

How to Create a Manageable Platform Architecture?

by *Gerrit Muller* Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

Abstract

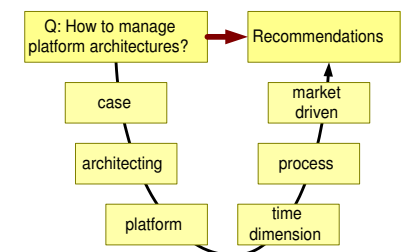
Today's fast pace of the market and the technology development forces the product creators to rethink their development approach. One of the directions is to maximize the return on investments of frequently used functions, for instance by re-use, component based design or by a platform approach. The architecting effort is a key success factor to combine re-use approaches with fast and innovative product creation.

In this presentation we will present a case, discuss the role of the architecture, and elaborate the essential architecture ingredients for a successful platform creation, and evolution, and innovative product creation.

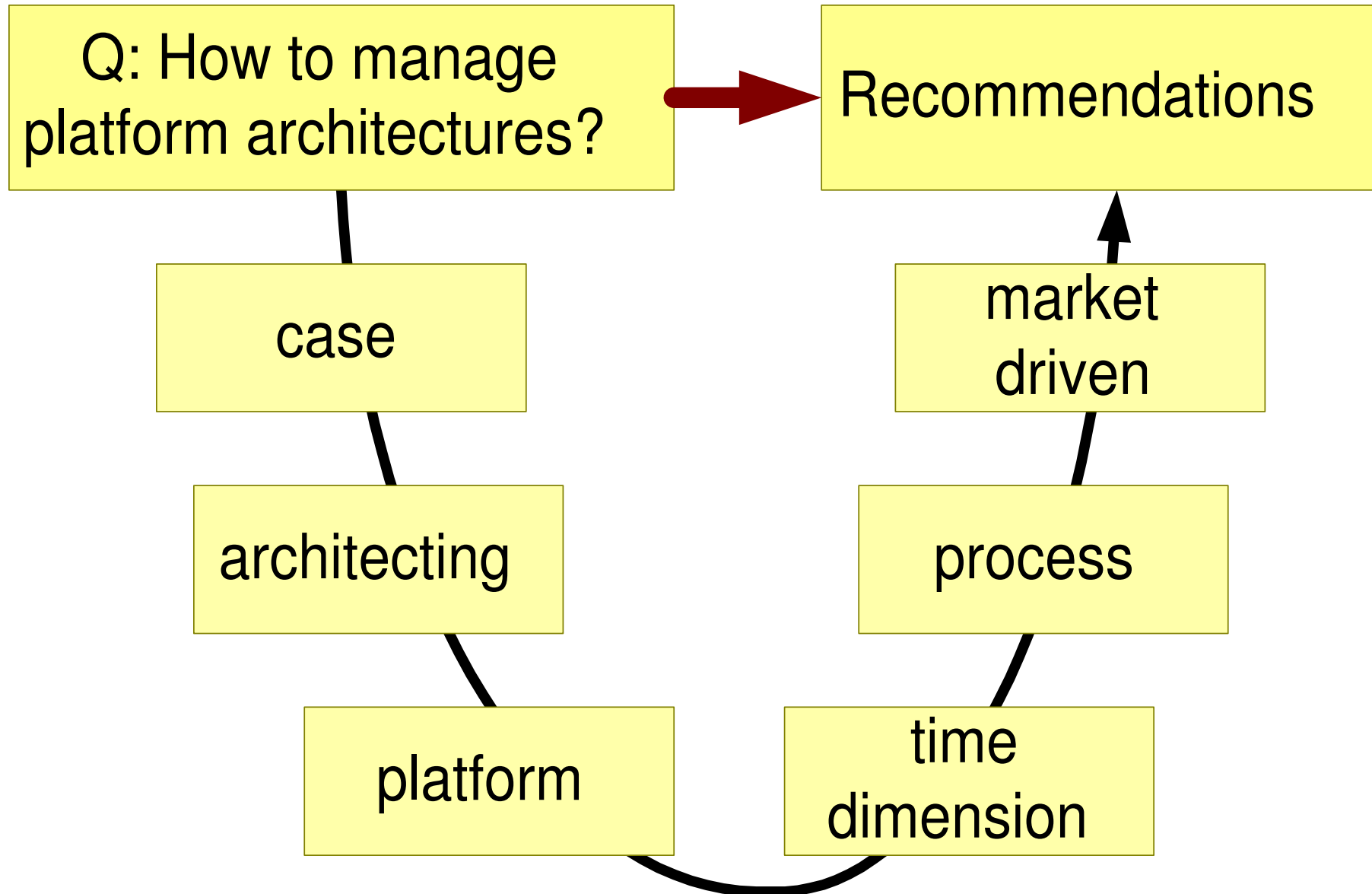
Distribution

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: concept
version: 1.0



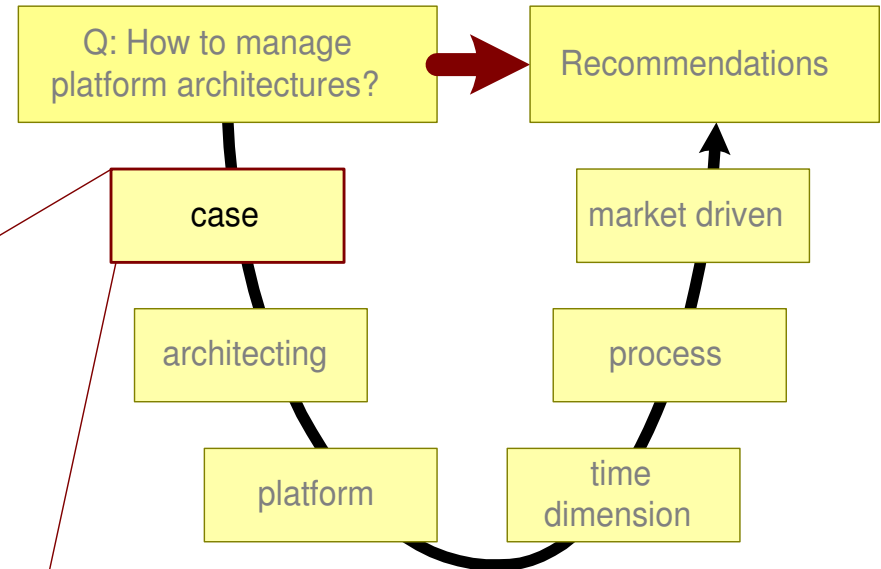
Presentation Outline



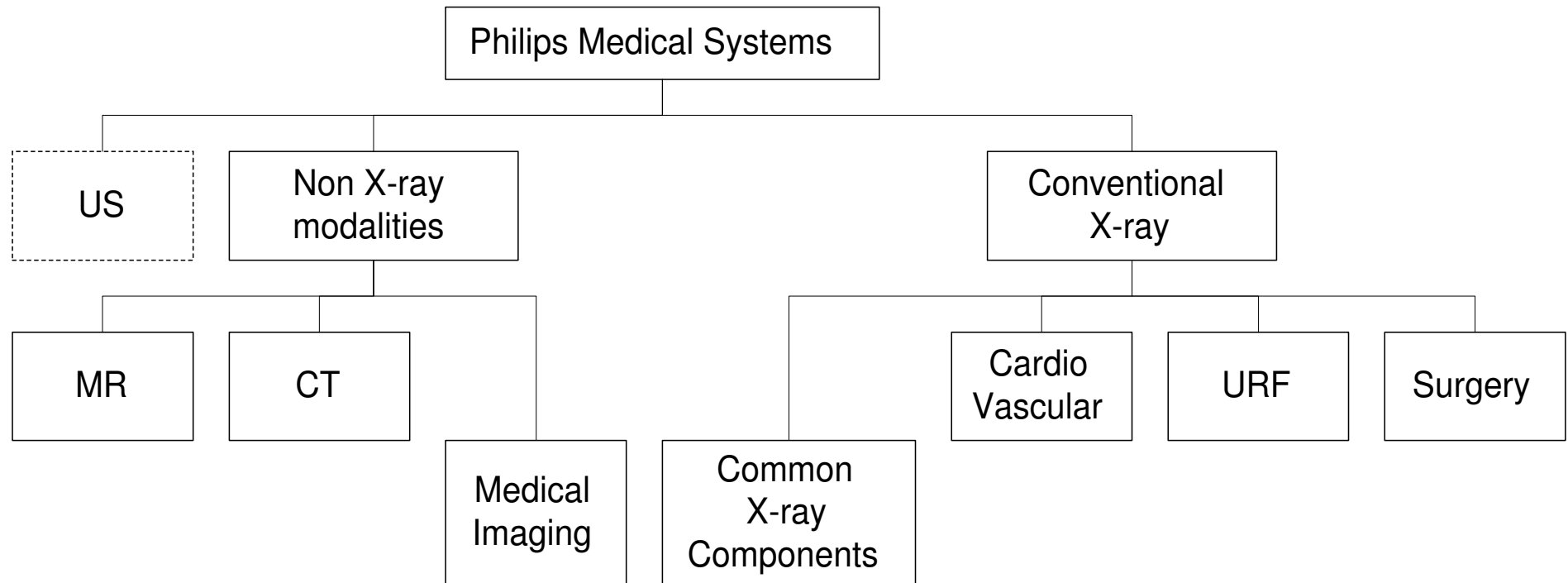
Case Outline

case:

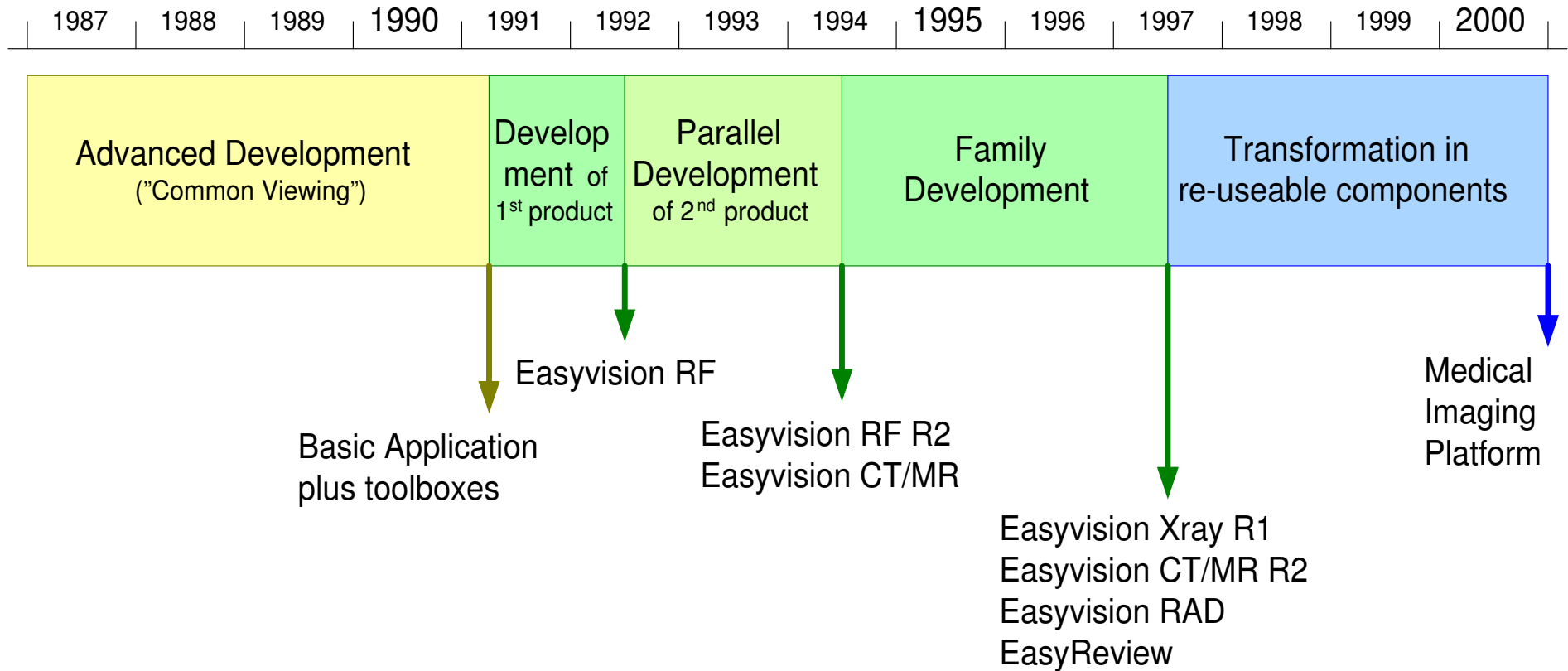
- company overview
- time line
- technology innovations
- 1991, 1992, 1994, and 1996
context, product(s) and design



Philips Medical Systems, schematic organization



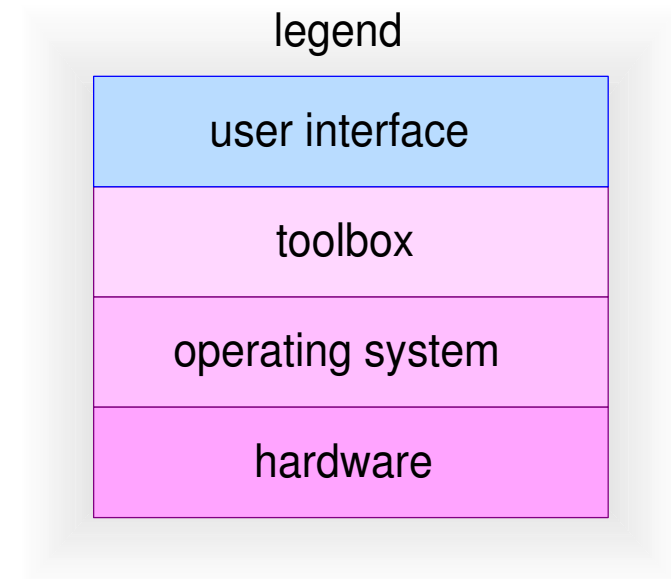
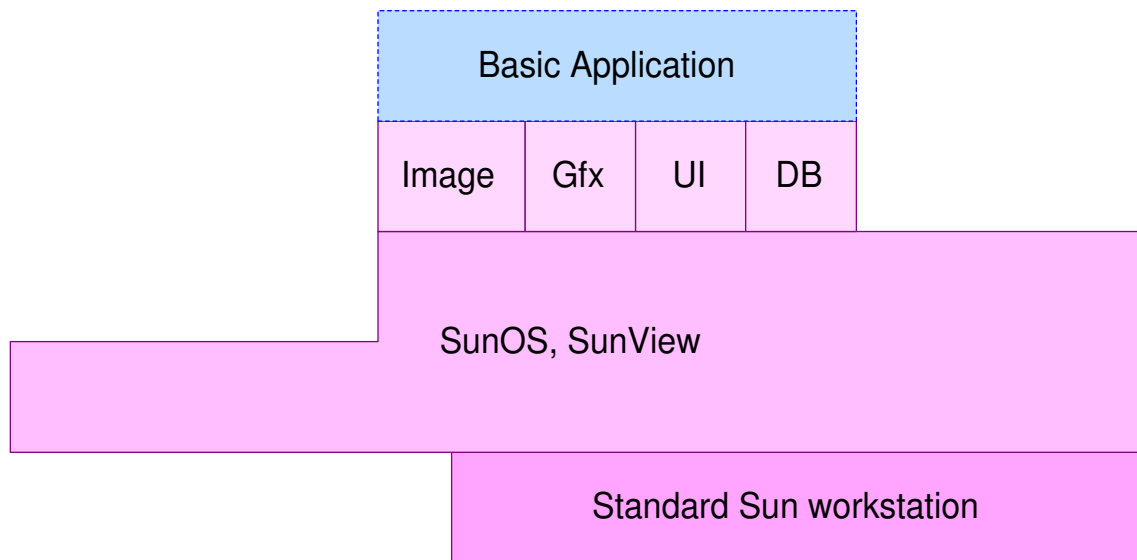
Phases of Medical Imaging



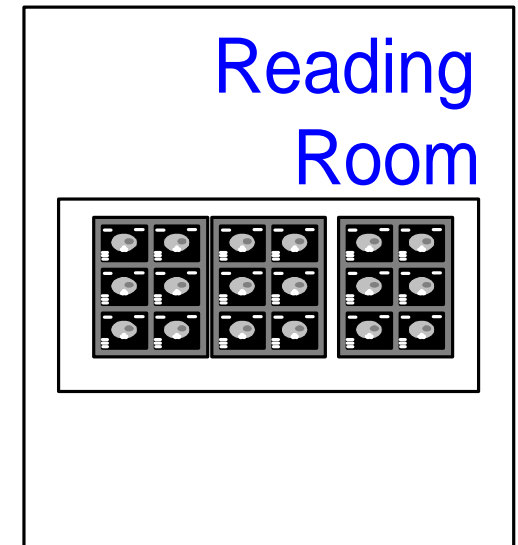
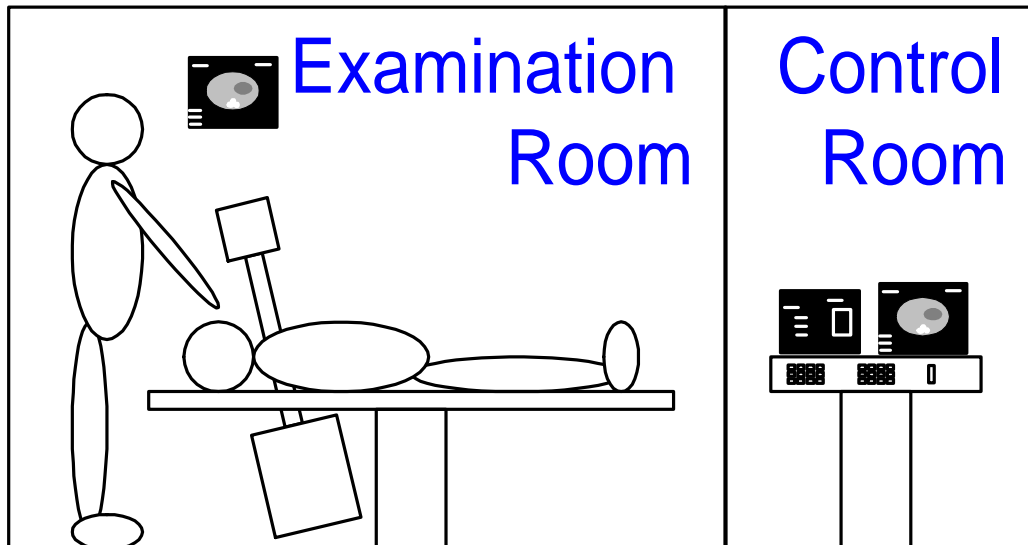
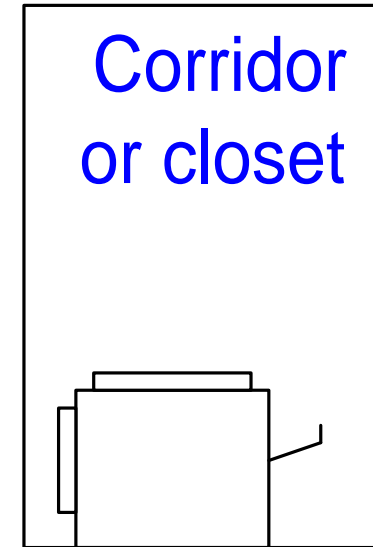
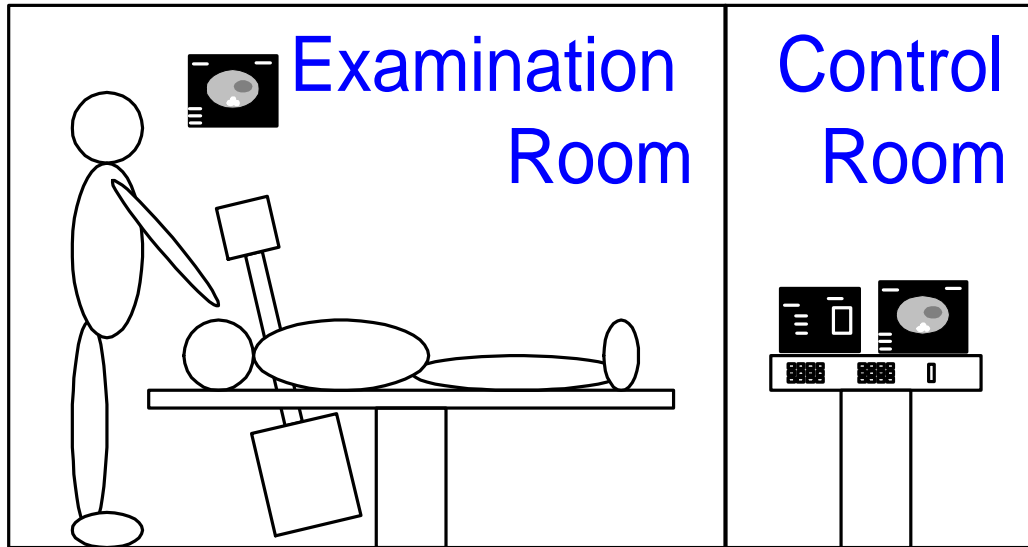
Technology innovations by Common Viewing

