

The conceptual view

by *Gerrit Muller* Embedded Systems Institute

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

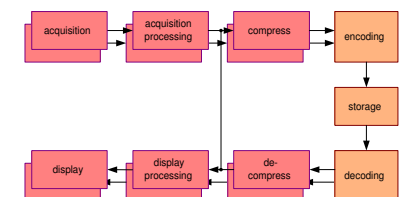
Abstract

The purpose of the conceptual view is described. A number of methods or models is given to use in this view: construction decomposition, functional decomposition, class or object decomposition, other decompositions (power, resources, recycling, maintenance, project management, cost, ...), and related models (performance, behavior, cost, ...); allocation, dependency structure; identify the infrastructure (factoring out shareable implementations), classify the technology in *core*, *key* and *base* technology; integrating concepts (start up, shutdown, safety, exception handling, persistency, resource management,...).

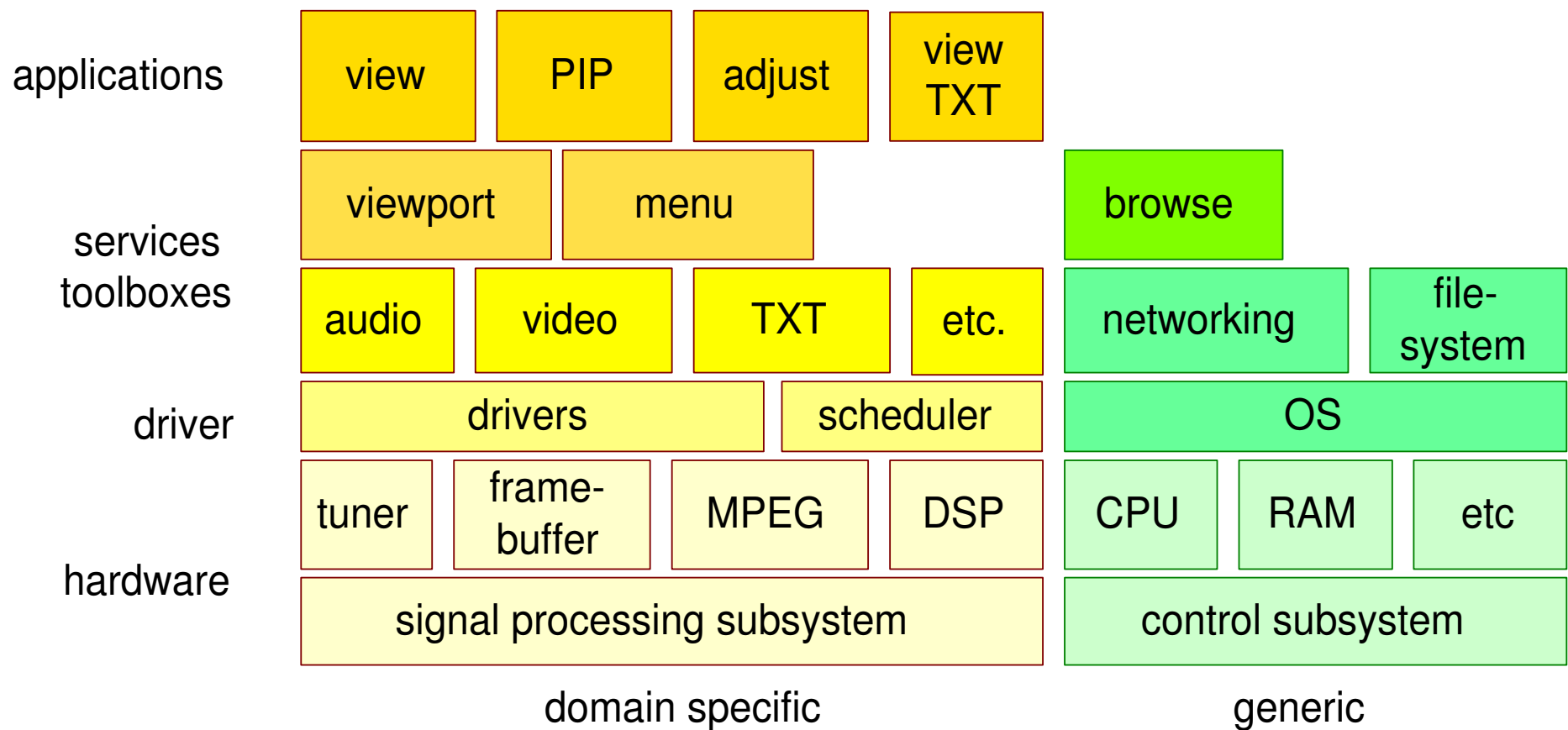
Distribution

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.

