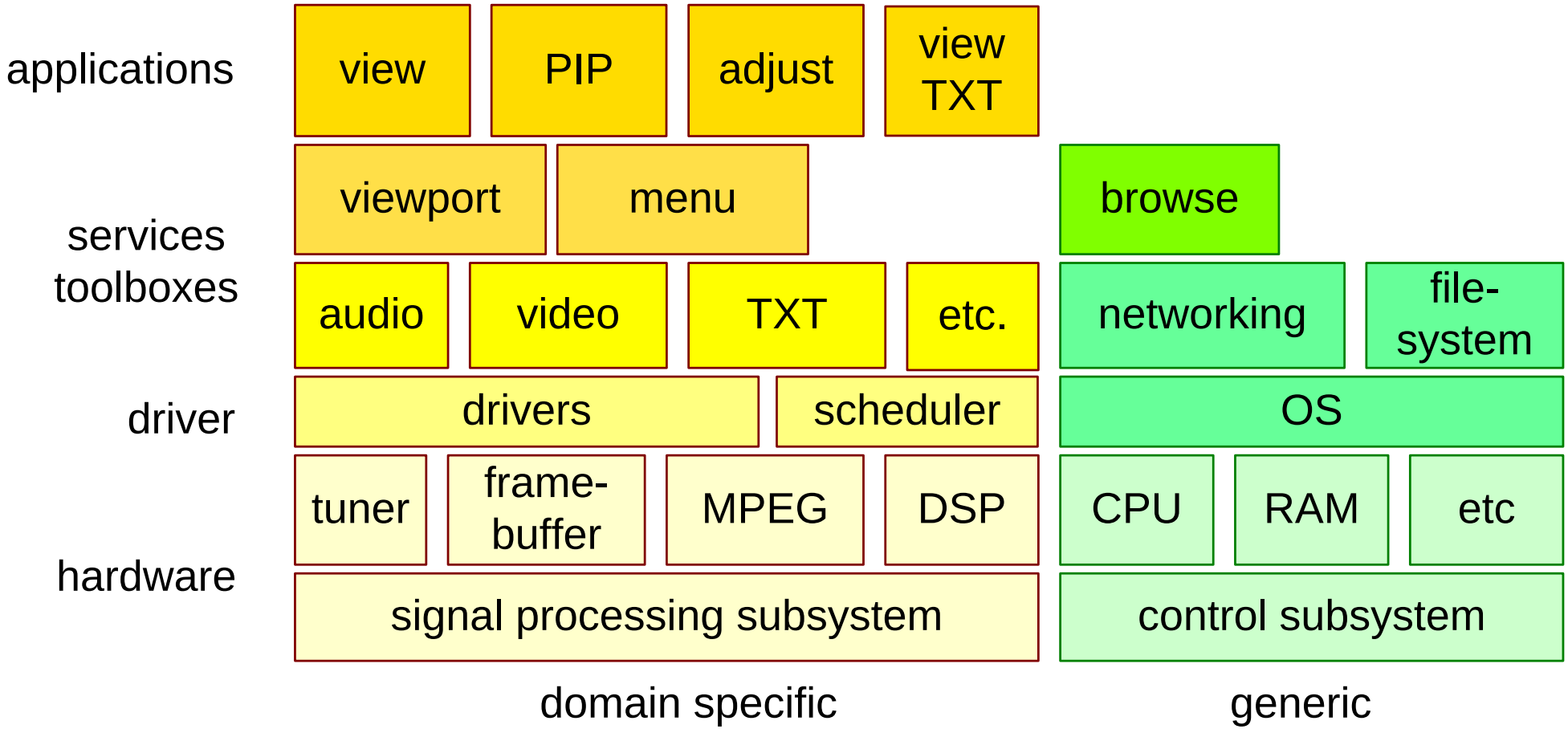
This chapter describes the *Conceptual* view and the *Realization* view. Both views are supported by a set of submethods to describe multi-disciplinary design, for example several decompositions and models are provided.