- standard UNIX based workstation
- full SW implementation, more flexible
- object oriented design and implementation (Objective-C)
- graphical User Interface, with windows, mouse et cetera
- call back scheduling, fine-grained notification
- data base engine, fast, reliable and robust
- extensive set of toolboxes
- property based configuration
- multiple co-ordinate spaces

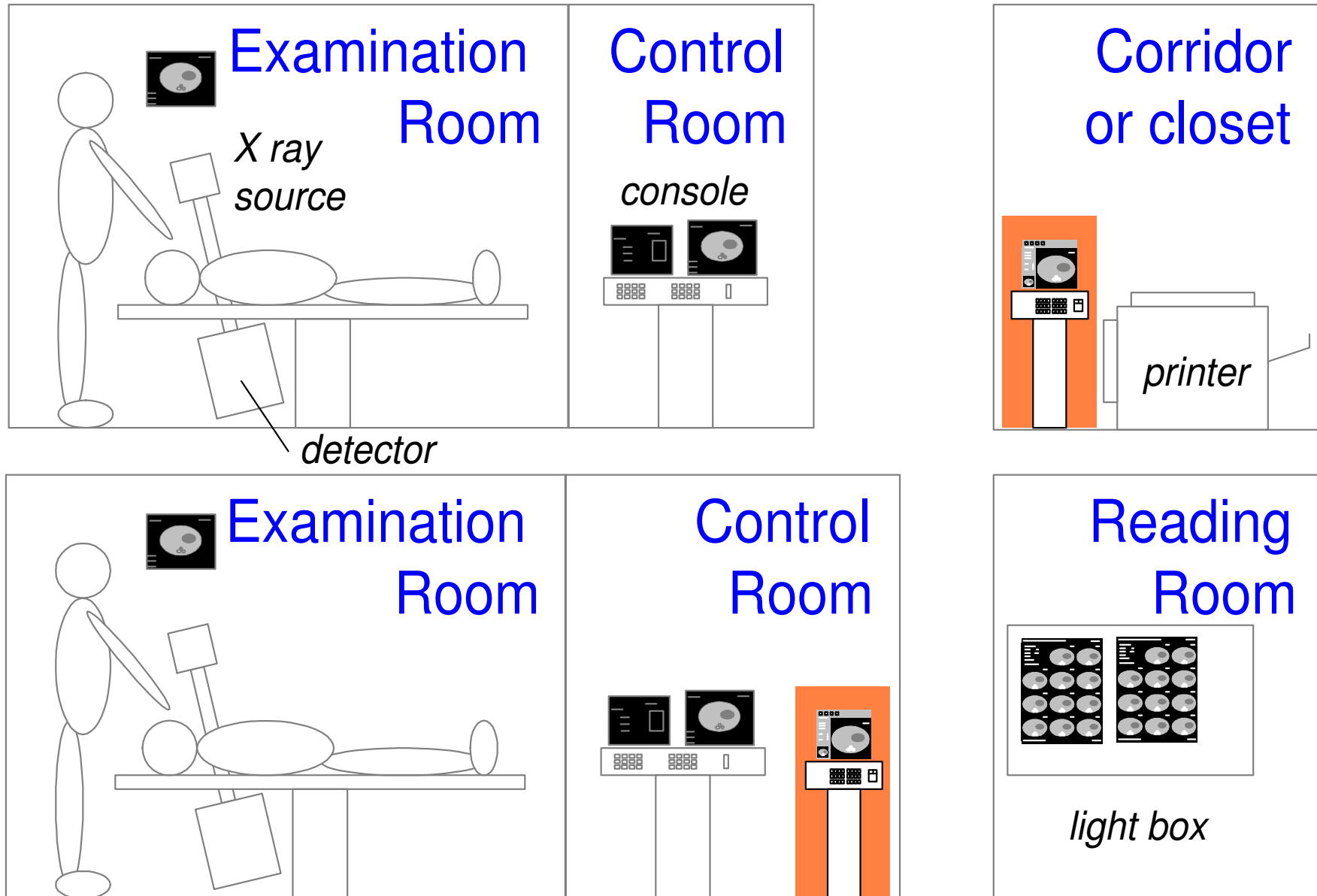
Idealized layers September 1991



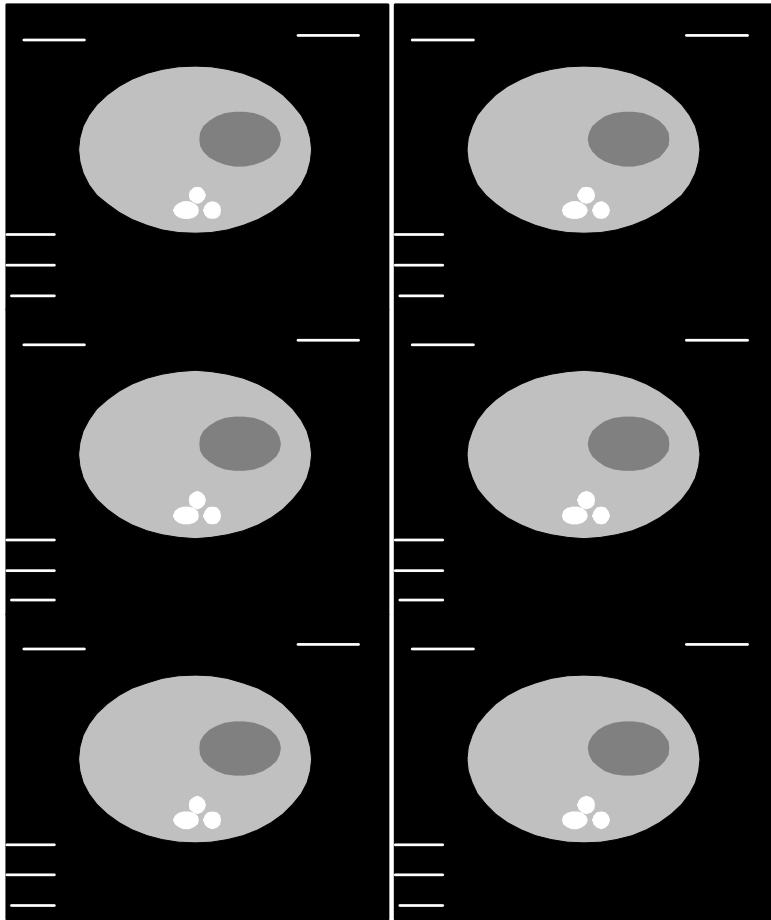
X-ray rooms from examination to reading around 1990



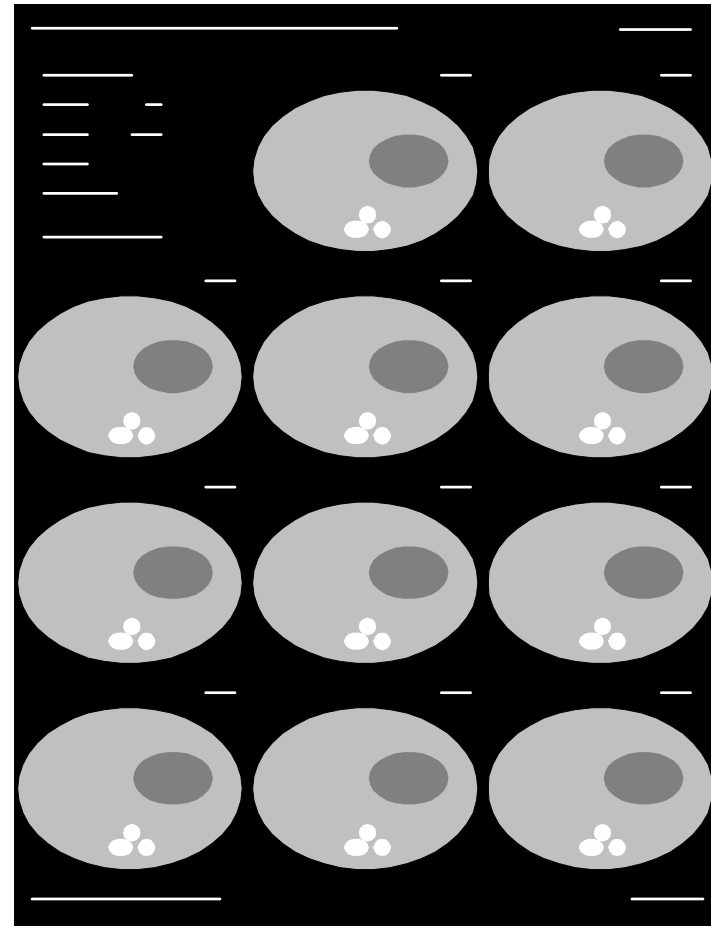
X-ray rooms with Medical Imaging applied as printserver



Comparison *screen copy* vs *optimized* film



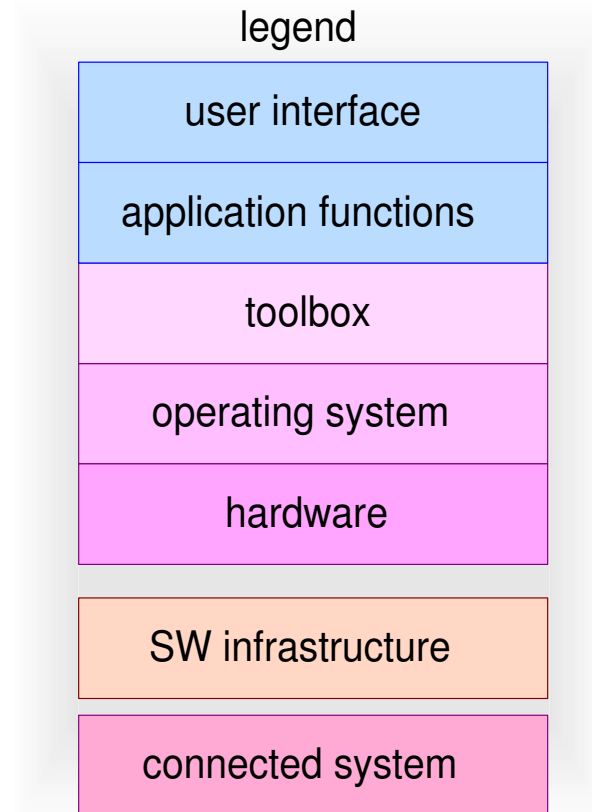
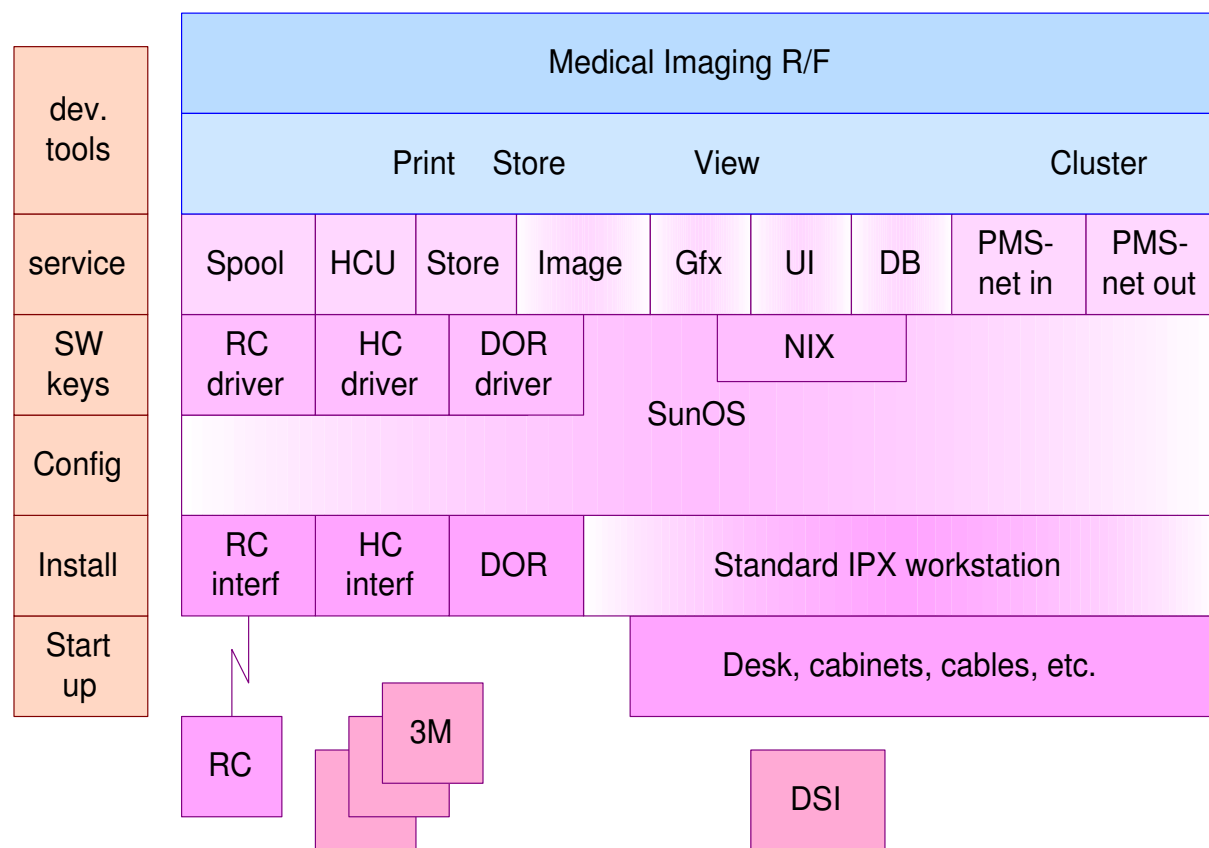
old: screen copy