July 1, 2011
status: preliminary
draft
version: 0.7



Example construction decomposition simple TV



Characterization of the construction decomposition

management of design

SW example

HW example

unit of
creation
storage
update

file

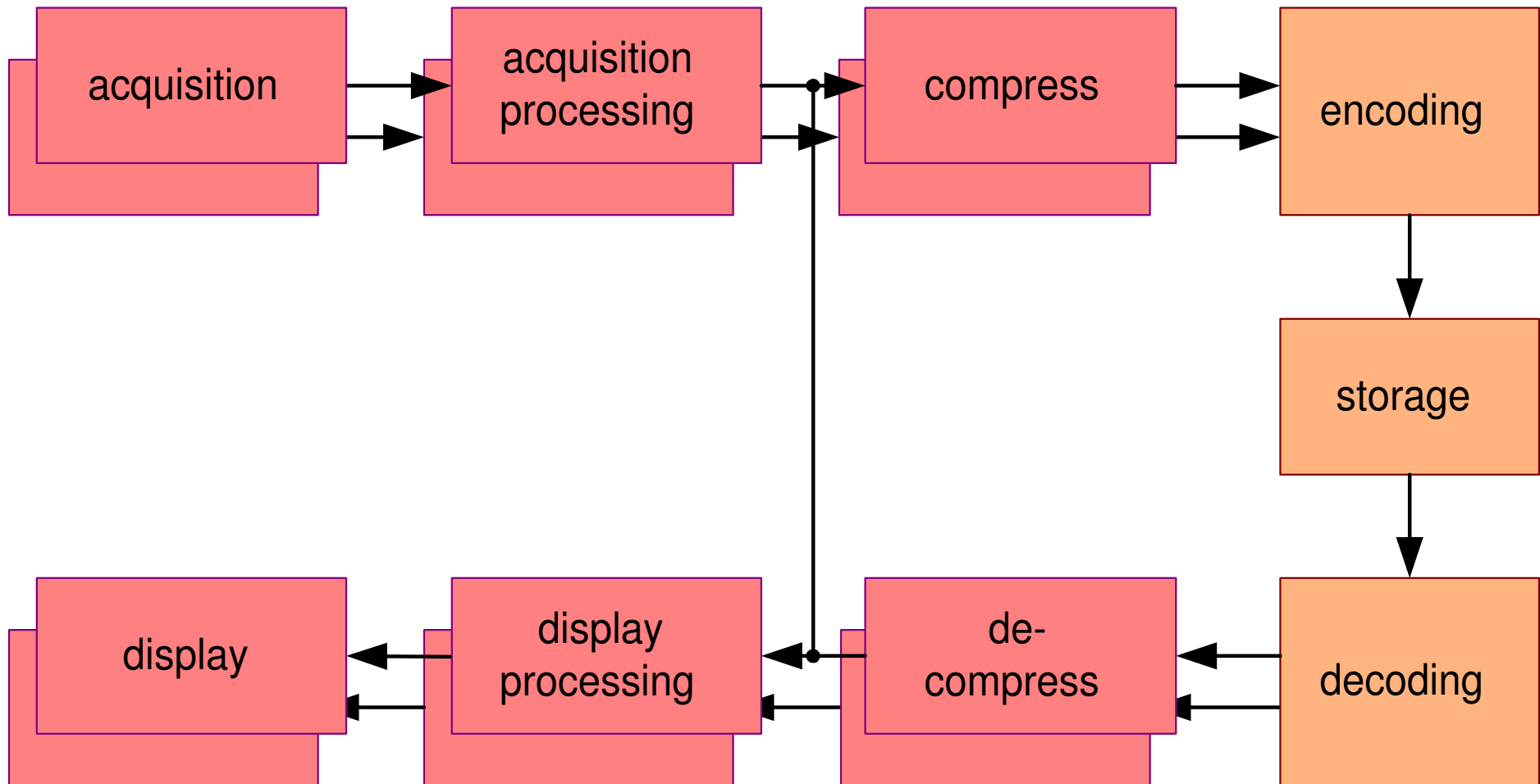
PCB
IP cells
IP core

unit of aggregation for
organisation
test
release

package
module

box
IP core
IC

Example functional decomposition camera type device



How ;
what is the flow of internal activities
to realise external functionality ?

some keywords:

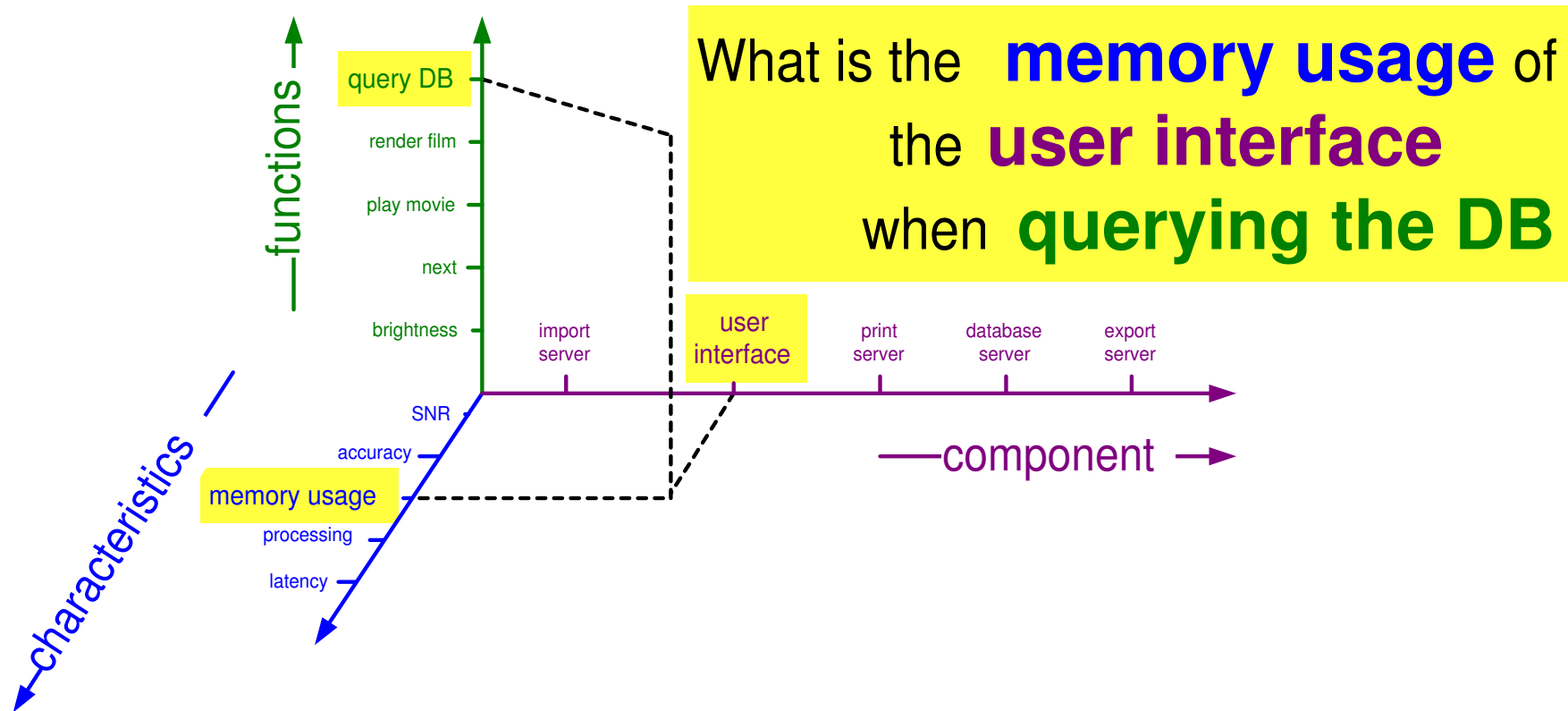
activities
transformation
input output

data flow
control flow

multiple functional decompositions
are possible and valuable!