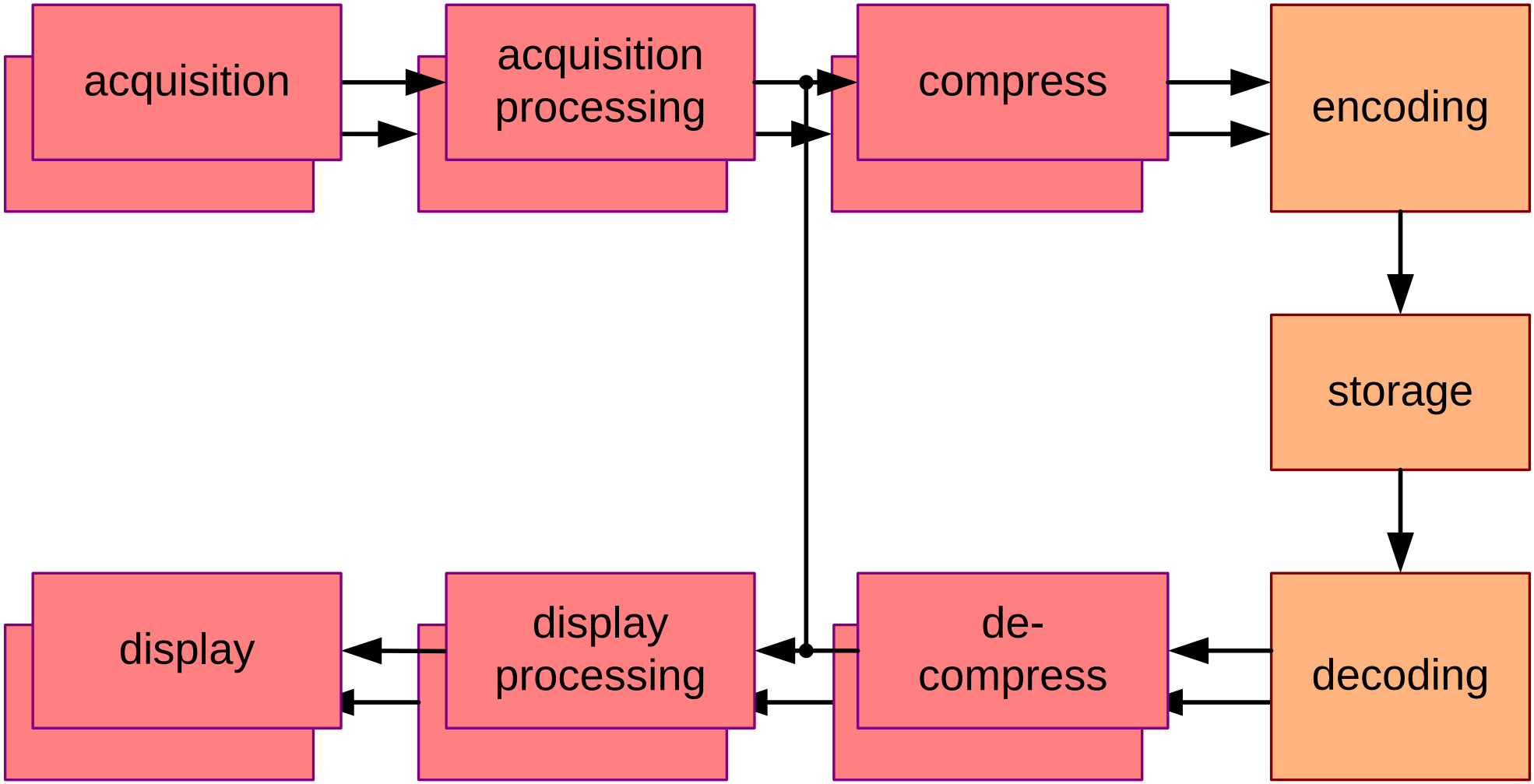
September 1, 2020
status: finished
version: 1.5