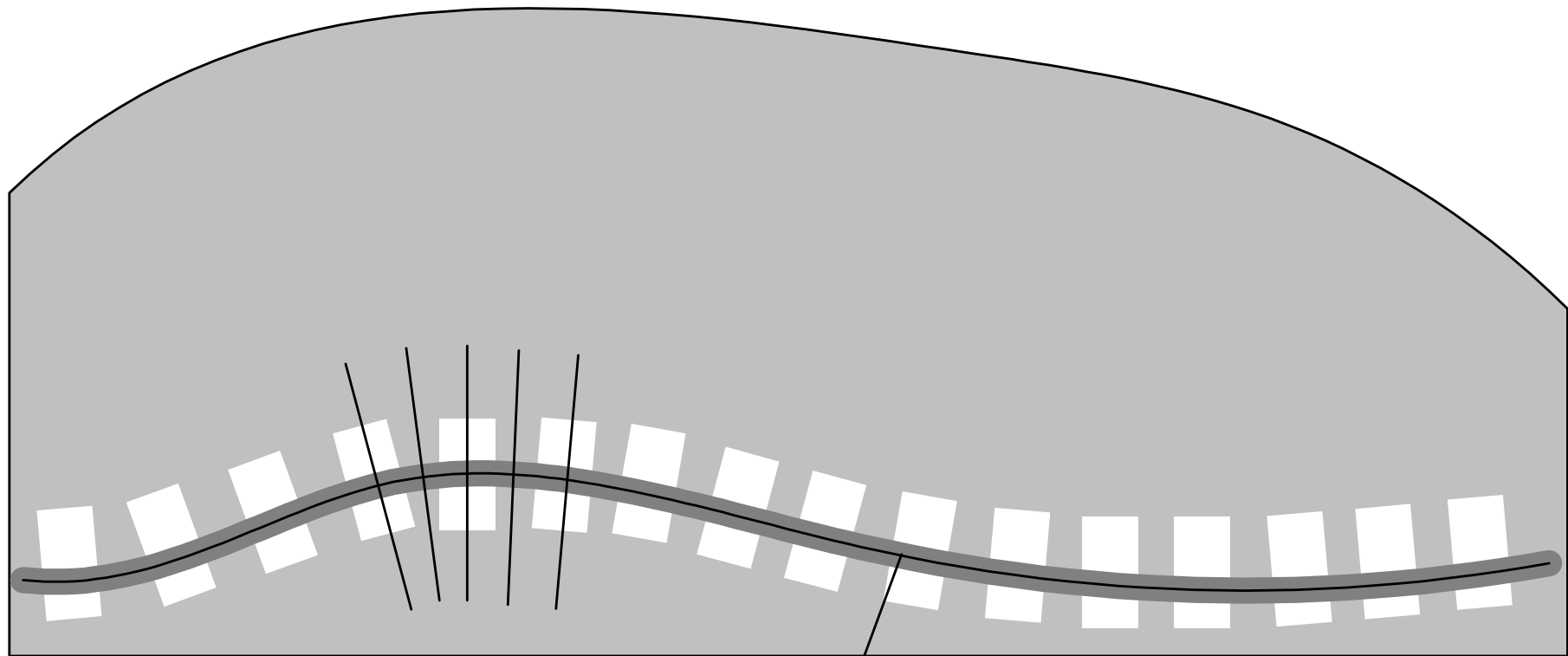
new: SW formatting

20 to 50% less film needed

Idealized layers september 1992



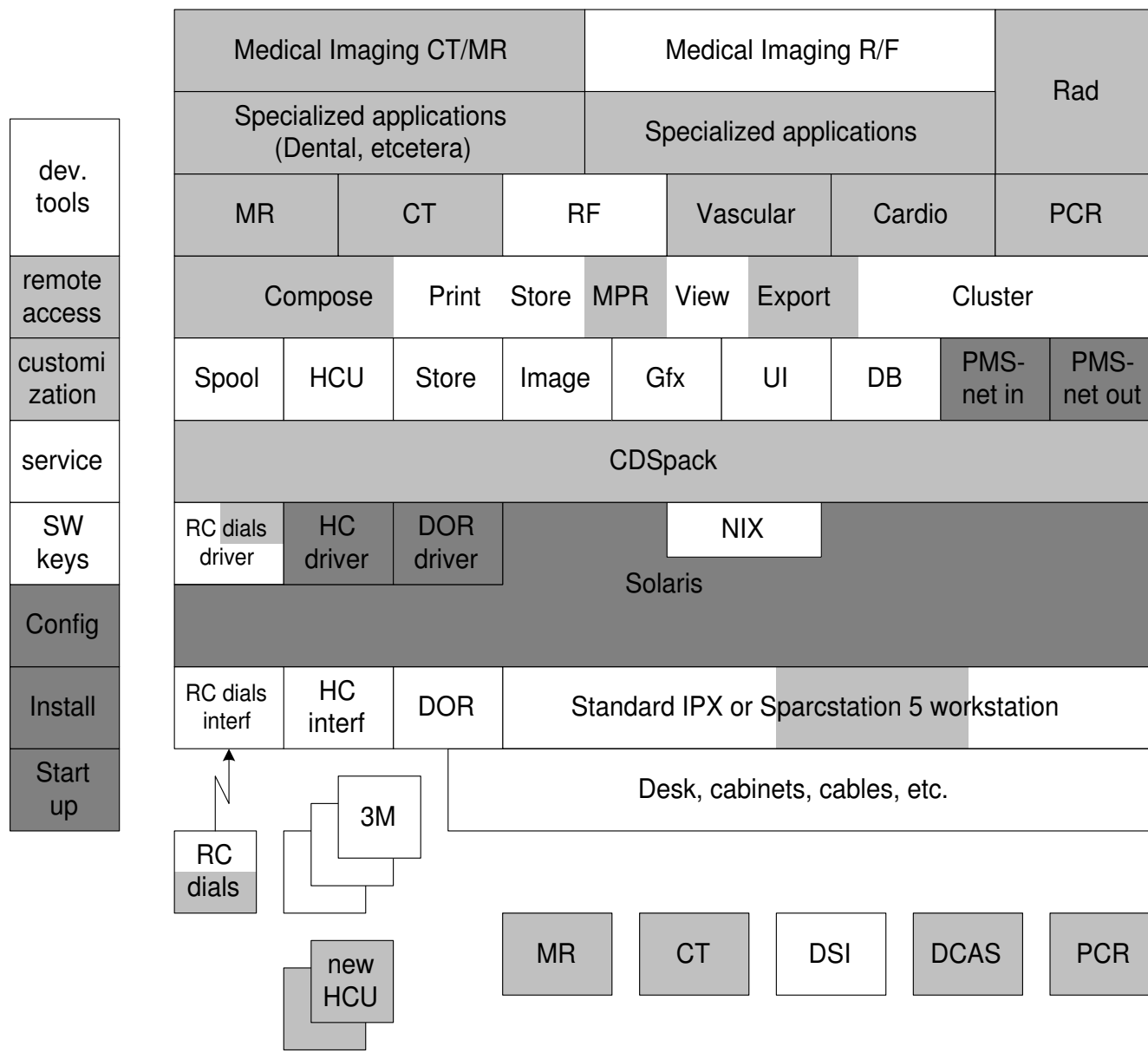
Example Multi Planar Reconstruction



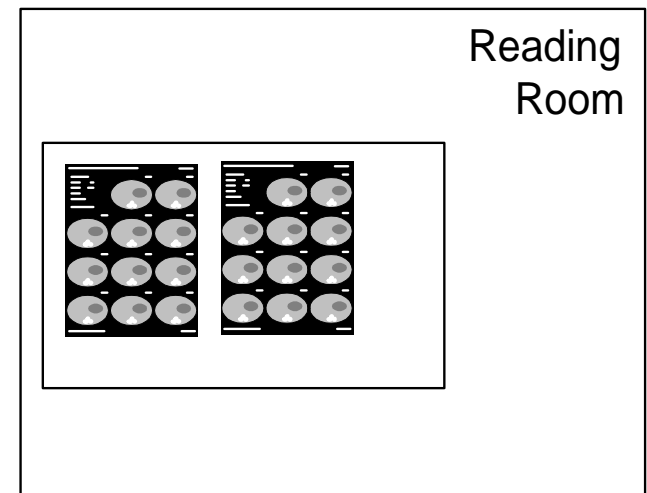
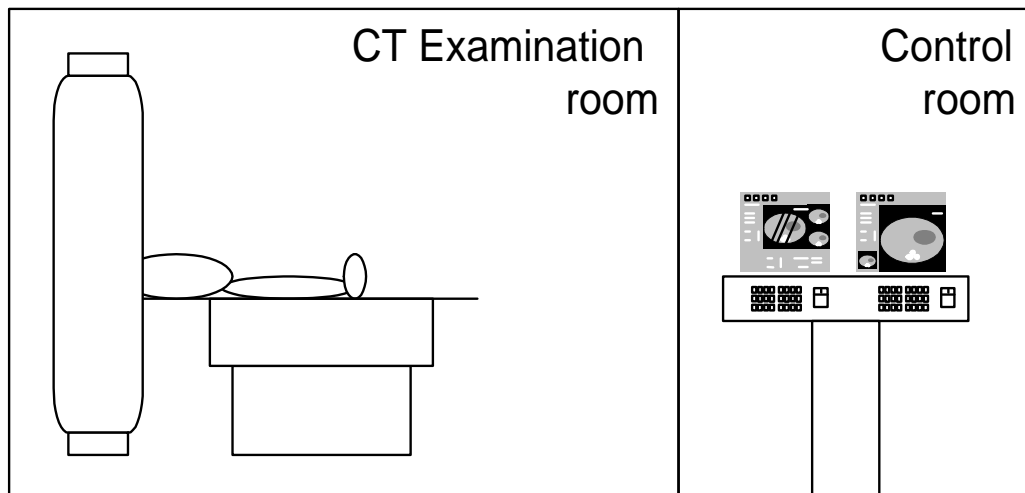
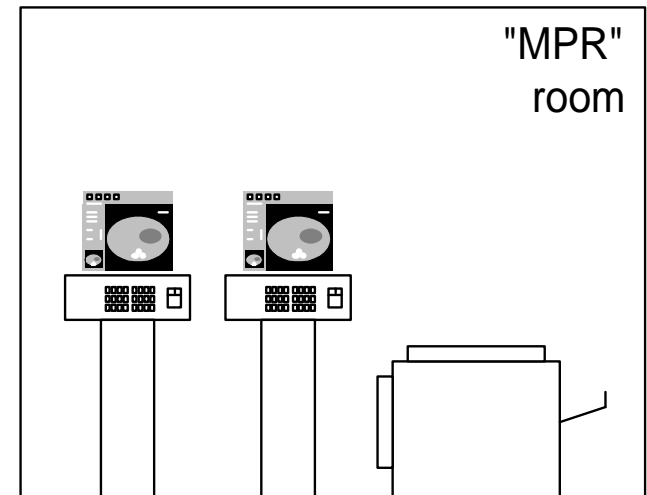
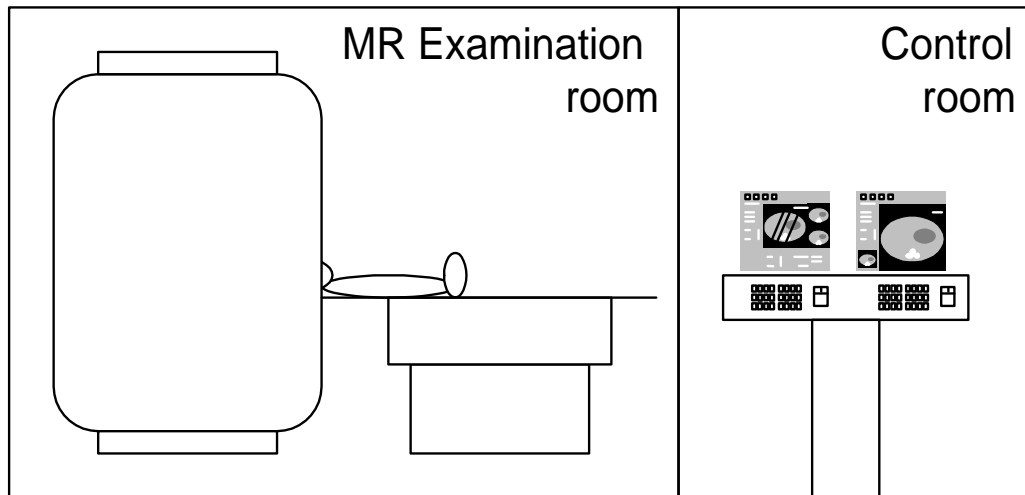
oblique
slices

curved
slice

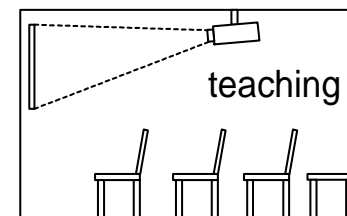
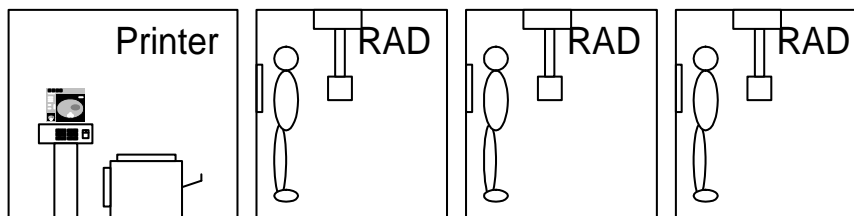
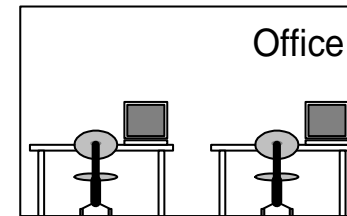
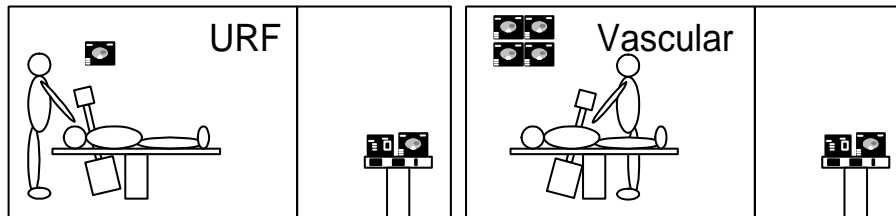
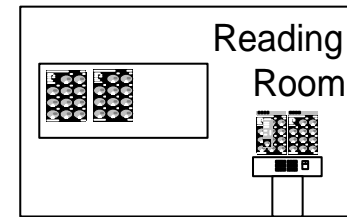
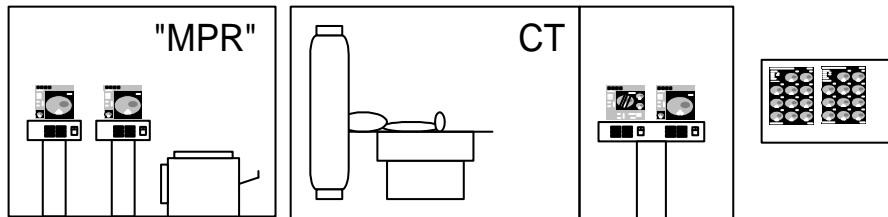
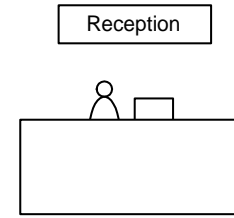
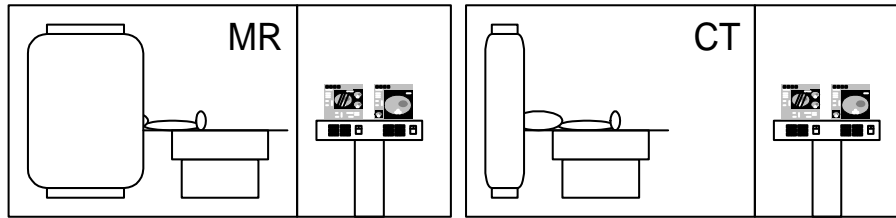
Idealized layers june 1994



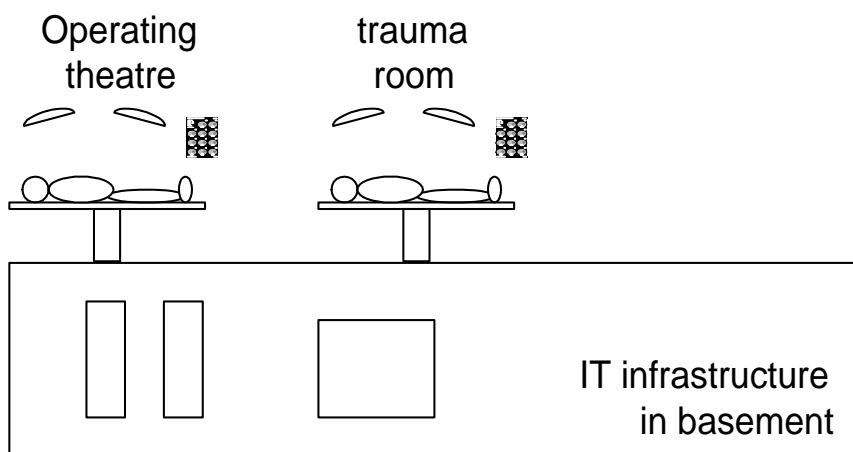
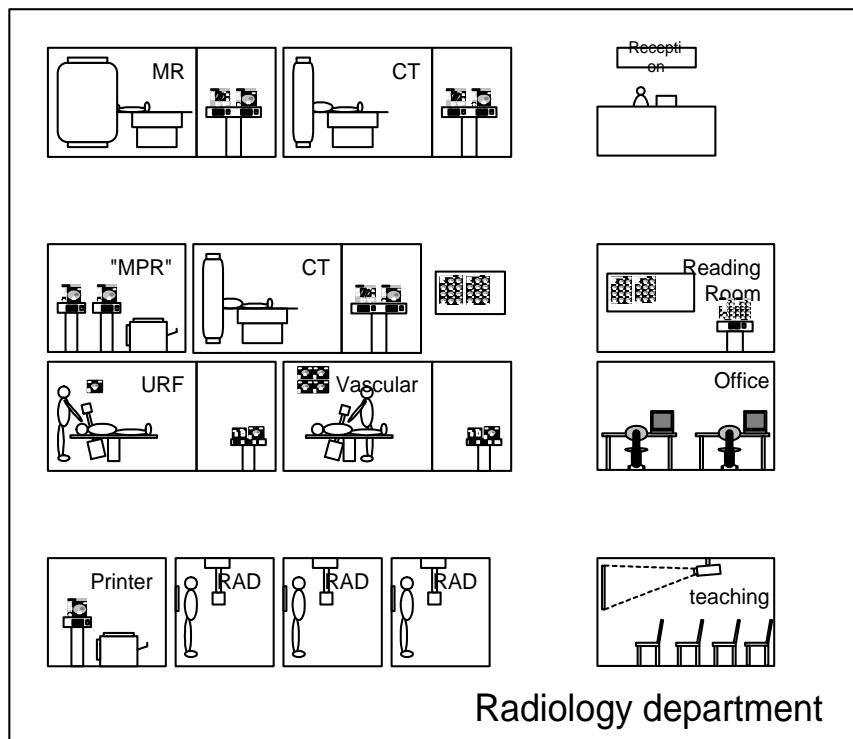
Example CT/MR department