Question generator for multiple decompositions

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



Critical for system performance

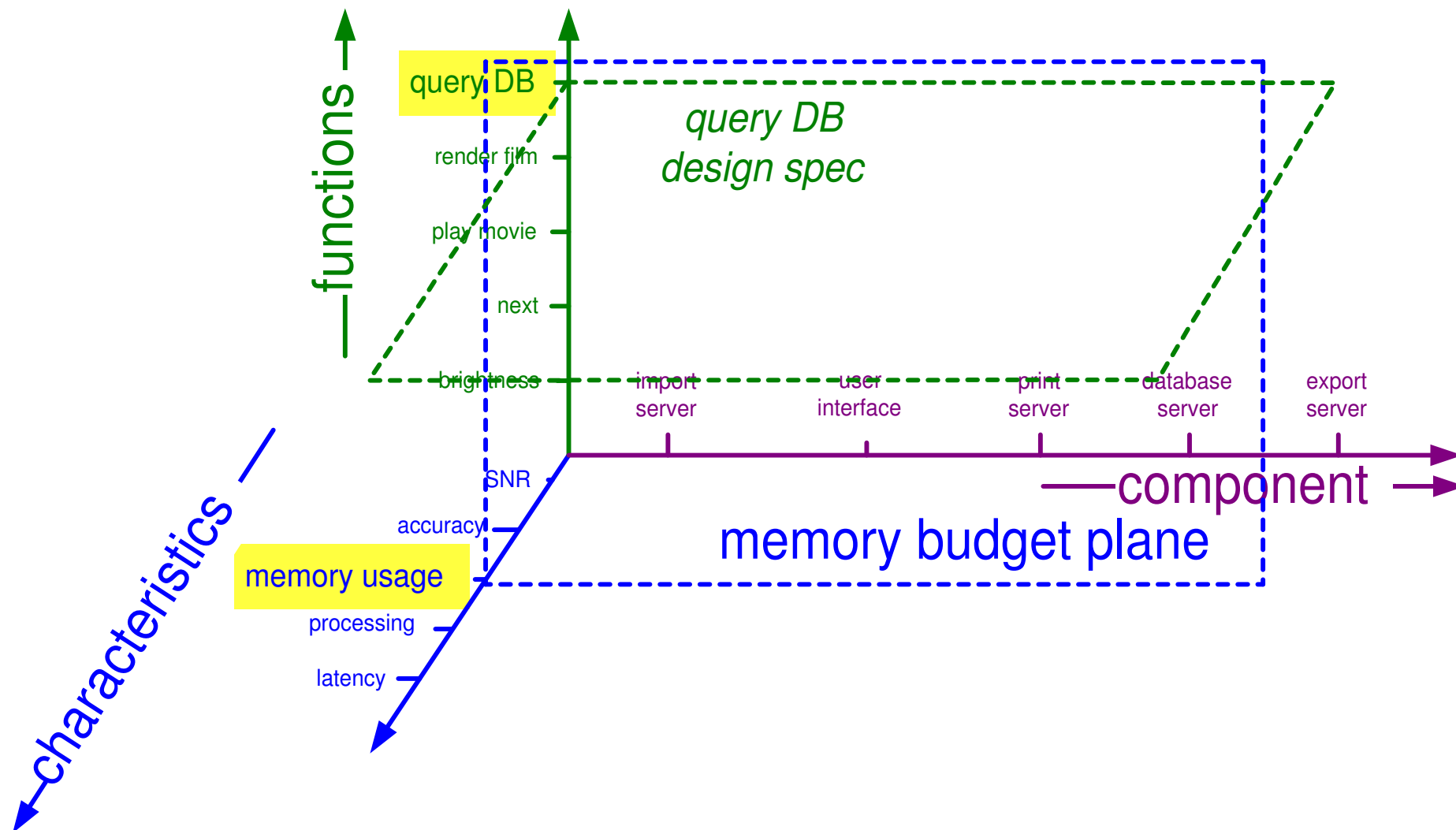
Risk planning wise

Least robust part of the design