logo
TBD

# Construction Decomposition



# Functional Decomposition



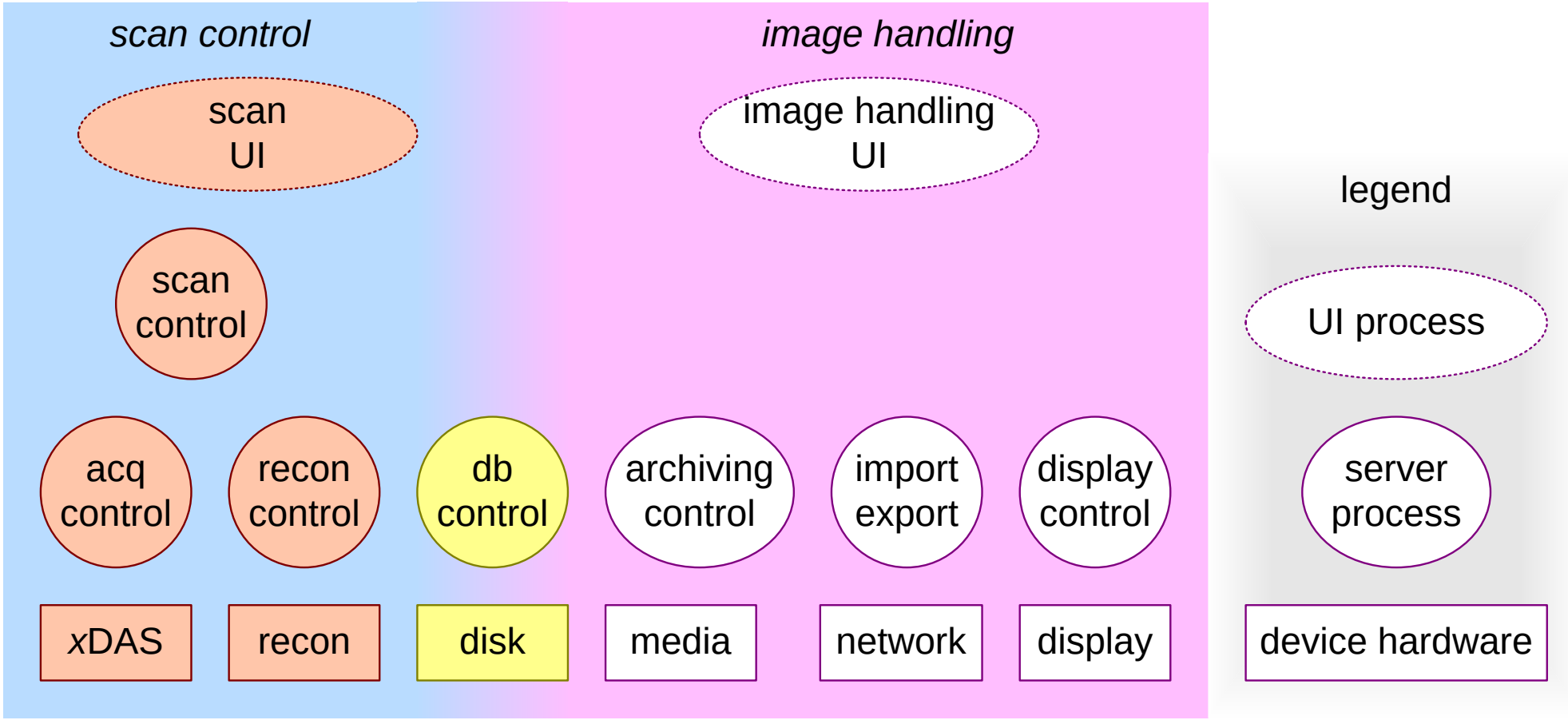
# Question Generator

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

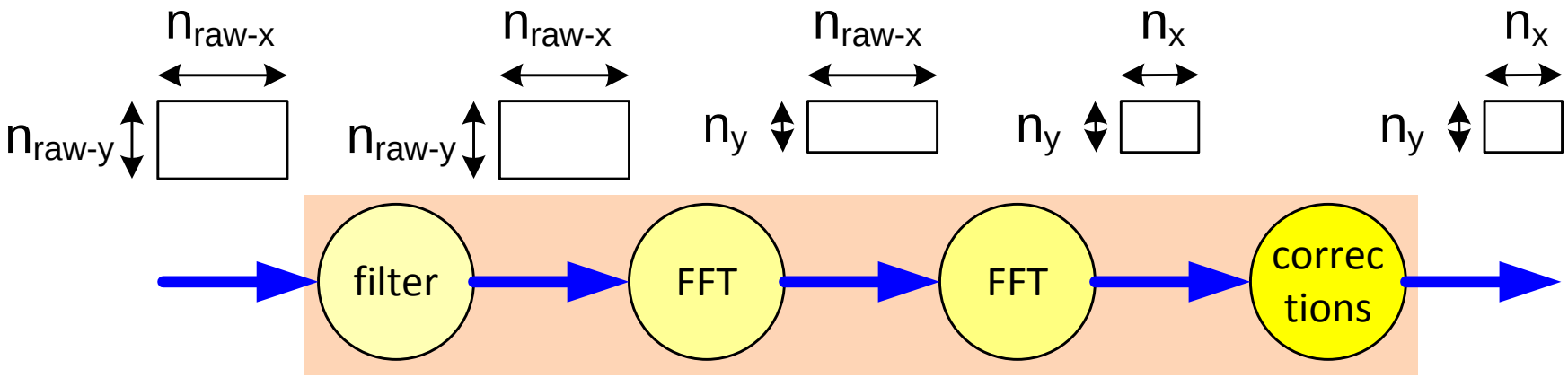
| | |
|---|---|
| characteristics | SNR accuracy **memory usage** processing latency … |
| components | import server **user interface** print server database server export server … |
| functions | **query DB** render film play movie next brightness … |