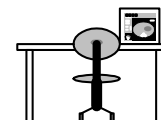
Radiology Department



Vision: Medical Imaging in Healthcare

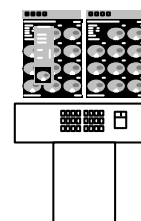


Radiologist at home

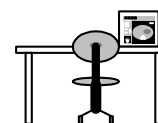


Radiologist somewhere in the hospital

Radiologist at other hospital



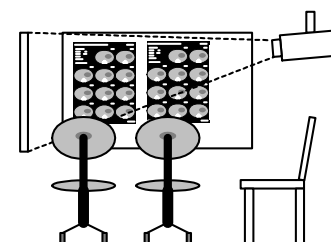
Referring Physician



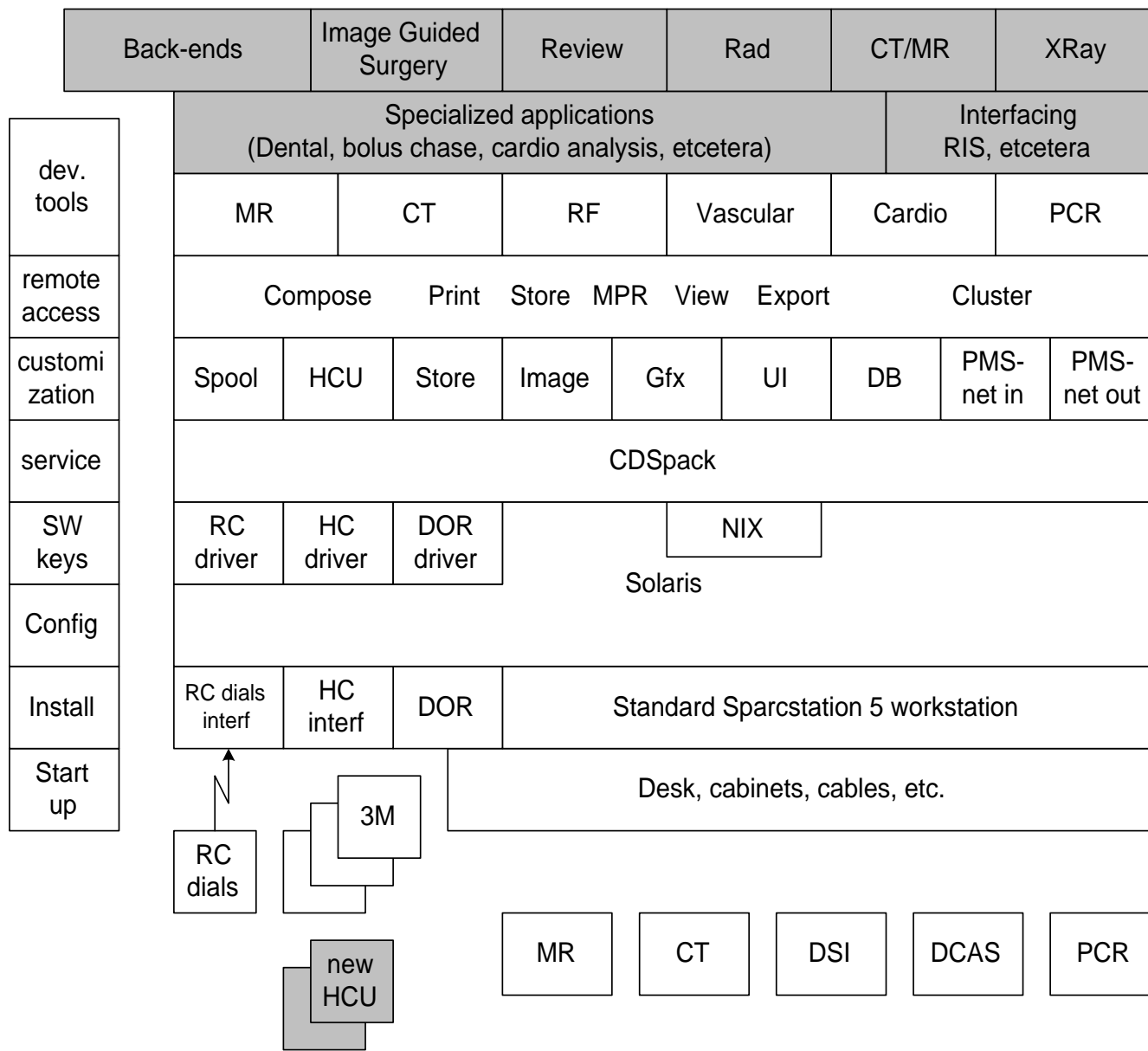
Referring Physician



Conference room



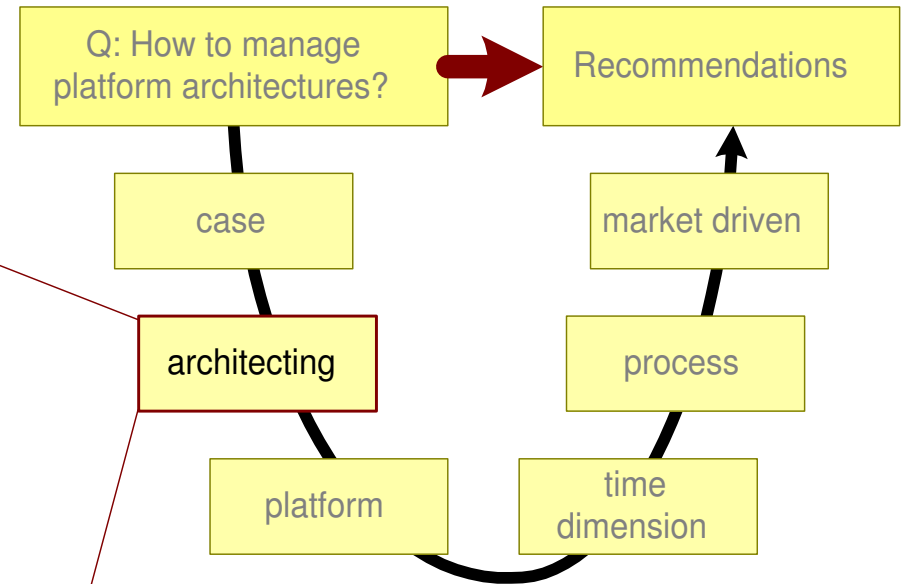
Idealized layers 1996



Architecting Outline

Architecting:

- What is Architecture?
- Architecture vs Description
- My View on Architecture
- "Guiding How"
- The Art of Architecting
- More than Decomposition



What is Architecture?

Mark all applicable boxes



specifications

indicators

overarching vision
market, business, technology
process, product

API's

components (implementations)

high level rules
layer n !calls n-k; k>1
layer n !calls n+k; k>0

guidance monitoring

infrastructure

concepts

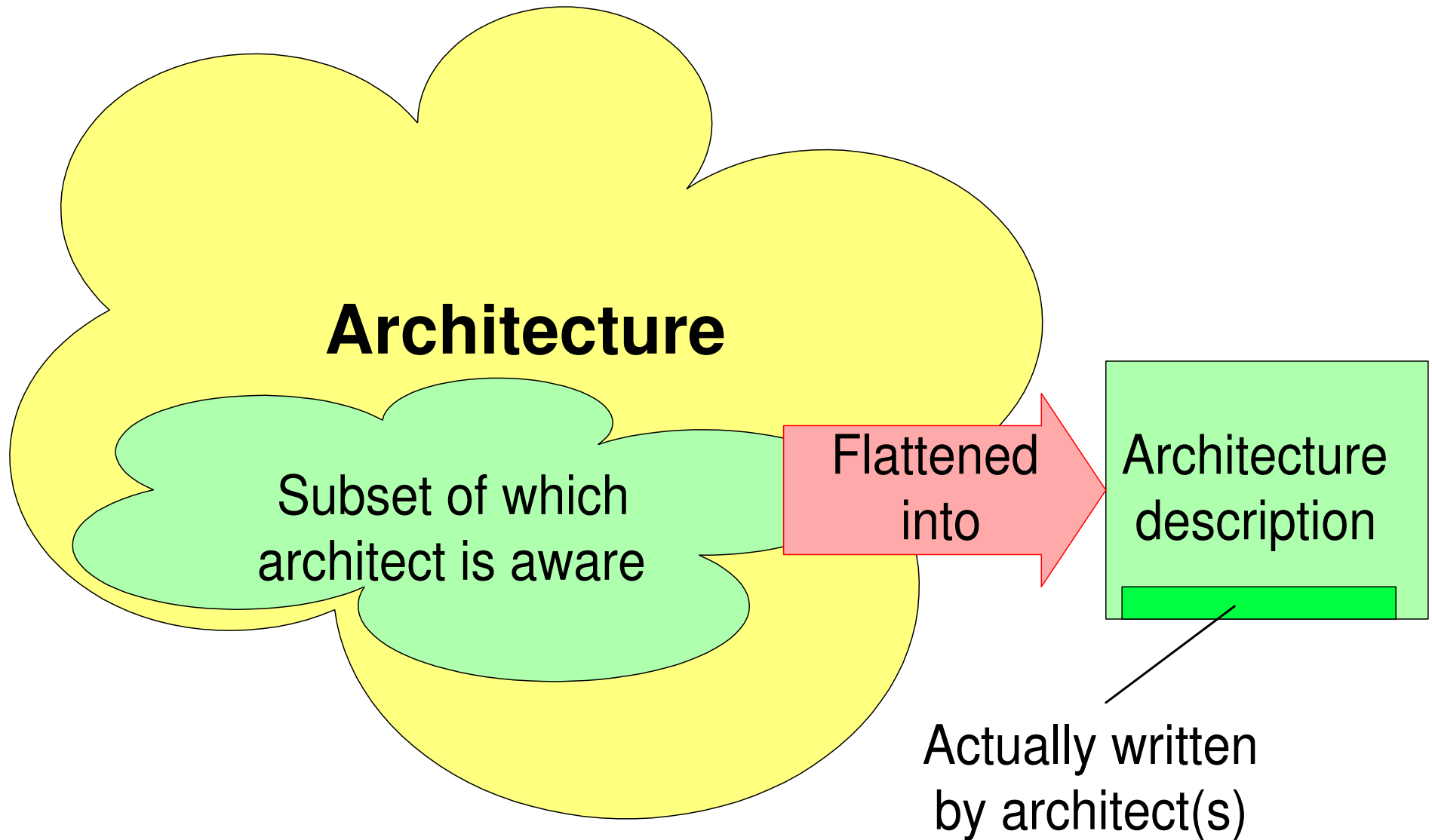
standards

domain codification

Java
SQL
FPGA
technology

other...

Architecture vs Description



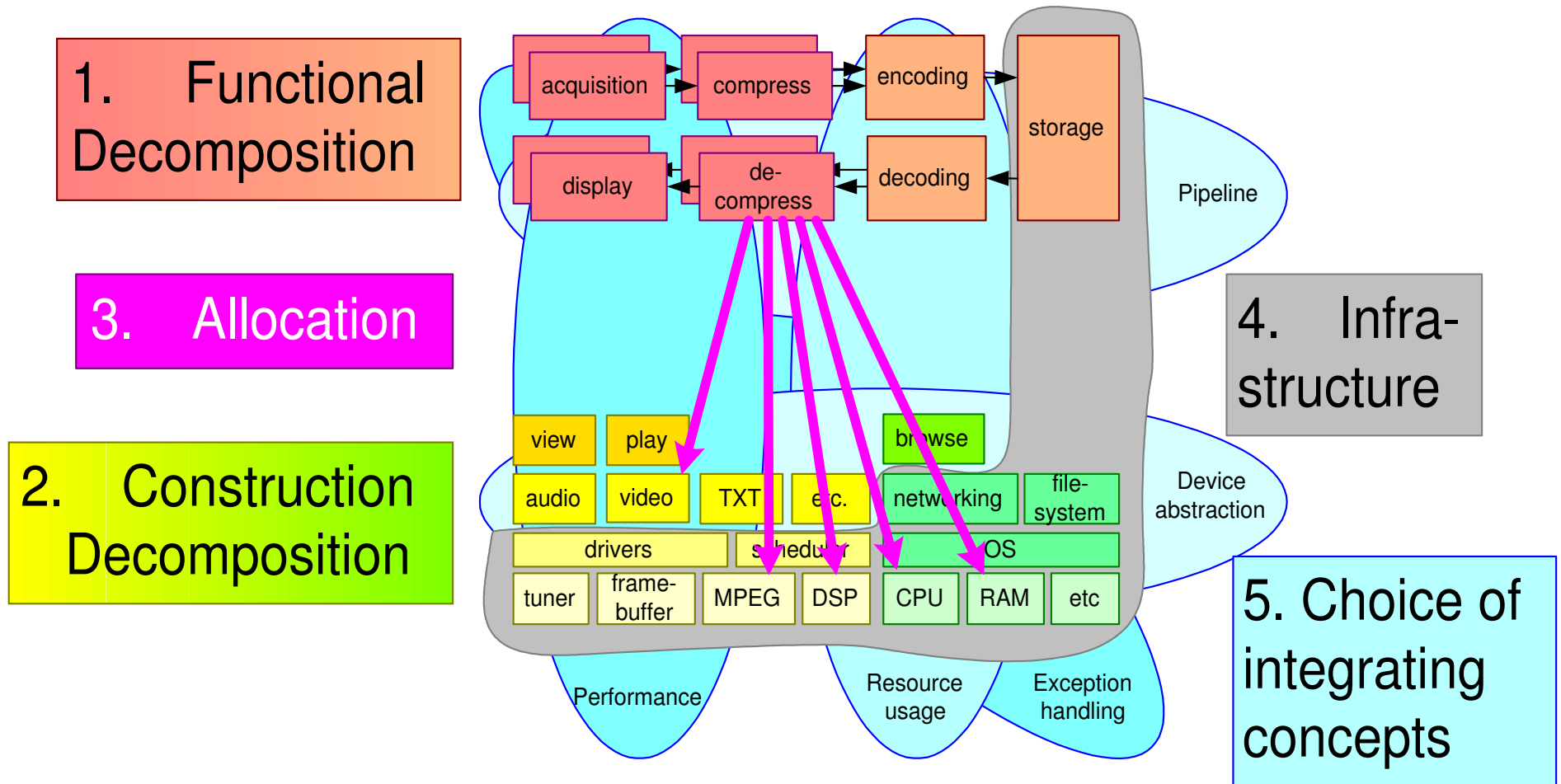
My View on Architecture



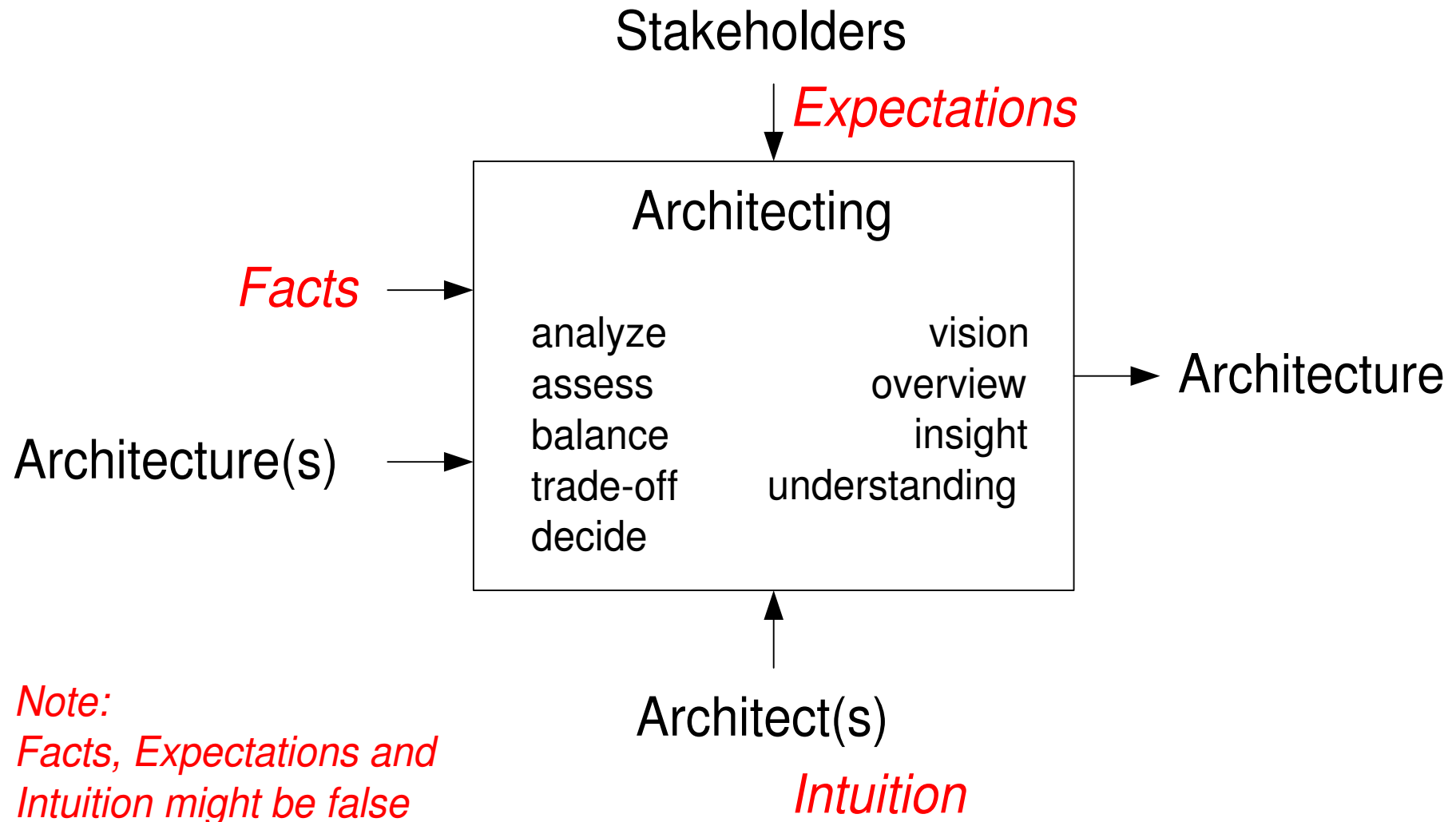
Do the right things

Do the things right

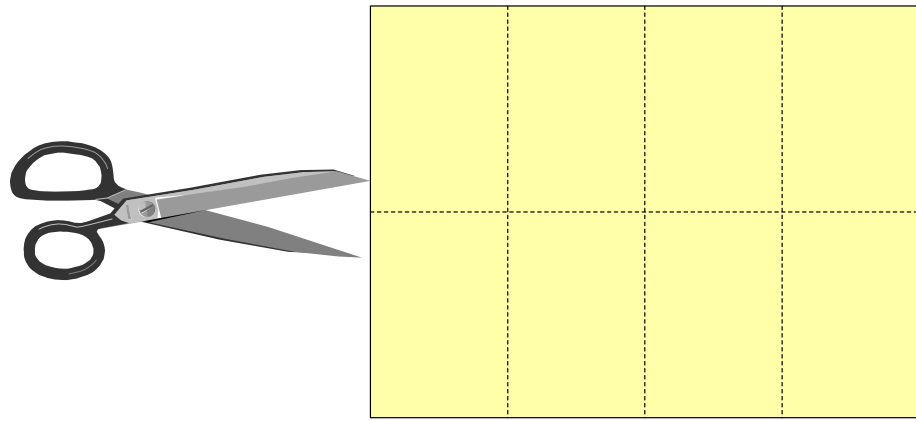
"Guiding How" by providing rules for:



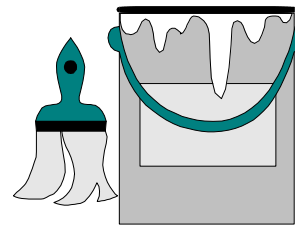
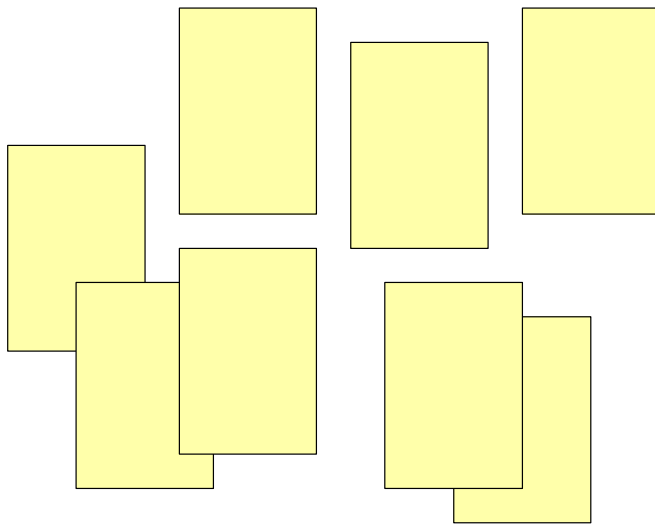
The Art of Architecting