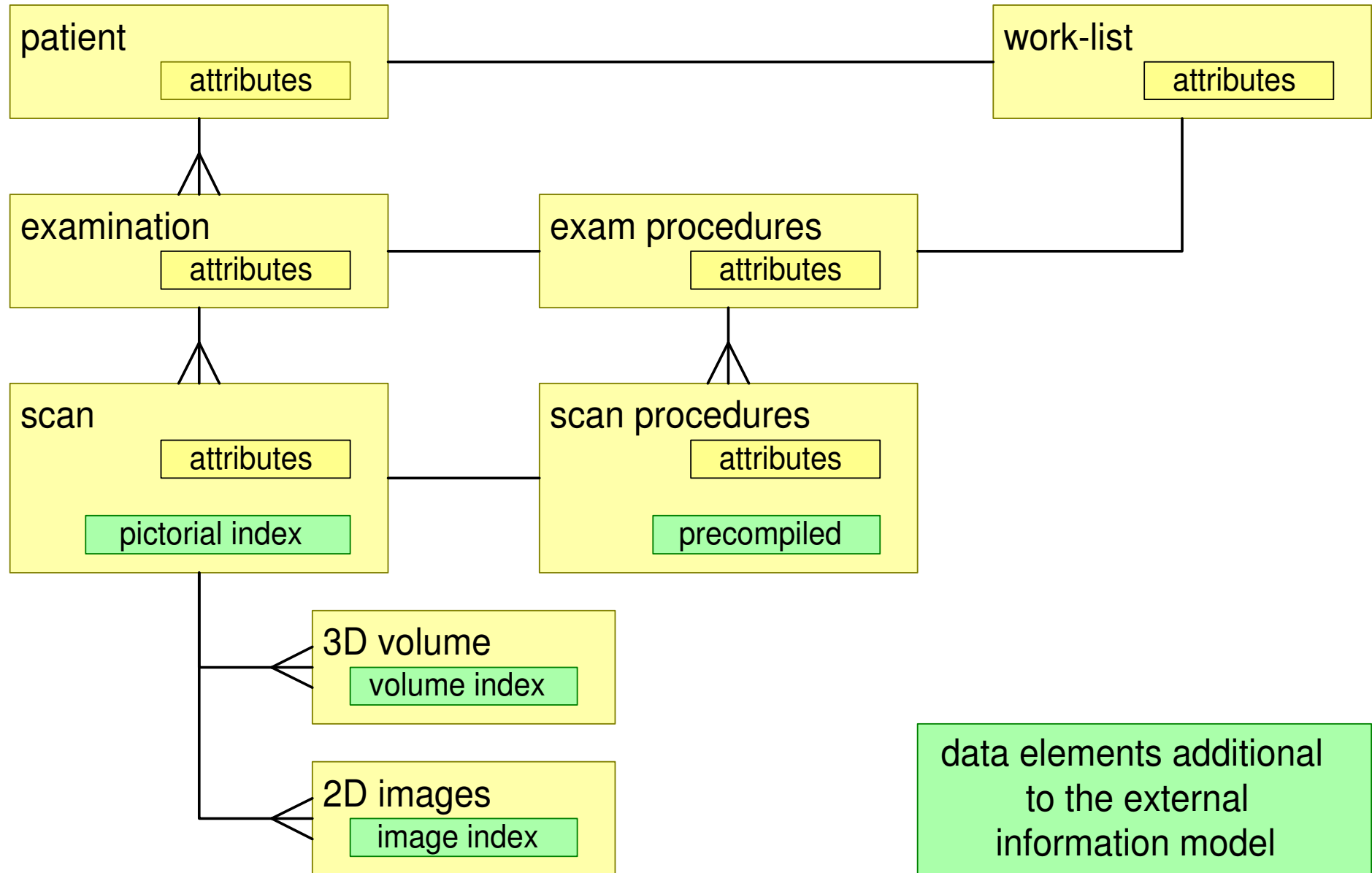
Suspect part of the design

- experience based
- person based

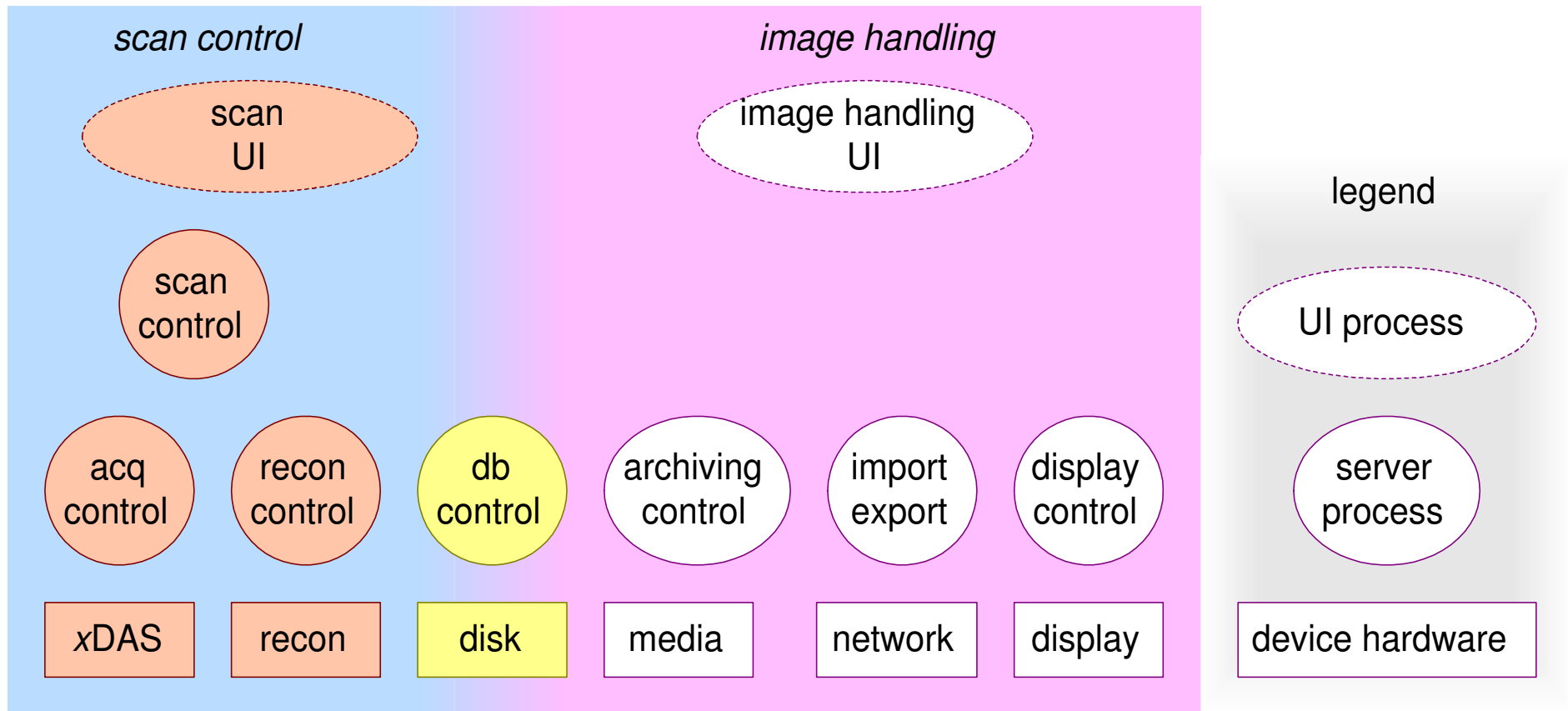
Addressing planes or lines



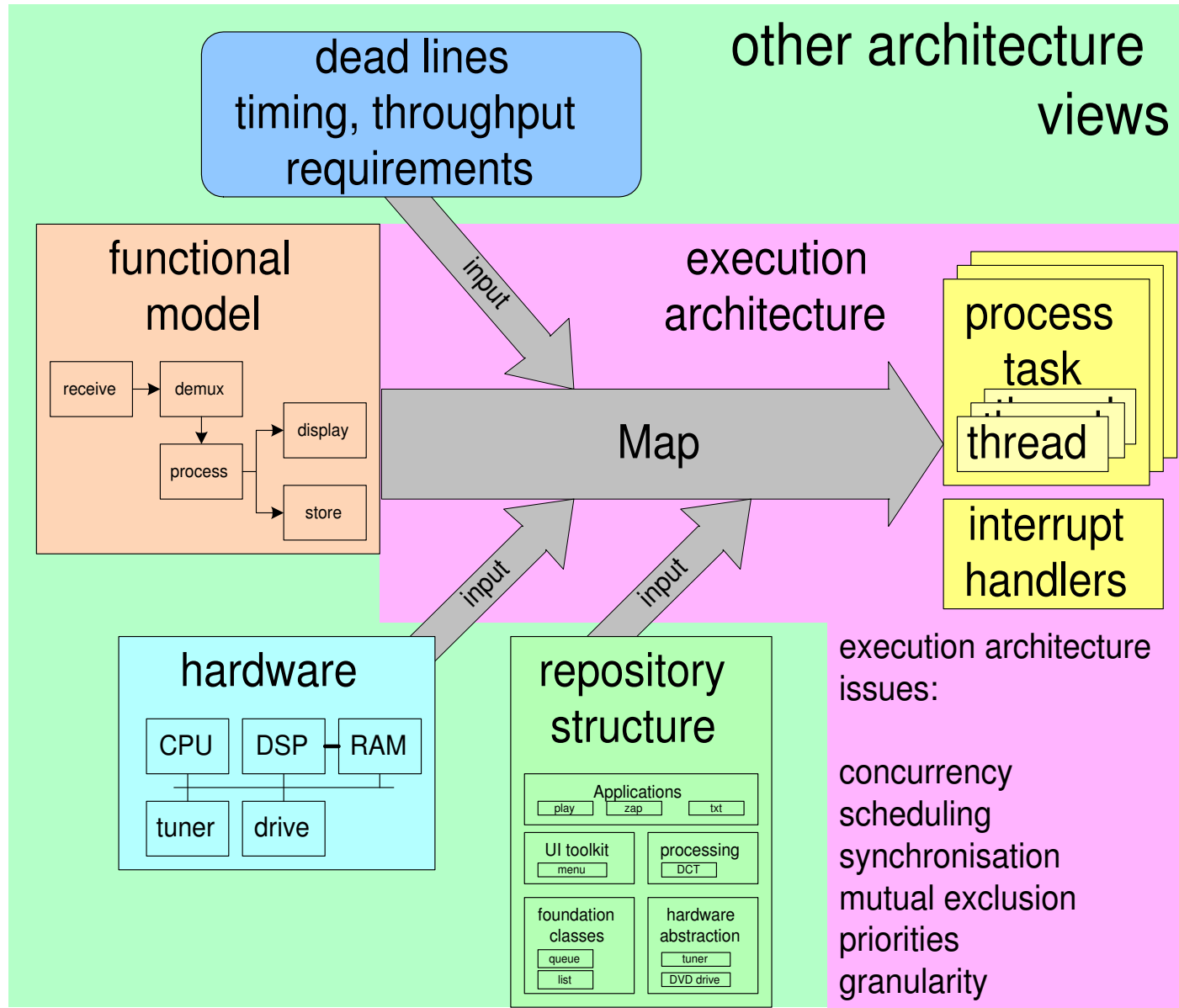
Example partial internal information model