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

# Process Decomposition



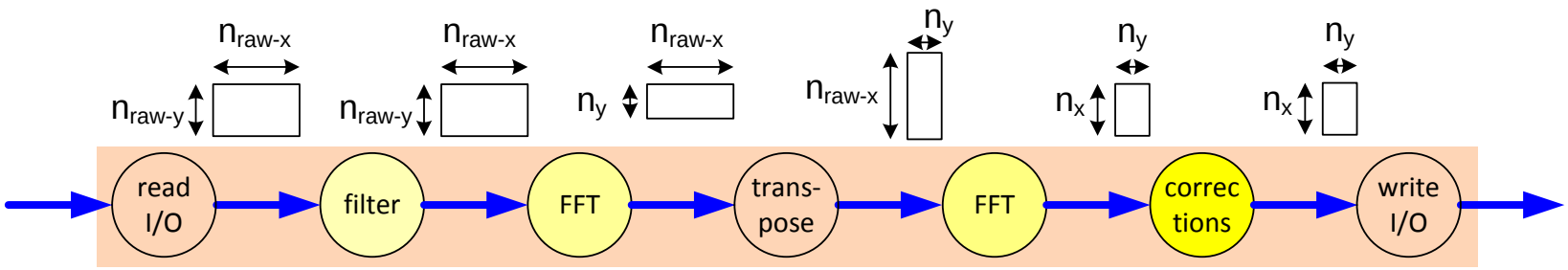
# Conceptual Performance Model




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

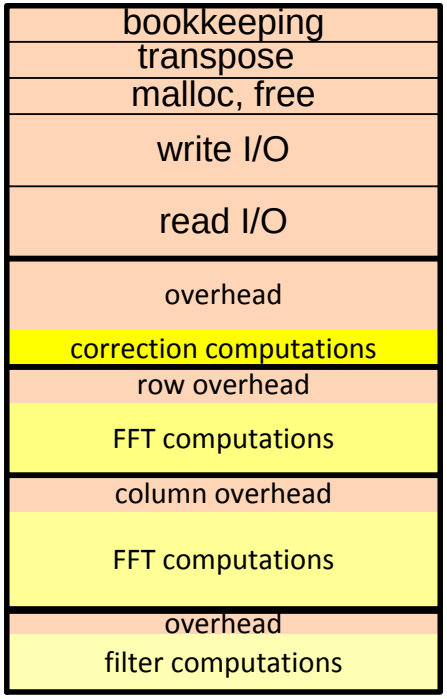
$$t_{fft}(n) = c_{fft} * n * \log(n)$$

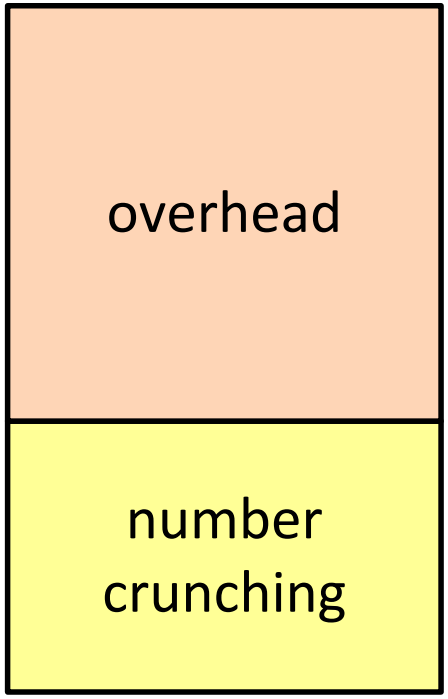
# Model After More Detailed Performance Analysis