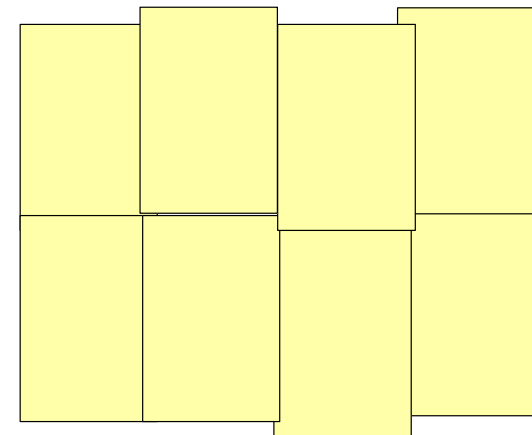
Architecting is much more than Decomposition



Decomposition
is "easy"

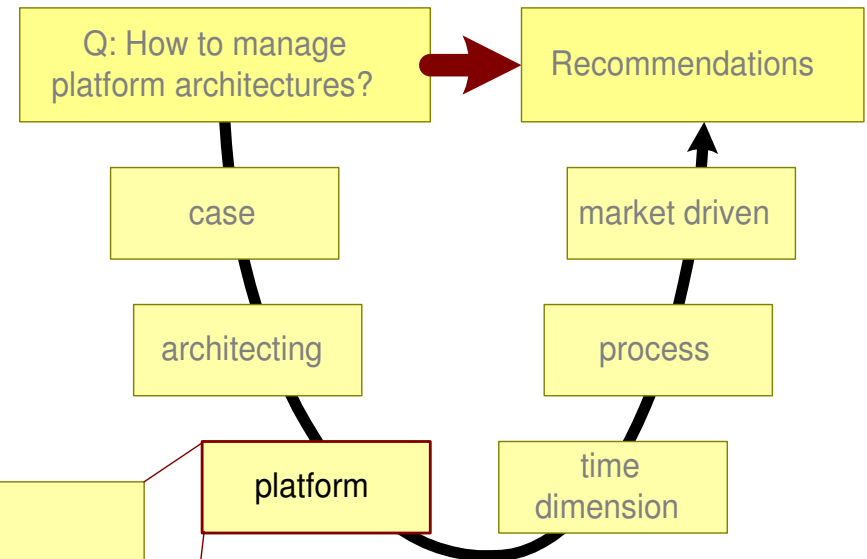


Integration is
difficult

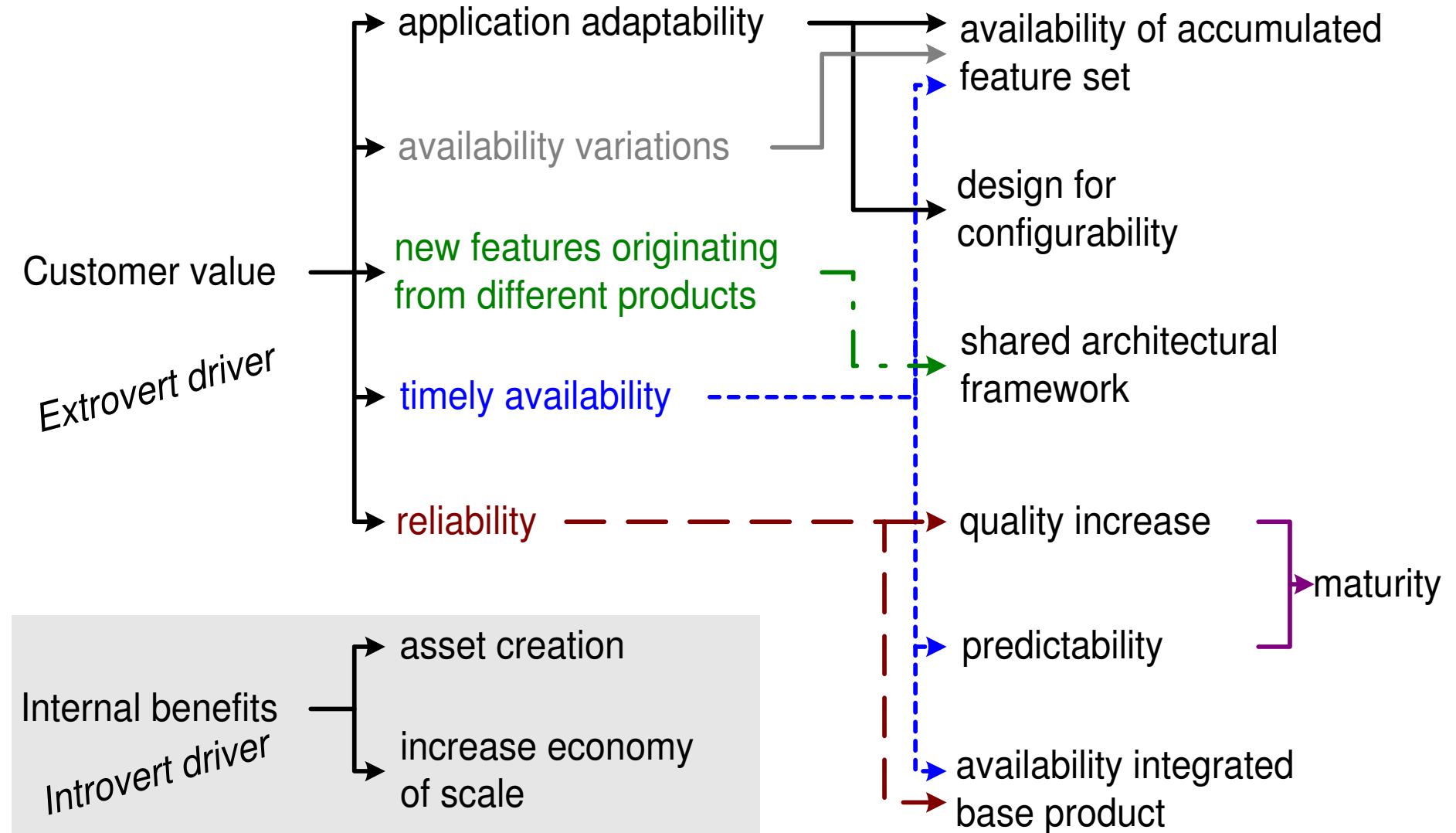


Platform Outline

- Platform:
- Why Platforms?
 - What is a Platform?
 - Platform Source Deliverables
 - Example of Platform Efficiency
 - Embedding Costs of Purchased SW



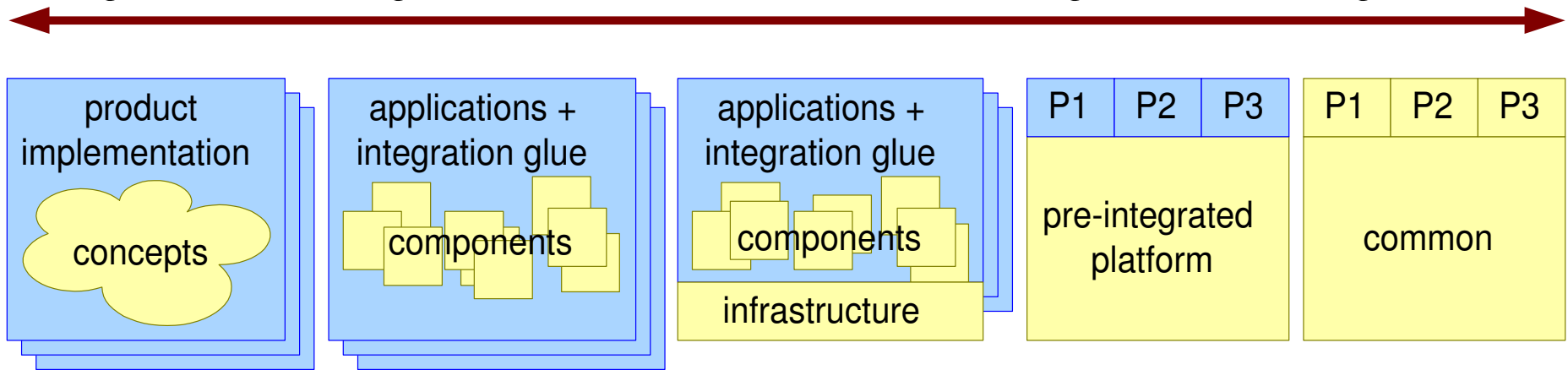
Why Platforms?



What is a Platform?

huge product integration effort
very flexible
low coupling
configuration management???

no product integration effort
not flexible
high coupling
configuration management

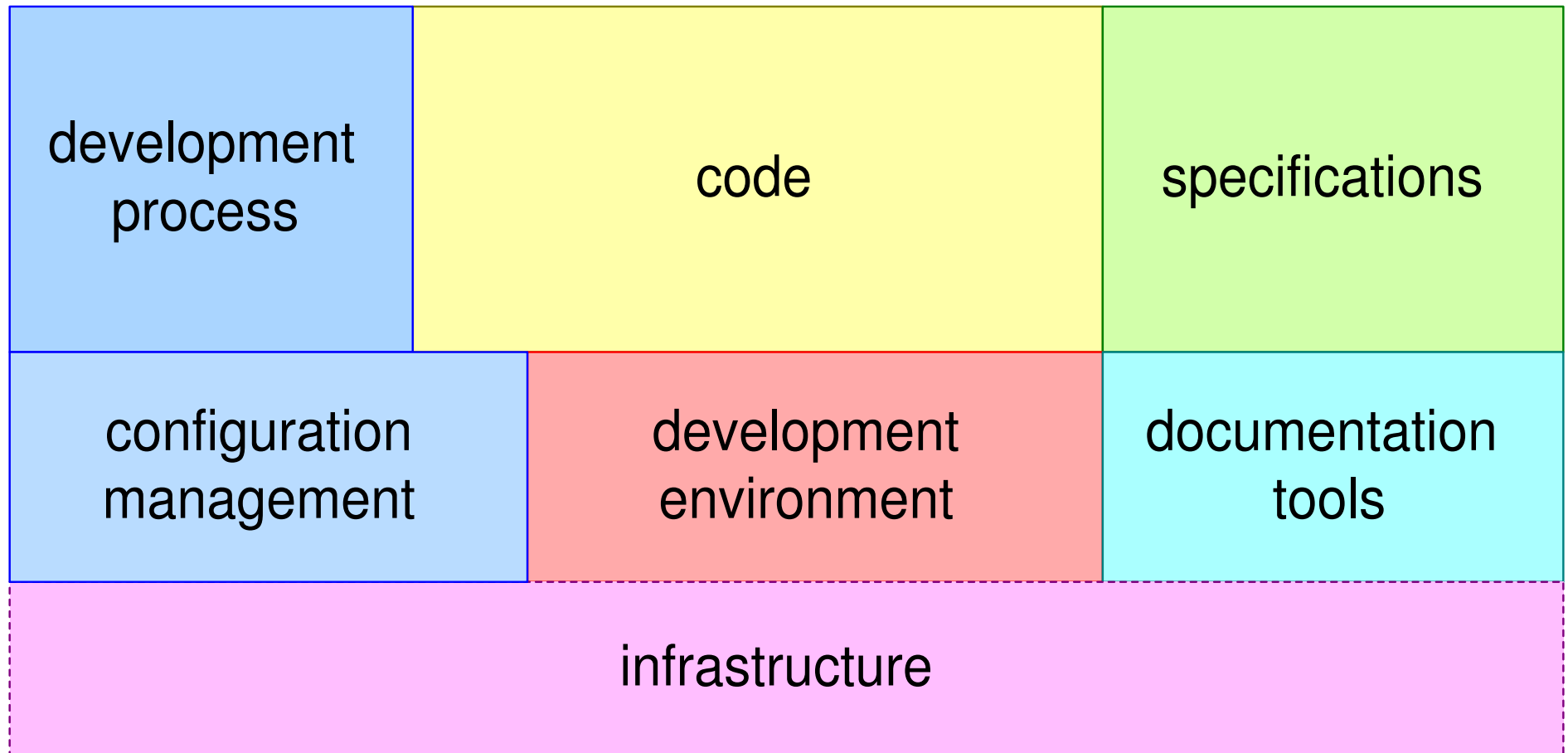


legend

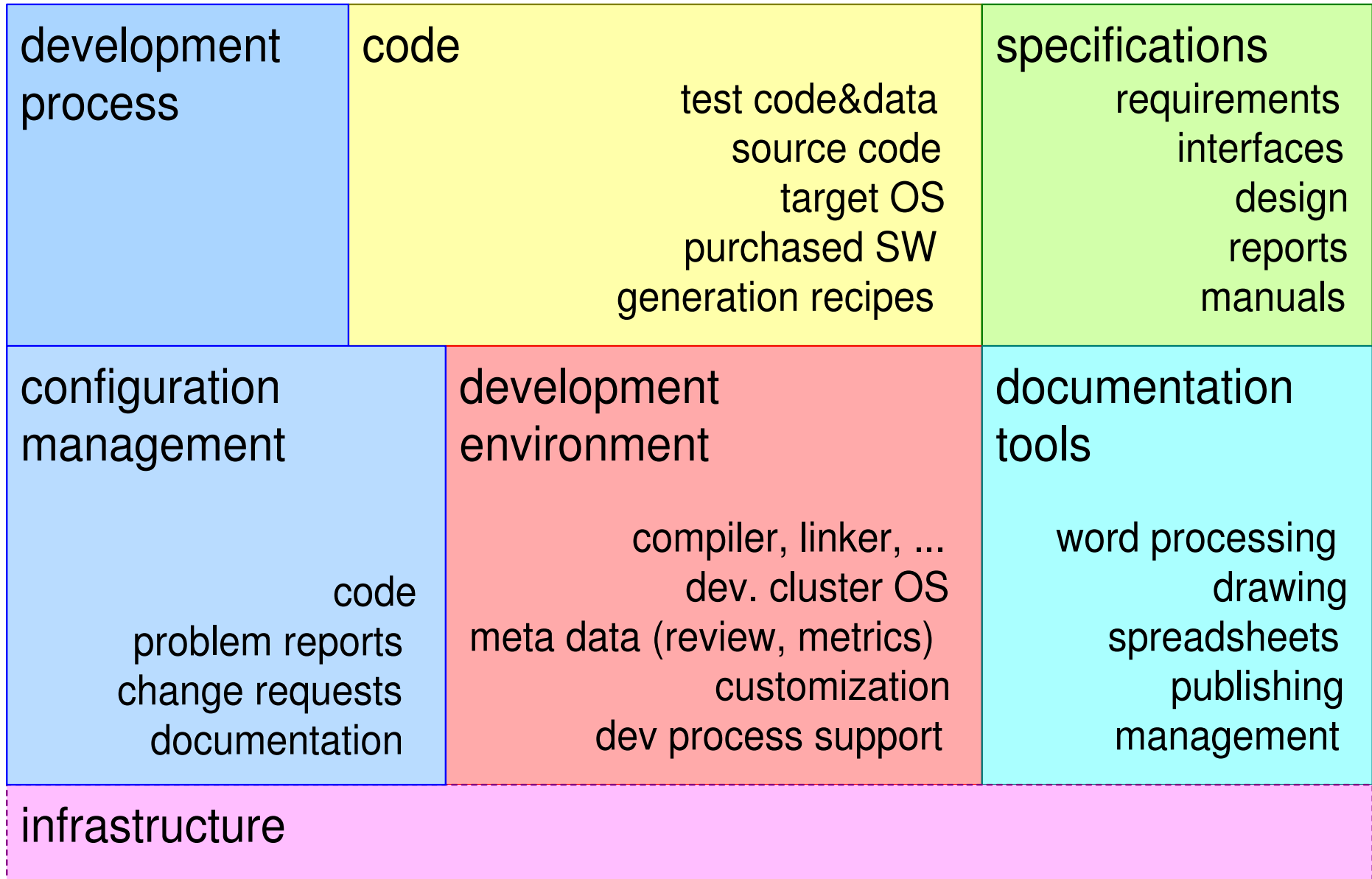
product

platform

Platform Source Deliverables



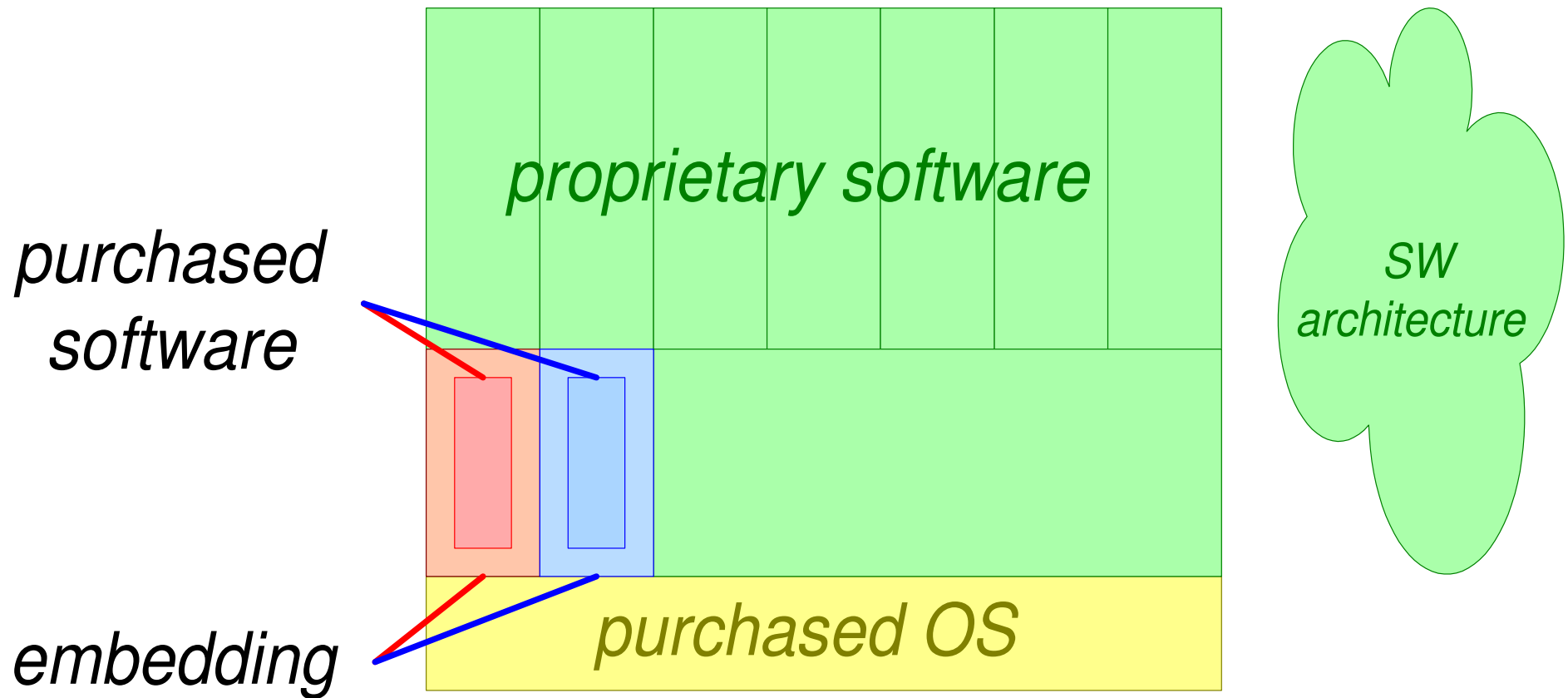
And now in More Detail...