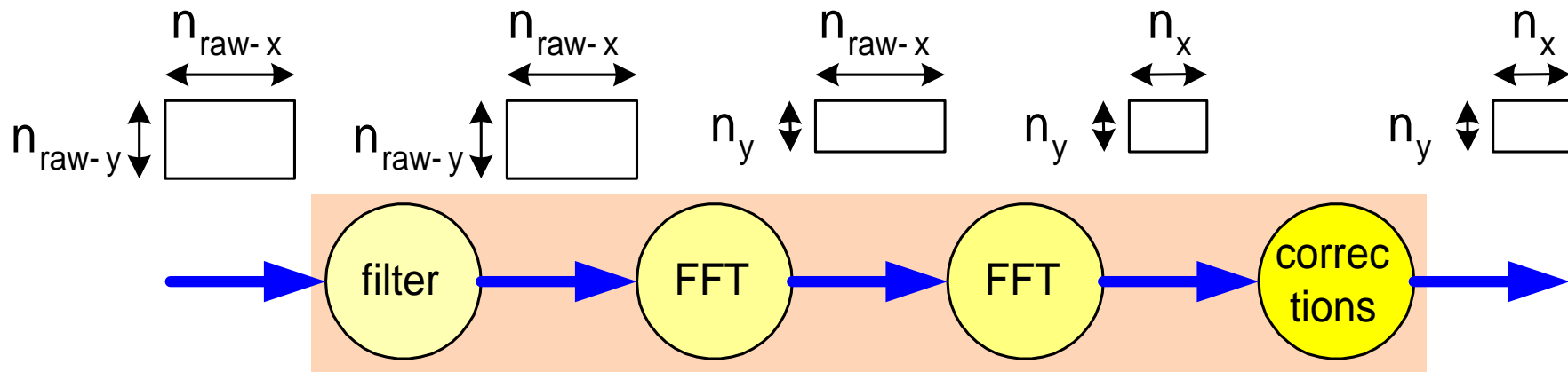
Example process decomposition



Execution architecture



Performance Model



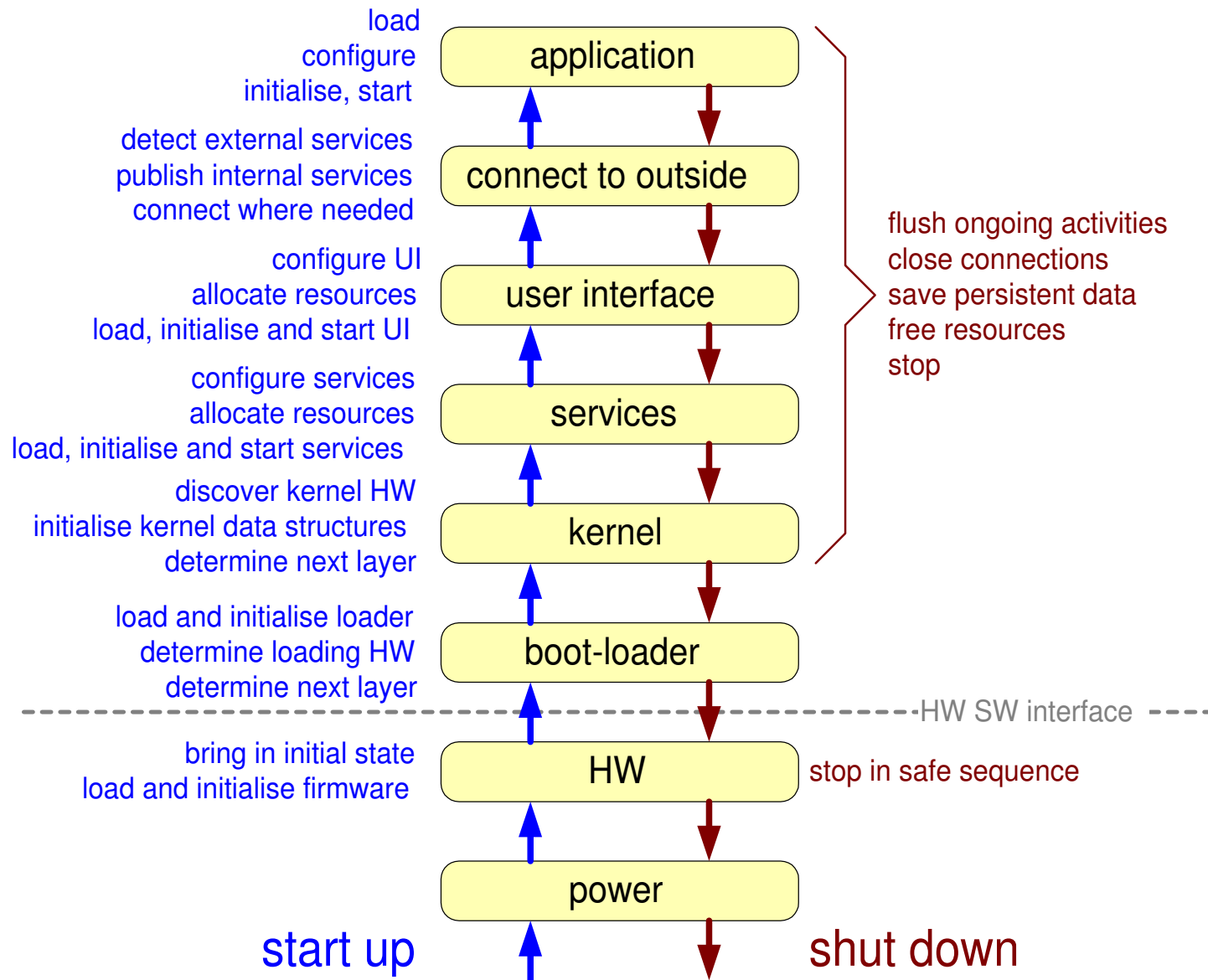
$$t_{\text{recon}} = t_{\text{filter}}(n_{\text{raw-x}}, n_{\text{raw-y}}) + n_{\text{raw-x}} * (t_{\text{fft}}(n_{\text{raw-y}}) + t_{\text{col-overhead}}) + n_y * (t_{\text{fft}}(n_{\text{raw-x}}) + t_{\text{row-overhead}}) + t_{\text{corrections}}(n_x, n_y) + t_{\text{control-overhead}}$$

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

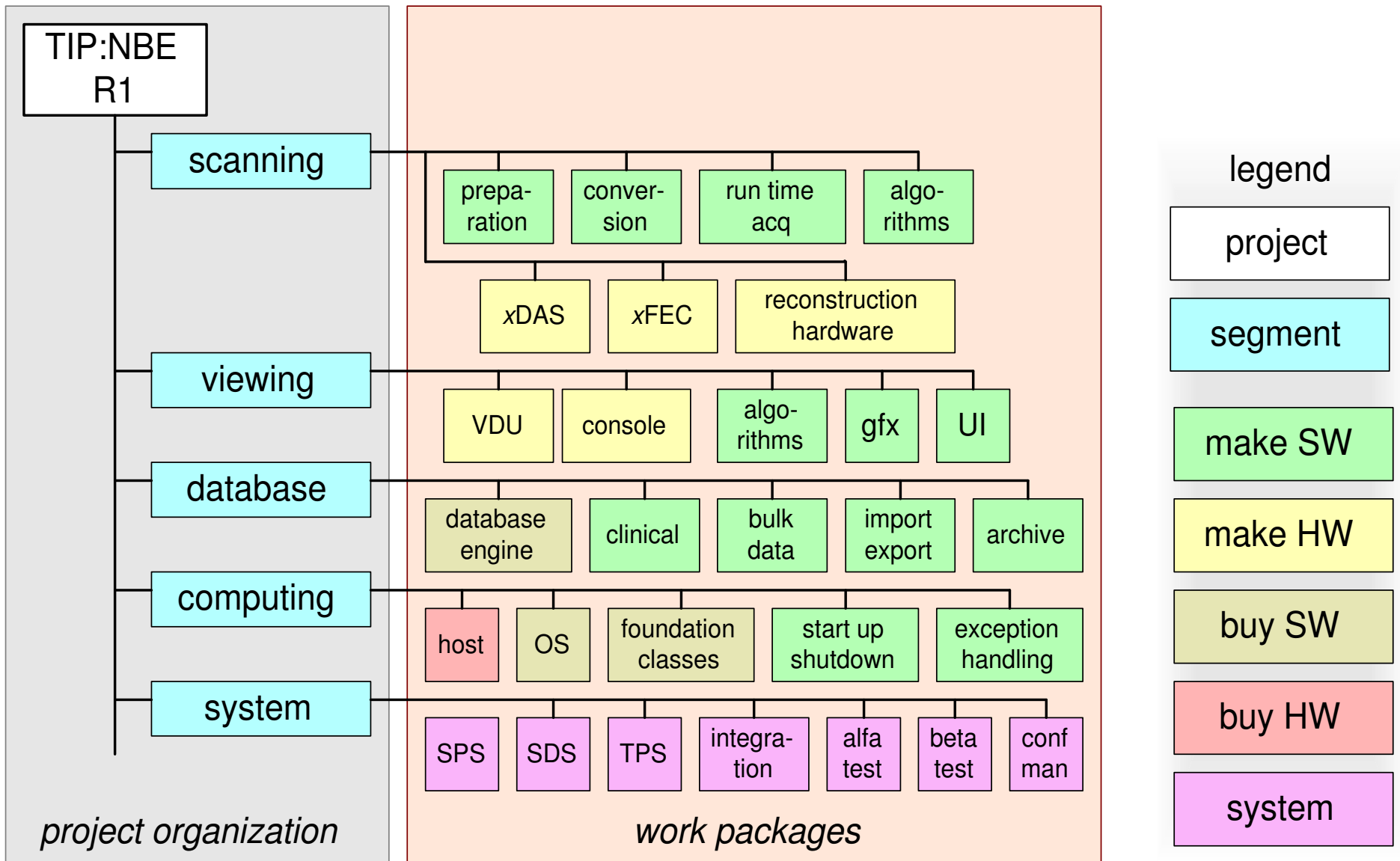
Safety, Reliability and Security concepts