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

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




focus on overhead reduction

is more important

than faster algorithms

this is not an excuse for sloppy algorithms

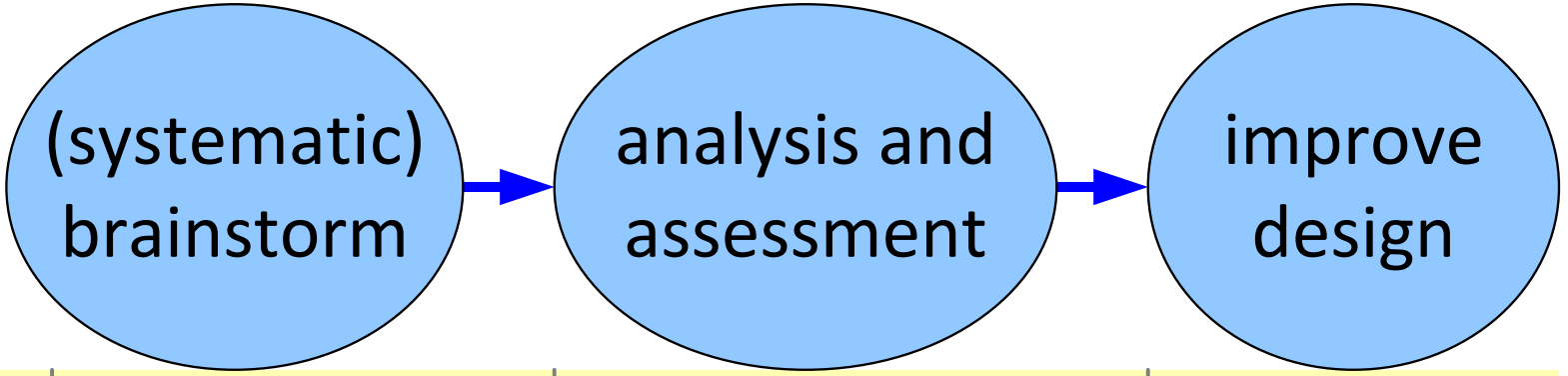
# Micro Benchmarks

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

# Budget Approach

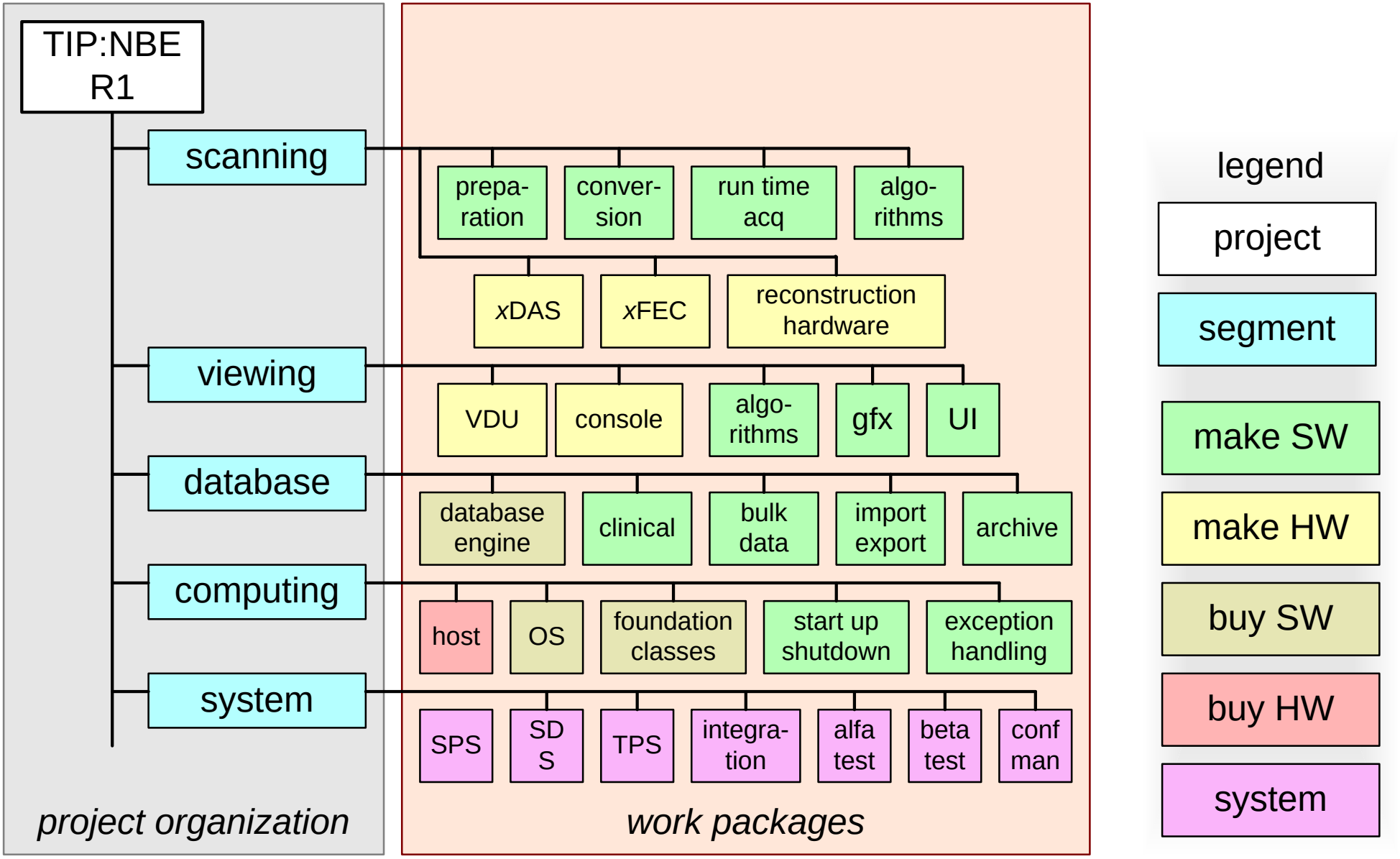
| step | example |
|---|---|
| 1A measure old systems | micro-benchmarks, aggregated functions, applications |
| 1B model the performance starting with old systems | flow model and analytical model |
| 1C determine requirements for new system | response time or throughput |
| 2 make a design for the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