Example of Platform Efficiency

		1992	1993	1994	1995	1996
<i>value metric</i>	applications	1	4	8	16	32
	number of inputs (a.o. modalities)	1	5	10	15	
<i>number of people</i>	platform			35	37	38
	applications			27	35	41
	total		52	62	72	79
<i>efficiency</i>	people per application		13	8	5	3

Purchased SW Requires Embedding



Embedding Costs of Purchased SW

- Installation
- Configuration
- Customization
- Start up, shutdown
- Specifications
- Interface to application SW
- Exception handling
- Resource allocation and monitoring provision
- Resource tuning, see above
- Safety design
- Security design

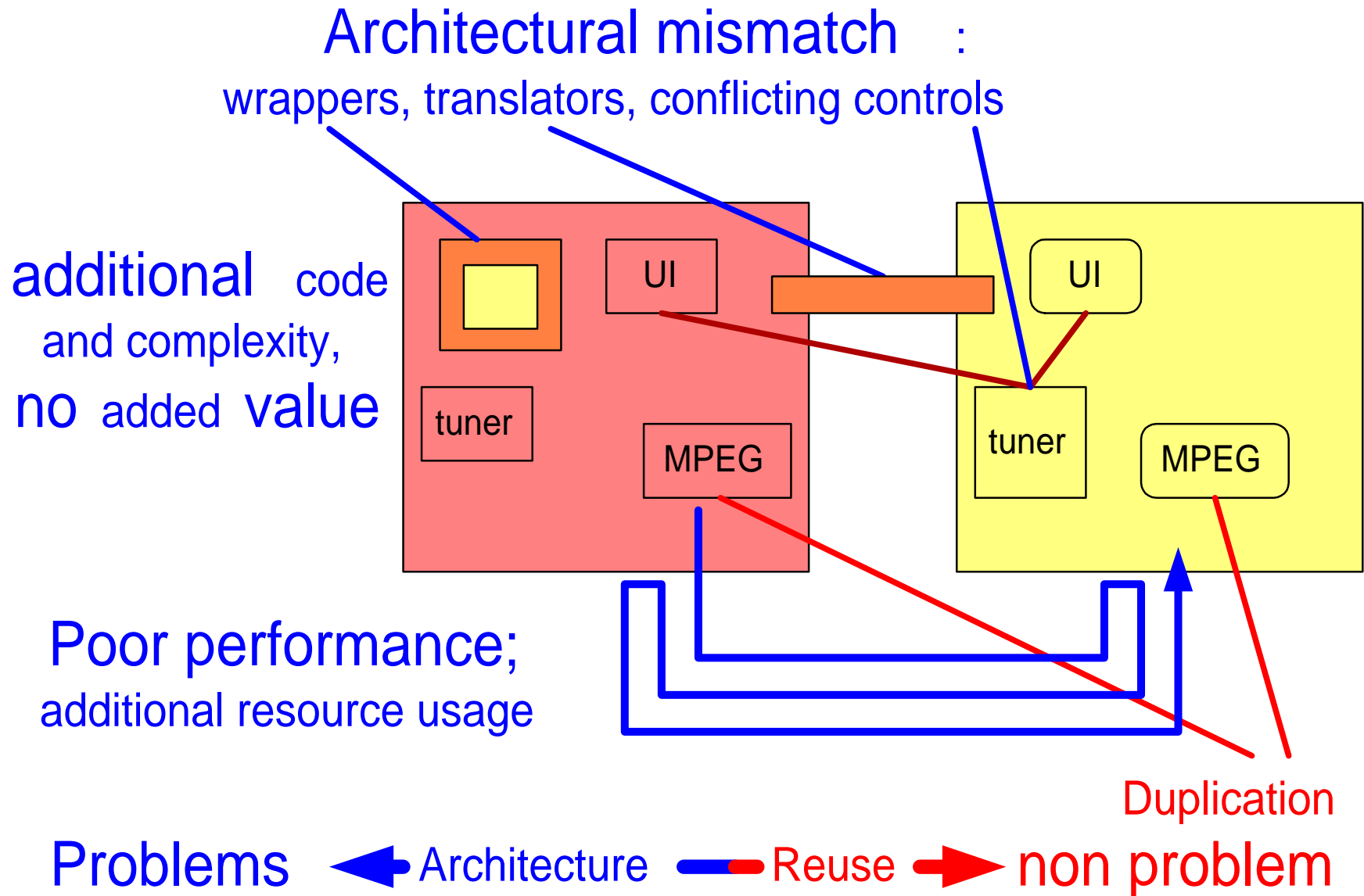
functional
system design
sw design

add semantics level
use of appropriate low level mechanisms
match to high level mechanisms:
- notification, scheduling
- job requests, subscriptions

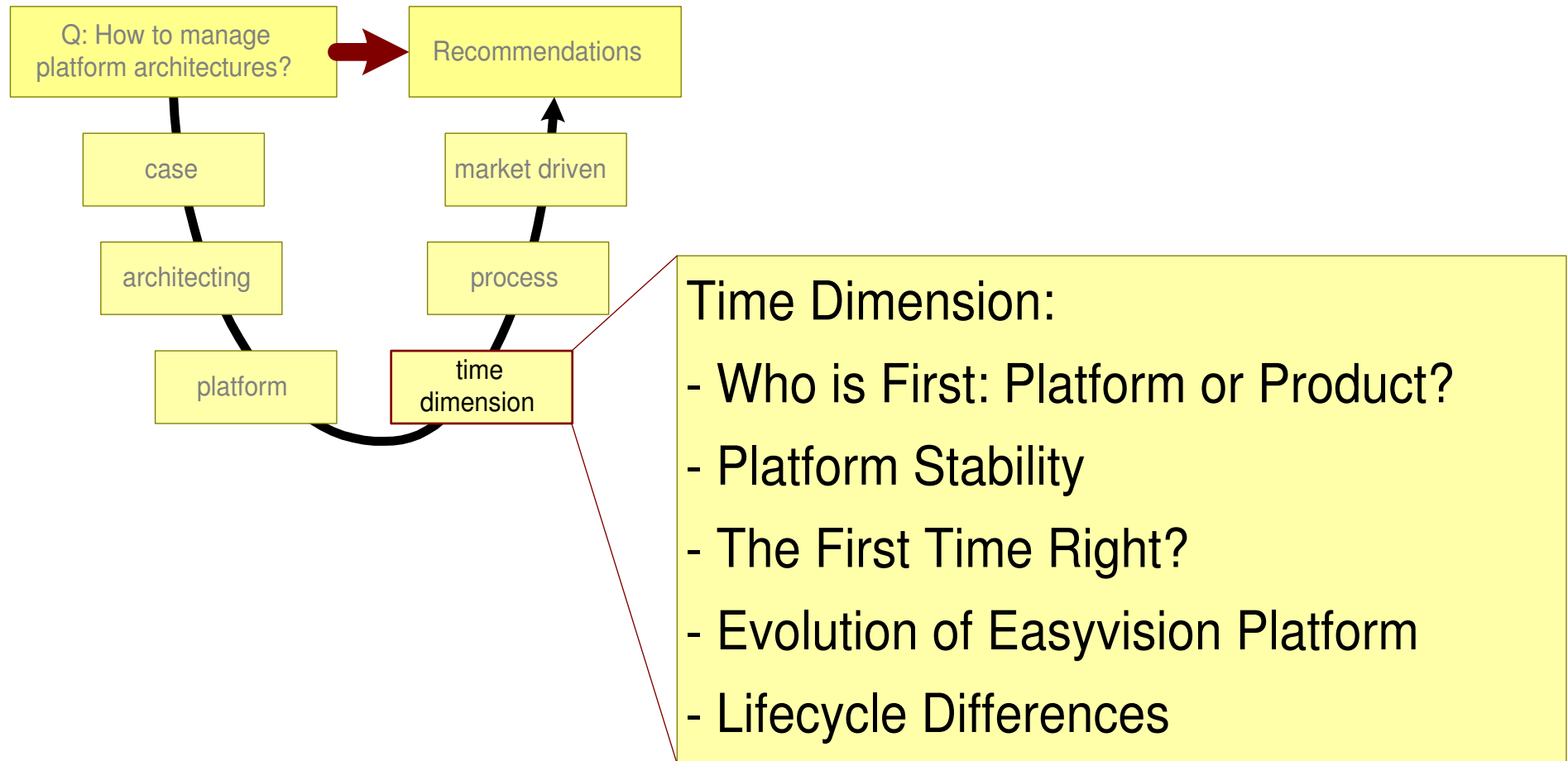
System monitor
Error propagation
Logging

CPU
Memory
Disk

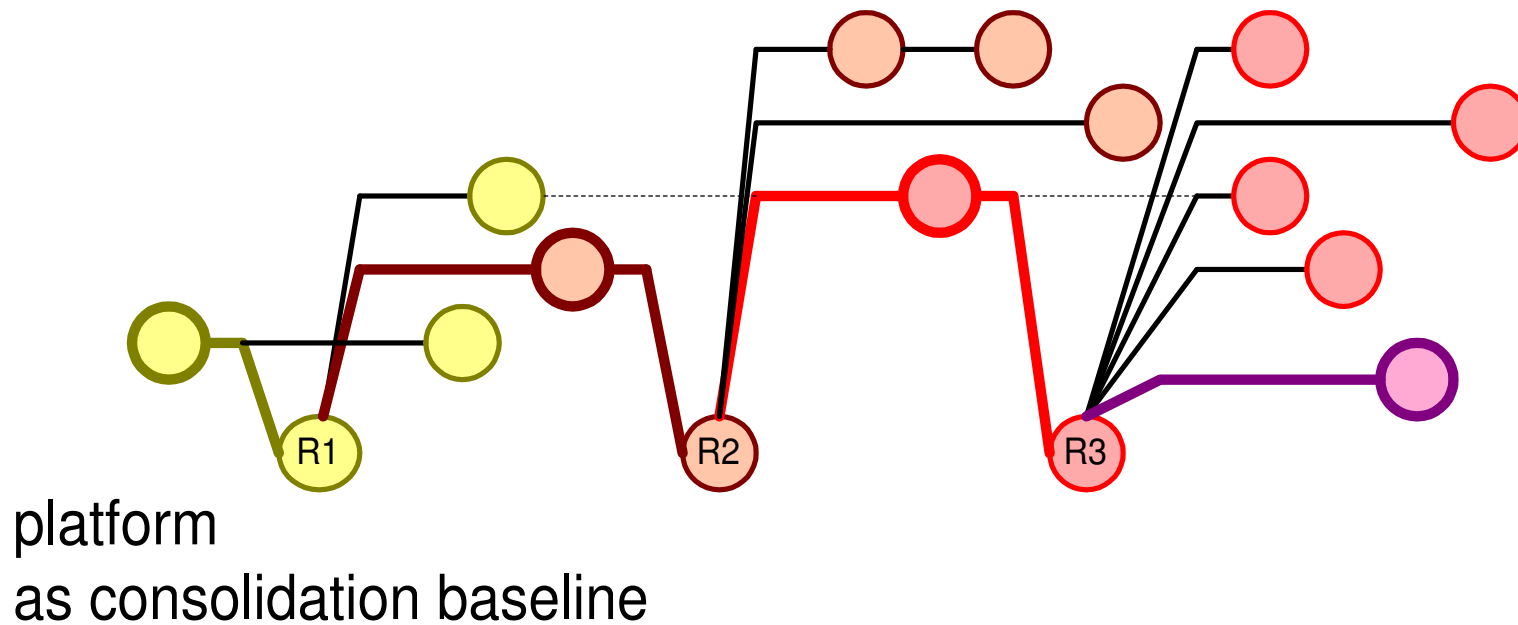
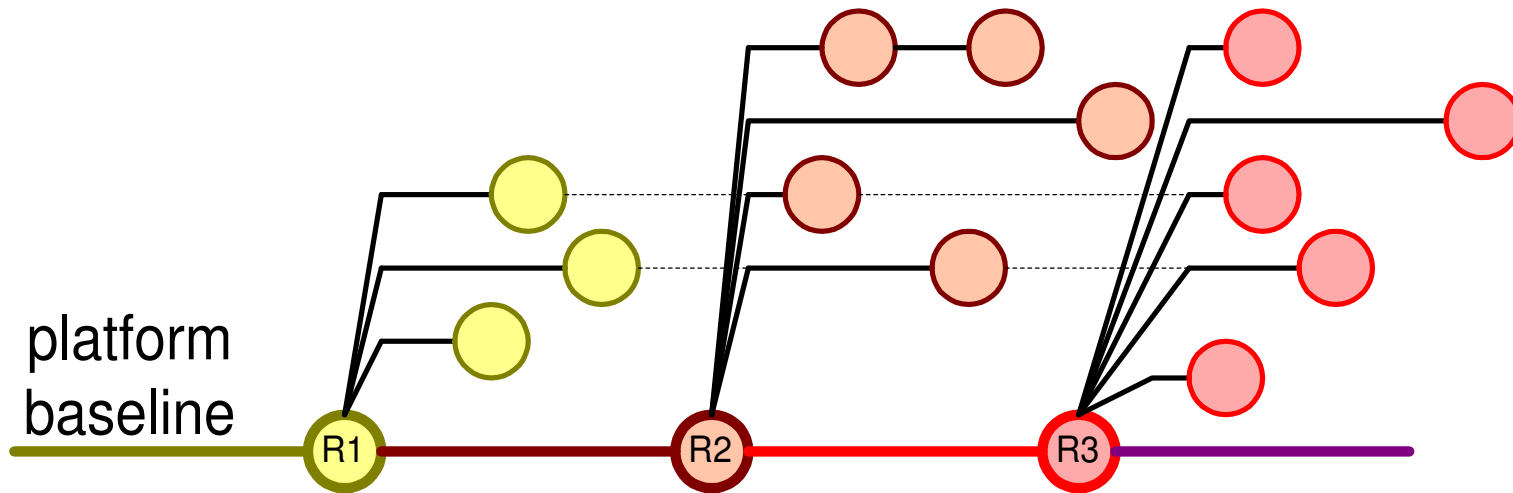
Example of Embedding Problems