- containment (limit failure consequences to well defined scope)
- graceful degradation (system parts not affected by failure continue operation)
- dead man switch (human activity required for operation)
- interlock (operation only if hardware conditions are fulfilled)
- detection and tracing of failures
- black box (log) for post mortem analysis
- redundancy

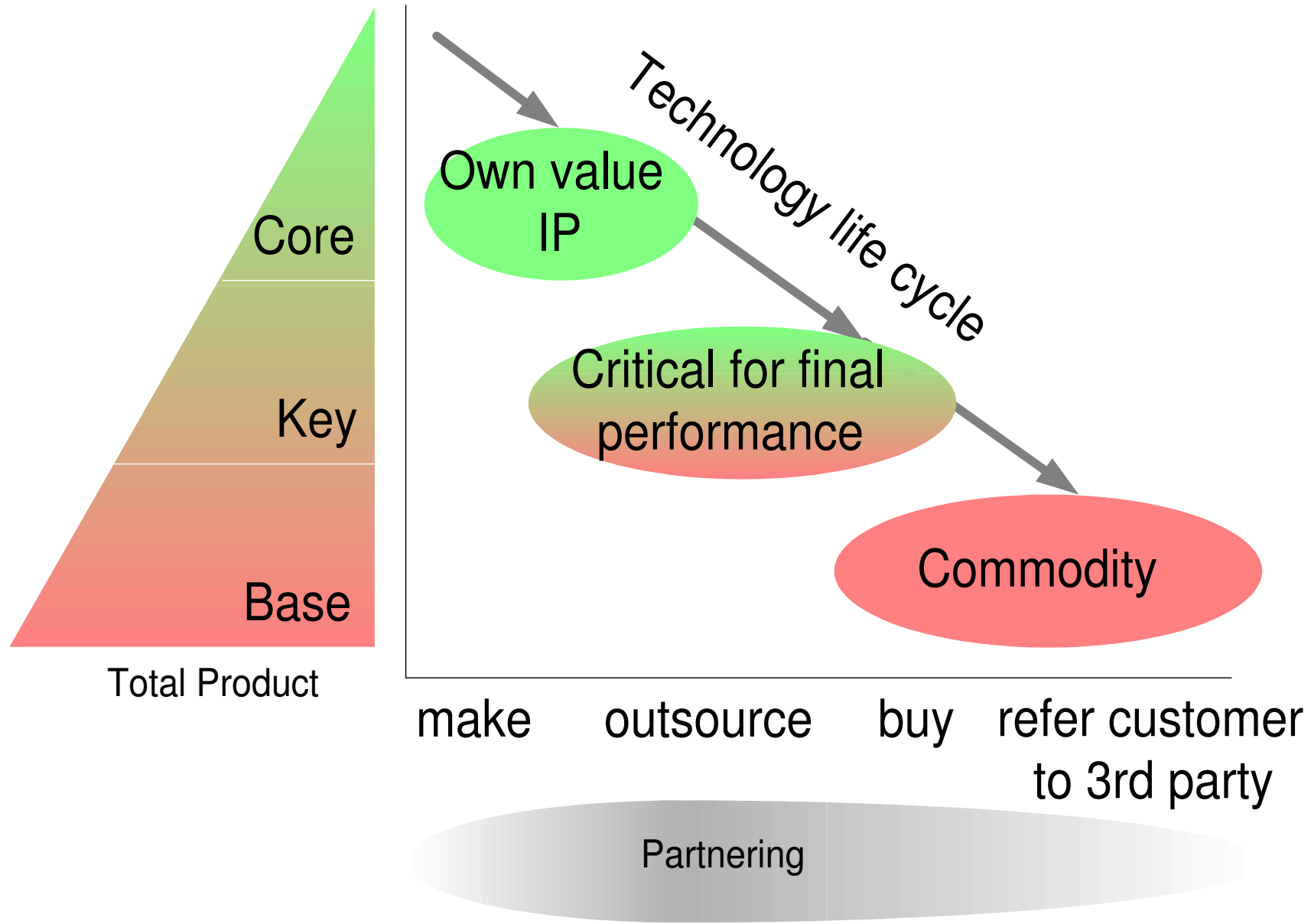
Simplified start up sequence



Example work breakdown



Core, Key or Base technology



Example integration plan