# Safety, Reliability and Security Analysis

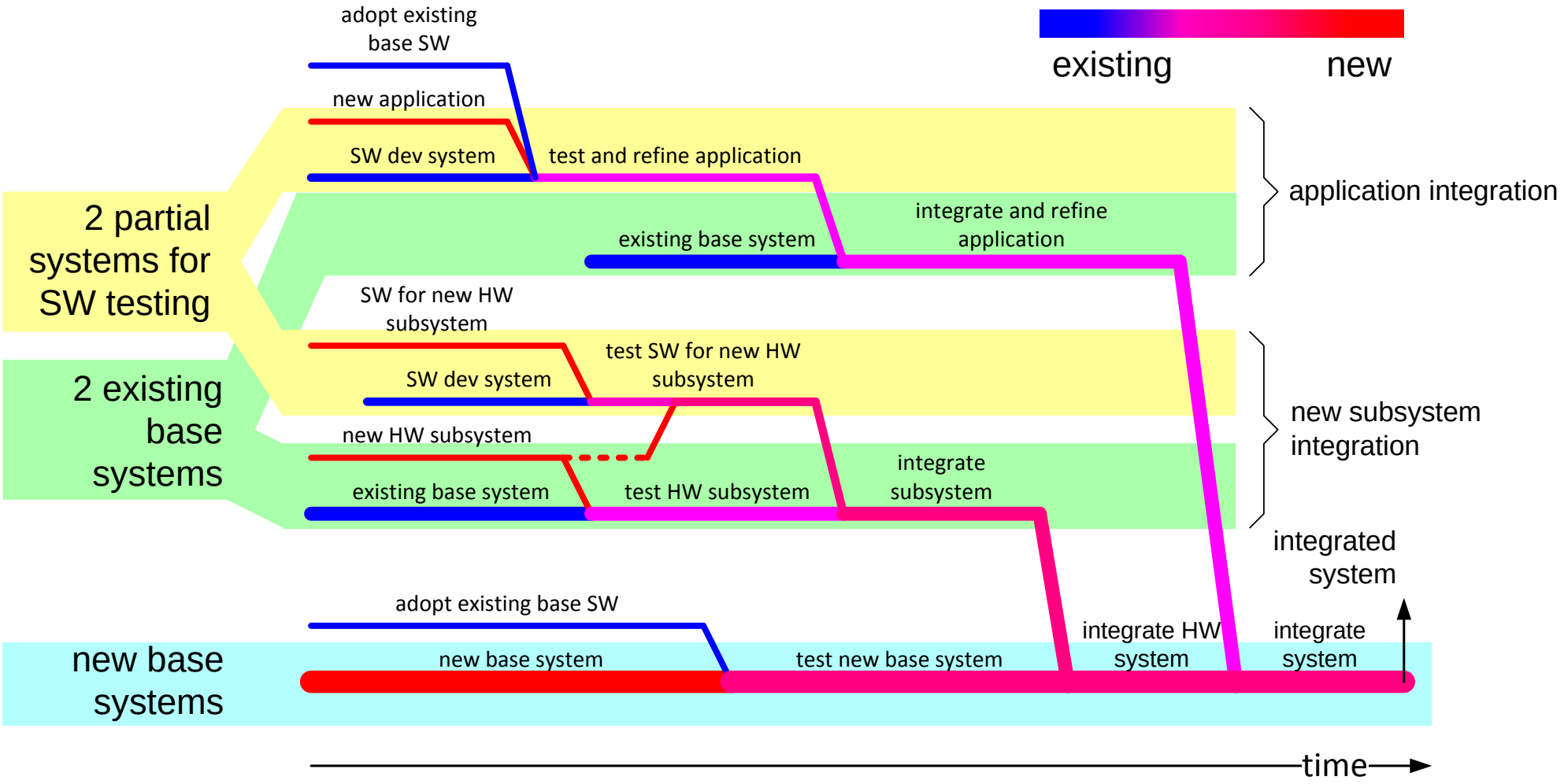



| | | | |
|---|---|---|---|
| safety<br>hazard analysis | potential hazards | probability<br>severity | measures |
| reliability<br>FMEA | failure modes | effects | measures |
| security | vulnerability risks | consequences | measures |

# Work Break Down



# Integration Plan



# Overview CR Submethods

| **C**onceptual | **R**ealization |
| --- | --- |
| construction decomposition<br>functional decomposition<br>designing with multiple decompositions<br>execution architecture<br>internal interfaces<br>performance<br>start up<br>shutdown<br>integration plan<br>work breakdown<br>safety<br>reliability<br>security | budget<br>benchmarking<br>performance analysis<br>value and cost<br>safety analysis<br>reliability analysis<br>security analysis<br>granularity determination |