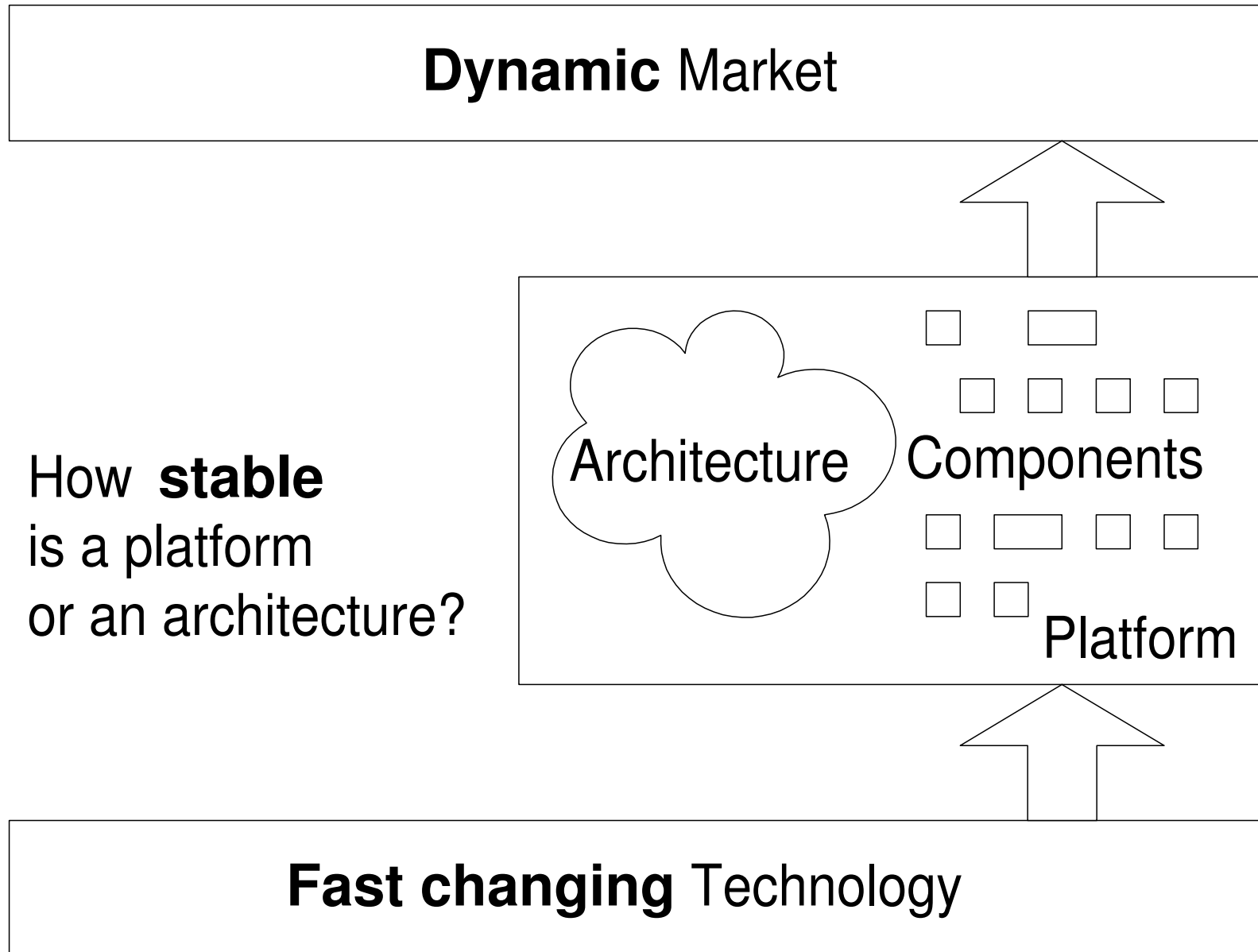
Time Dimension Outline



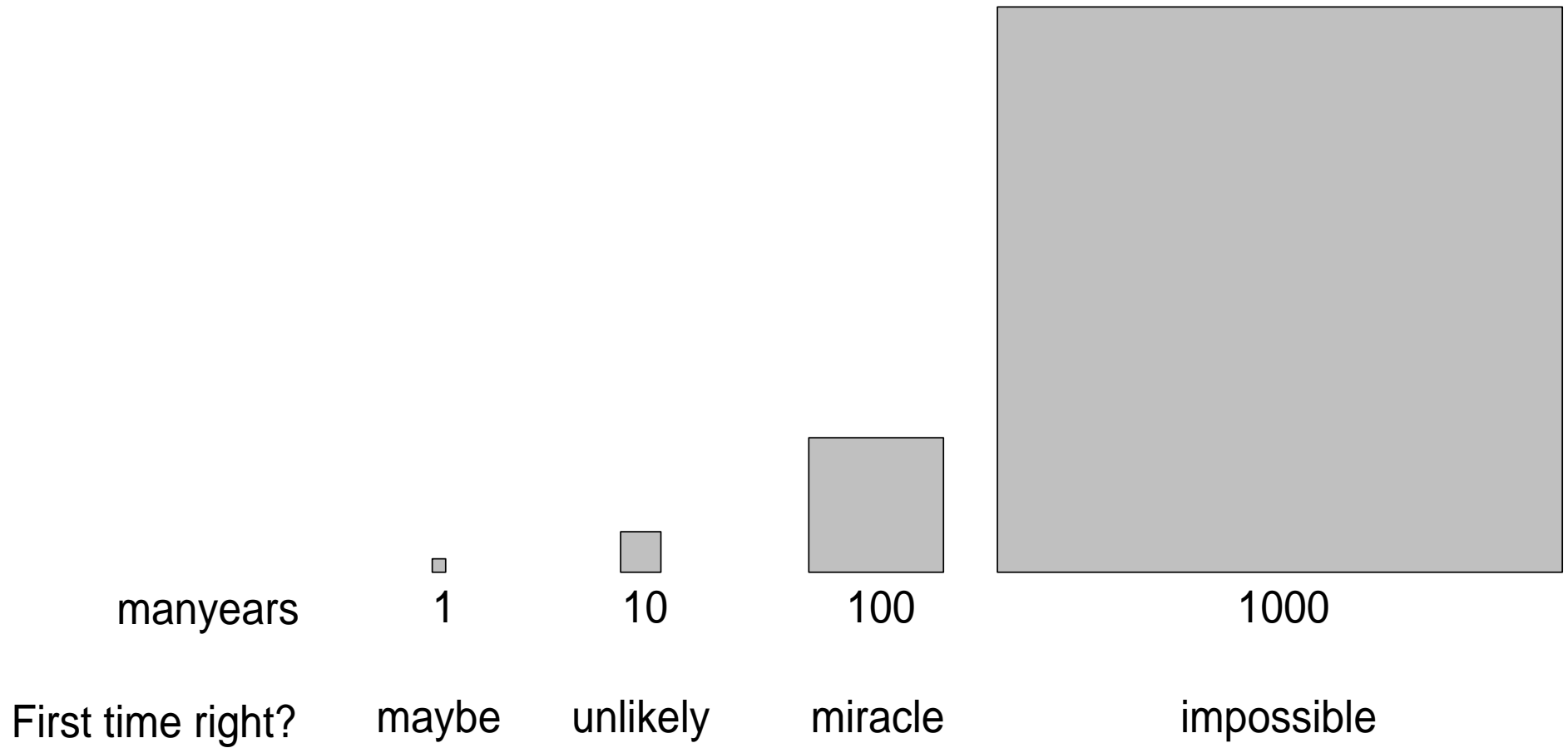
Who is First: Platform or Product?



Myth: Platforms are Stable

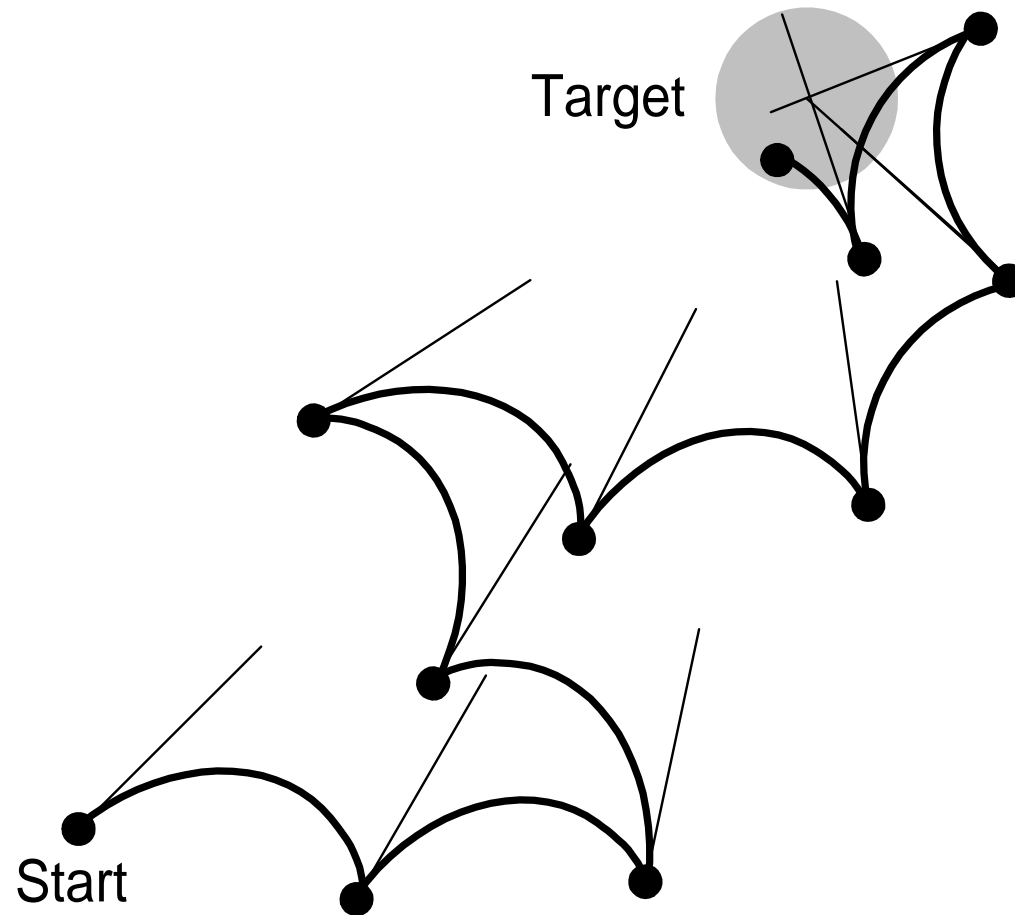


The First Time Right?



Feedback

stepsize: 3 months
elapsed time: 25 months

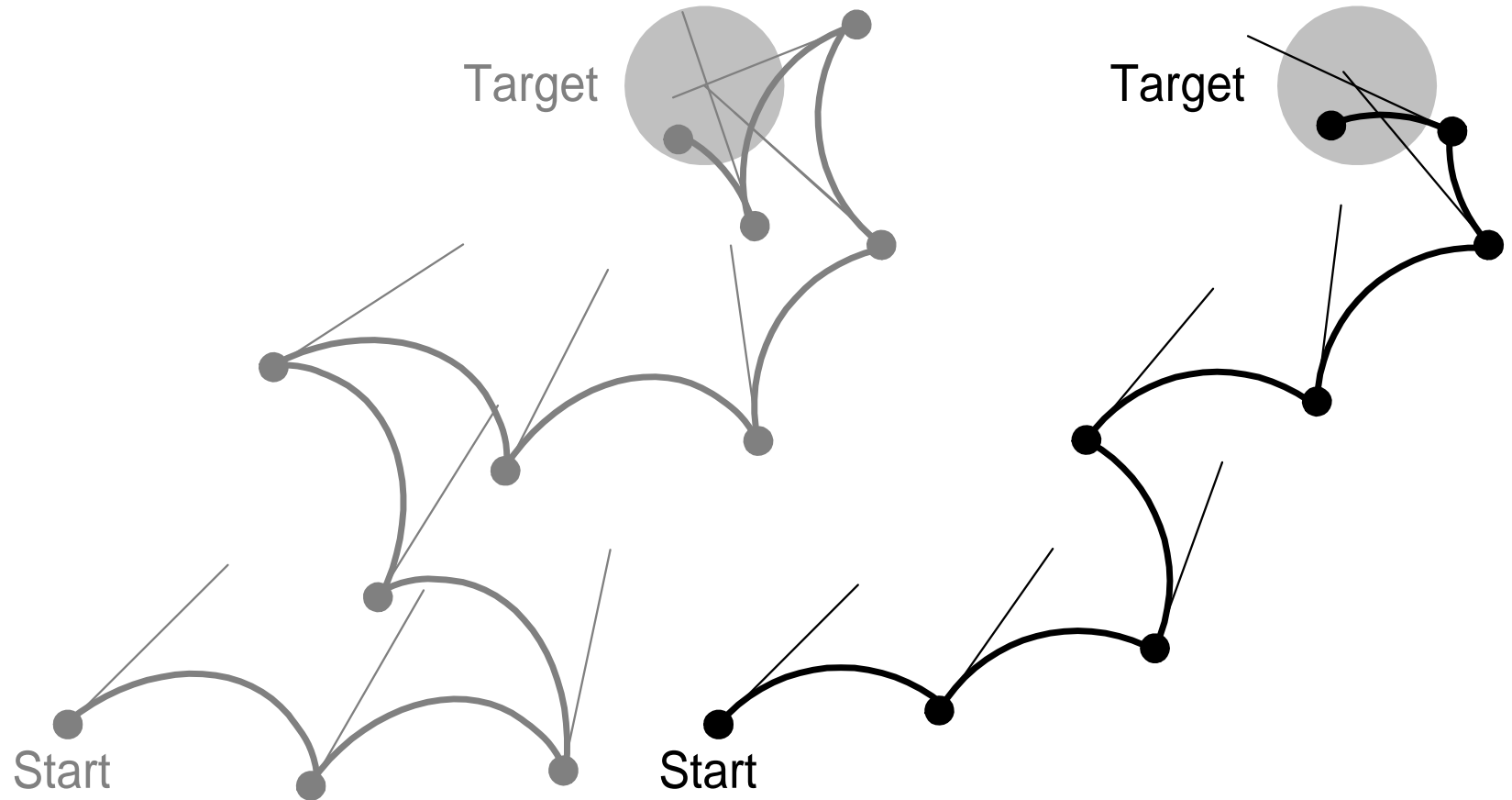


Feedback (2)

stepsize:
elapsed time

3 months
25 months

2 months
12 months



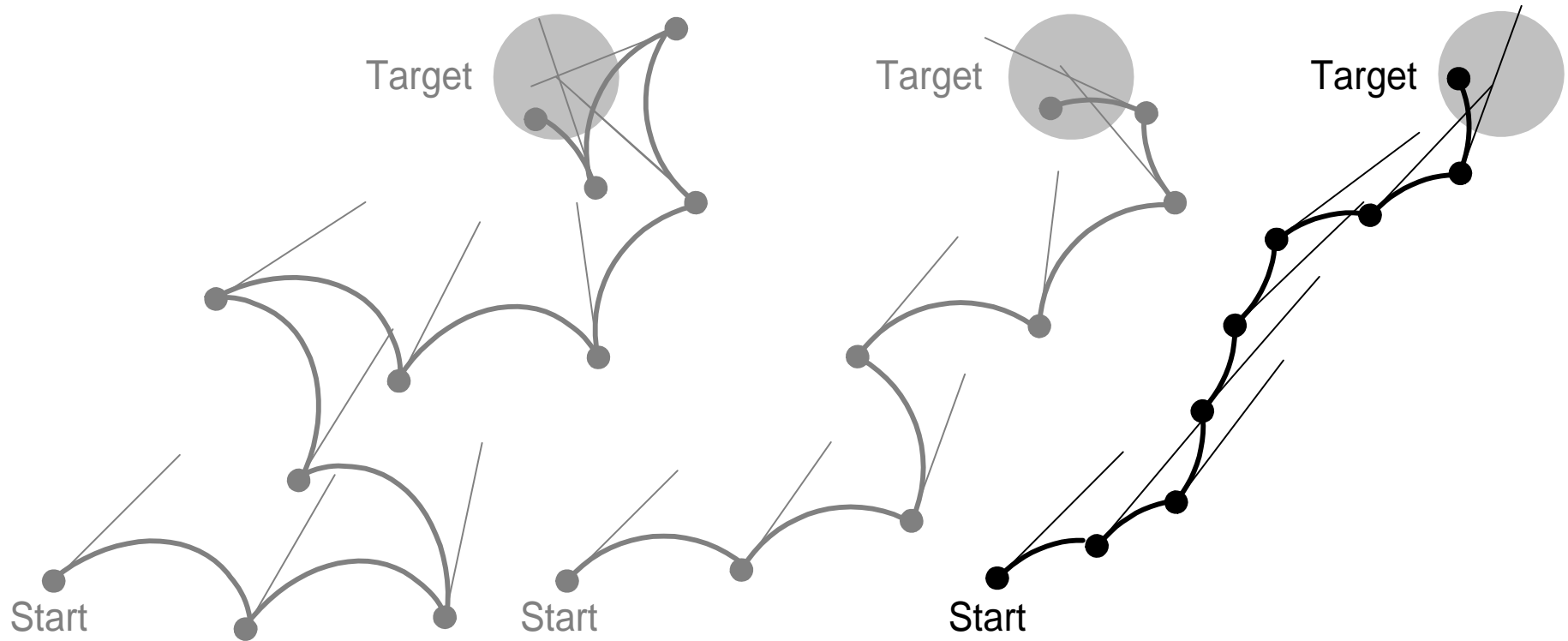
Feedback (3)

stepsize:
elapsed time

3 months
25 months

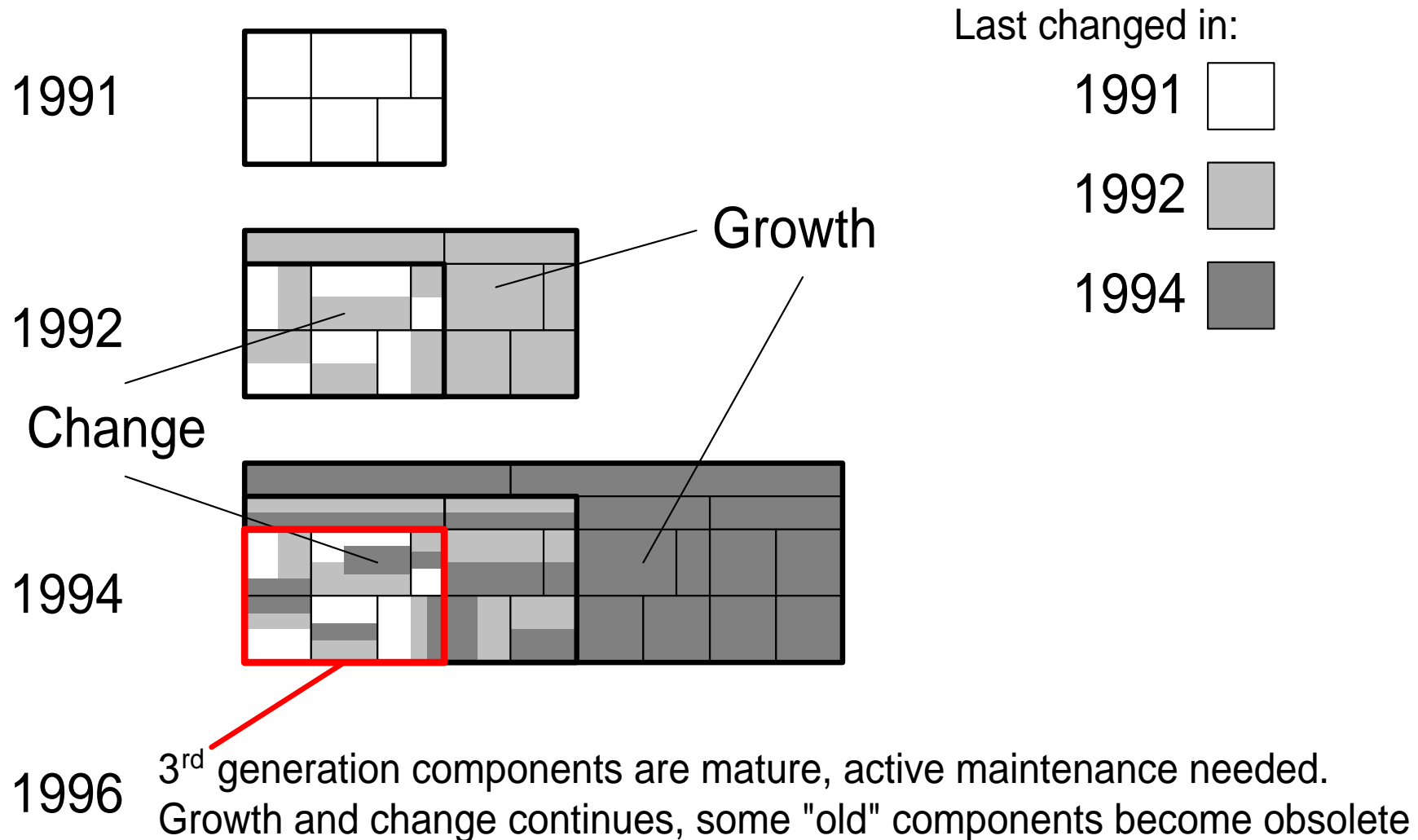
2 months
12 months

1 month
8 months

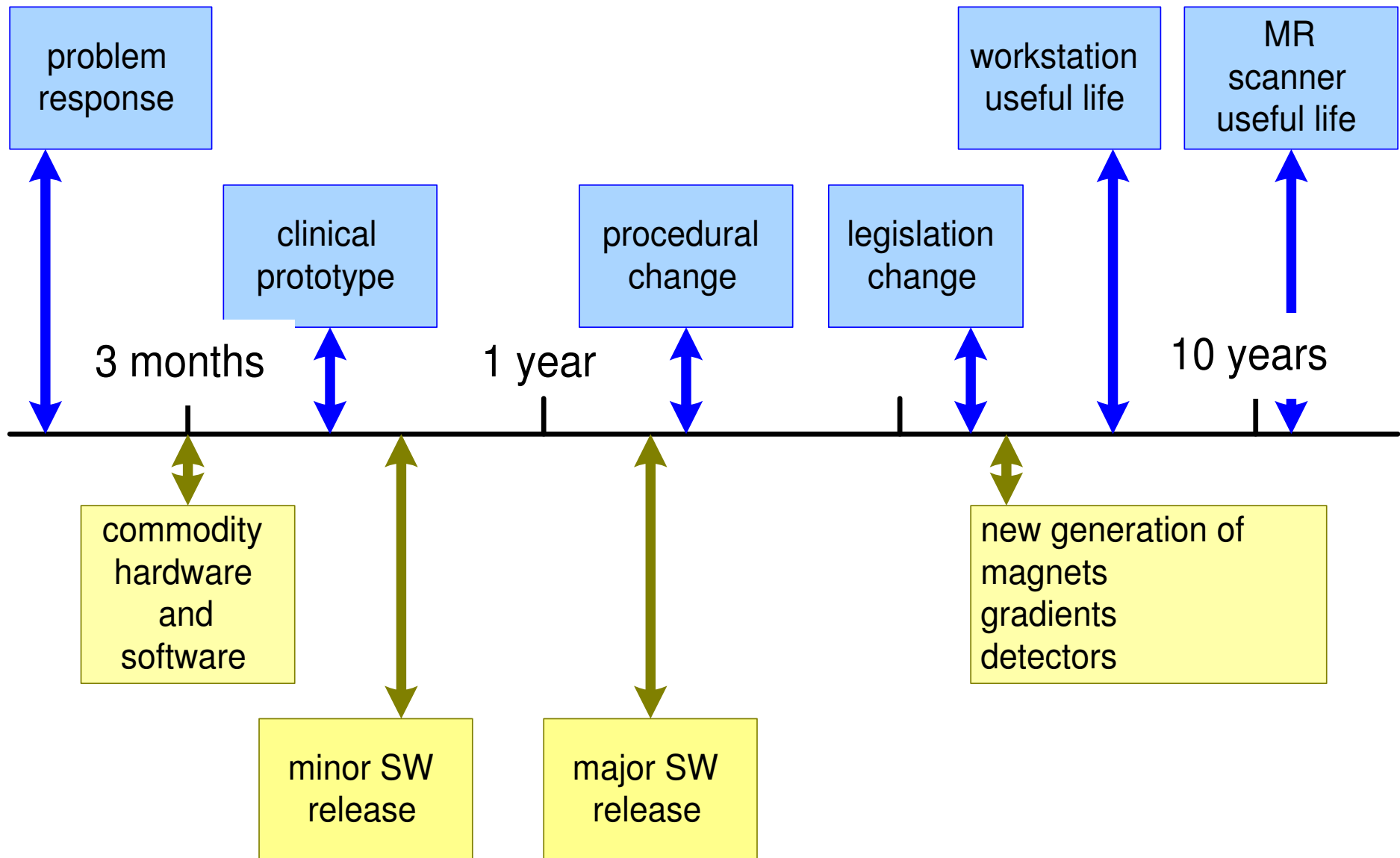


Small feedback cycles result in Faster Time to Market

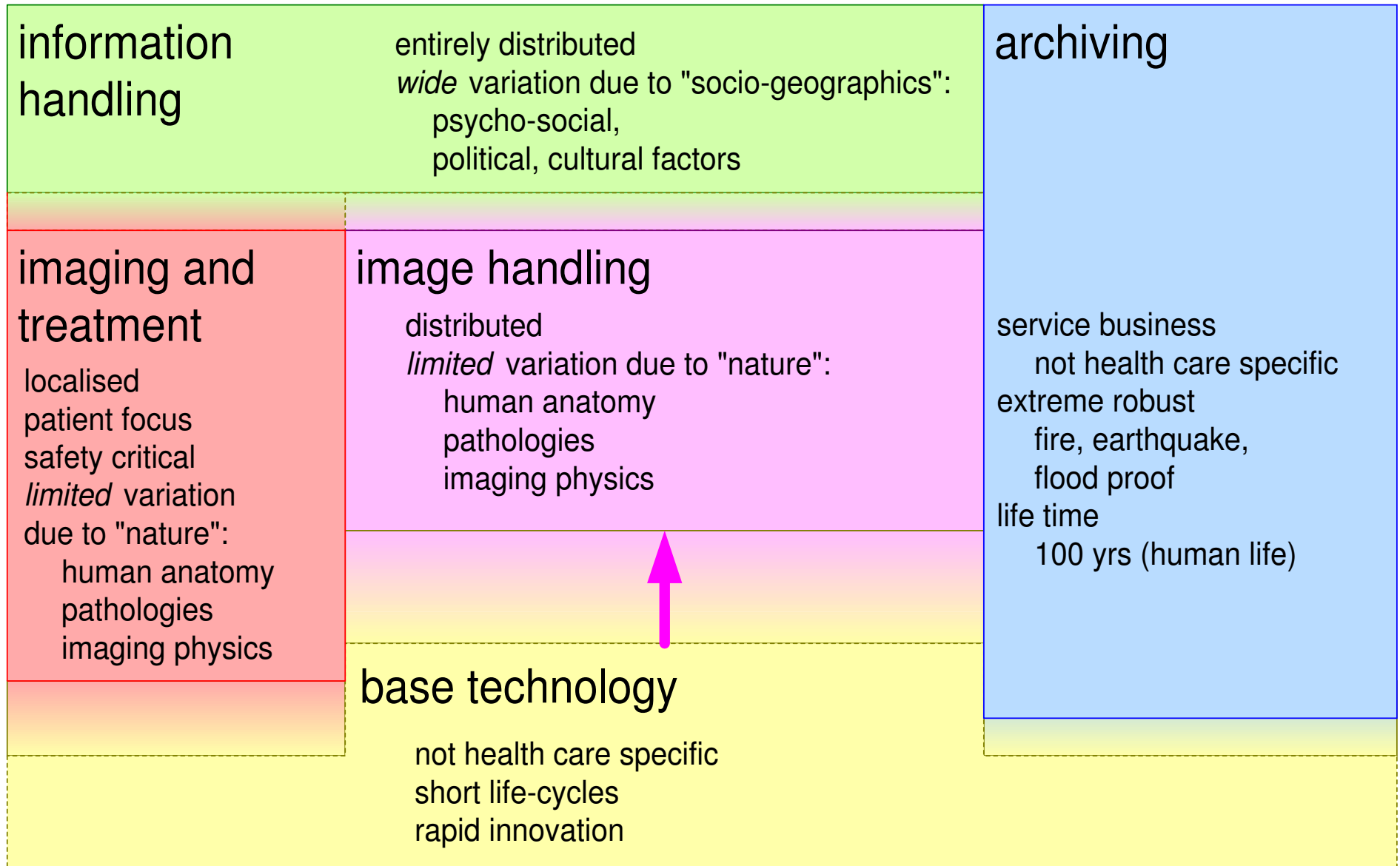
Platform Evolution (Easyvision 1991-1996)



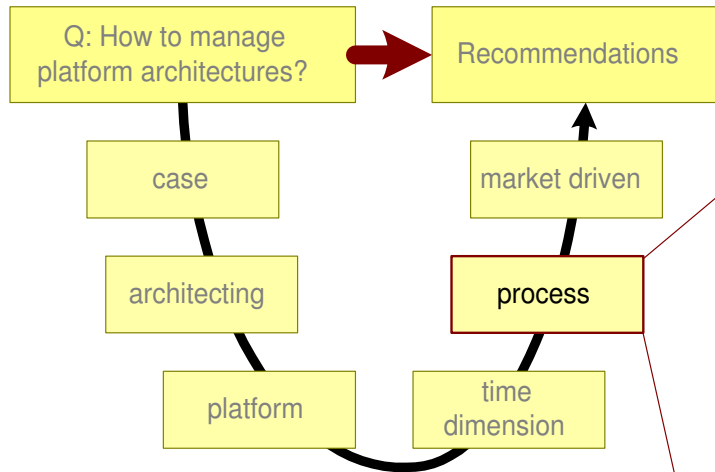
Lifecycle Differences



Reference Model for Healthcare Automation



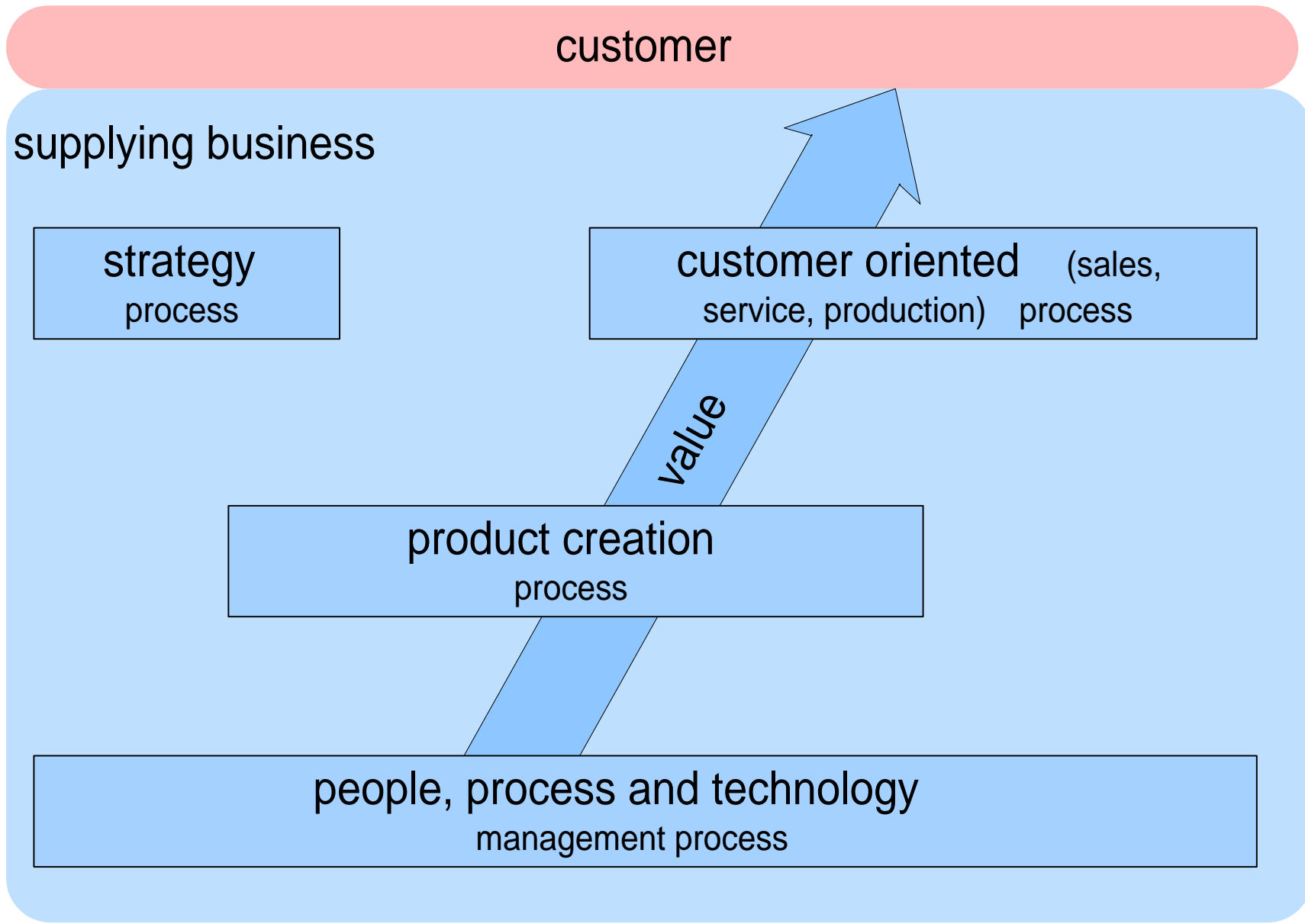
Process View Outline



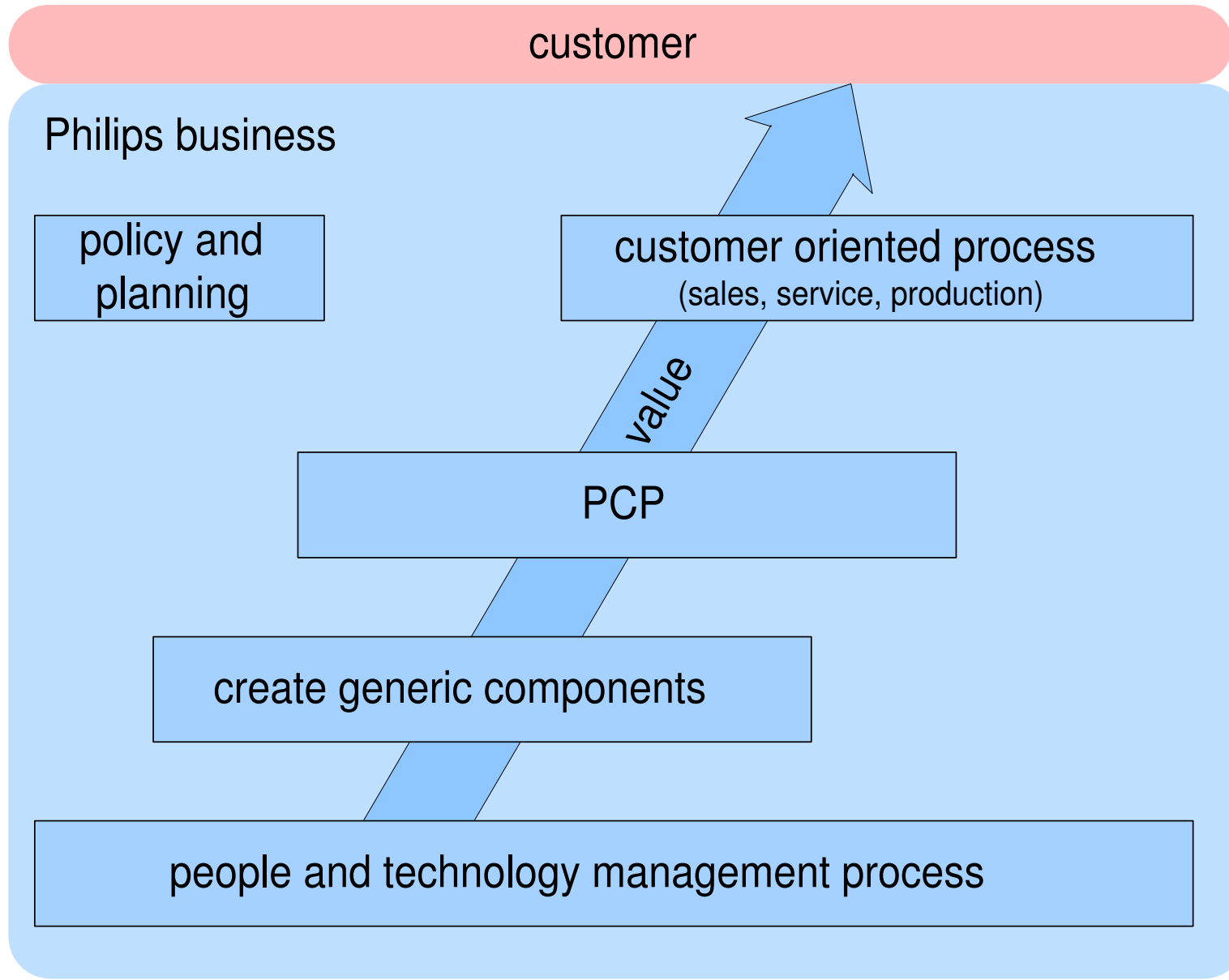
Process View:

- Simplified Process Decomposition
- Financial View on Process Decomposition
- Value and Feedback Flow
- Propagation Delay
- Sources of Failure
- Models for Generic Development

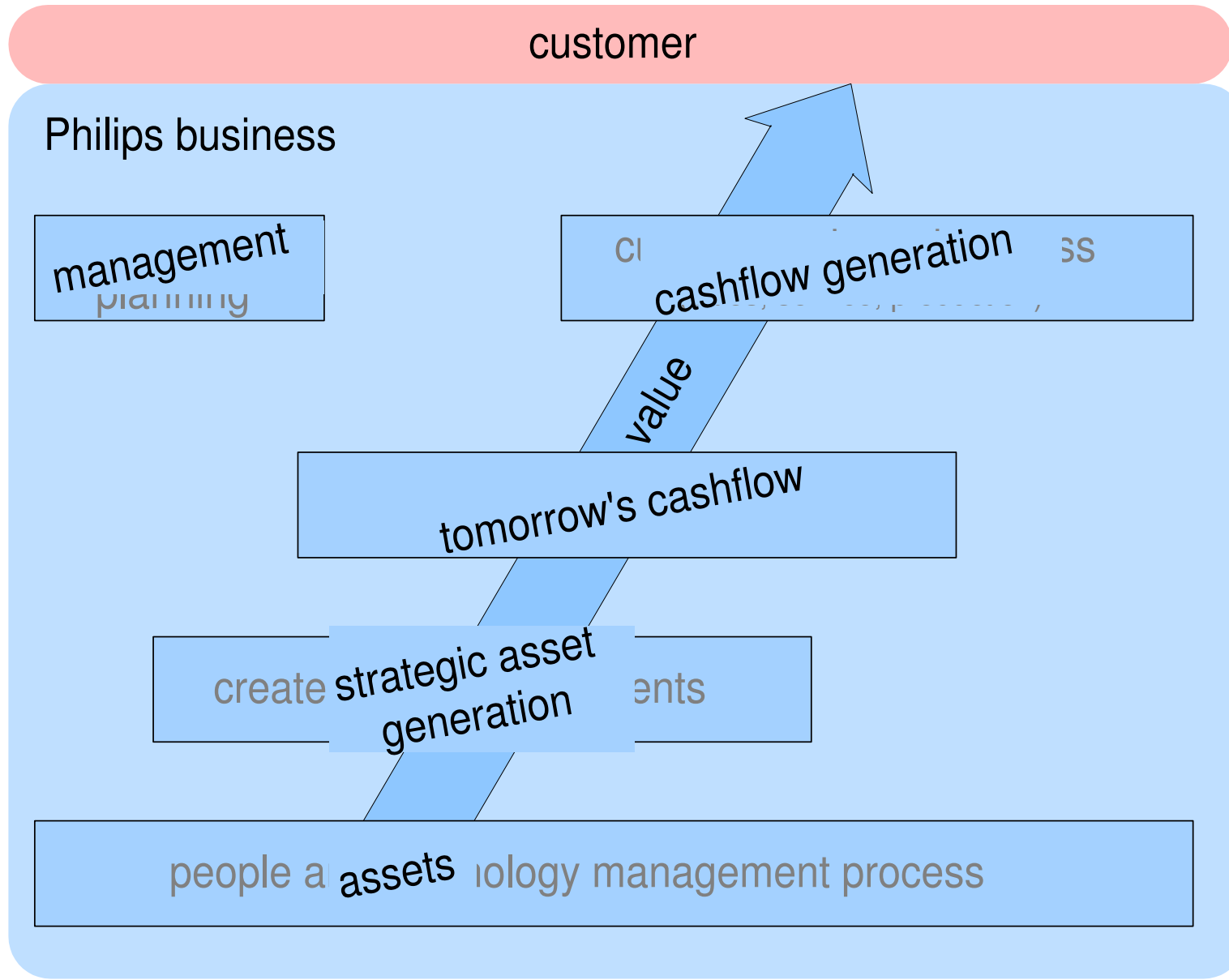
Simplified Process Decomposition



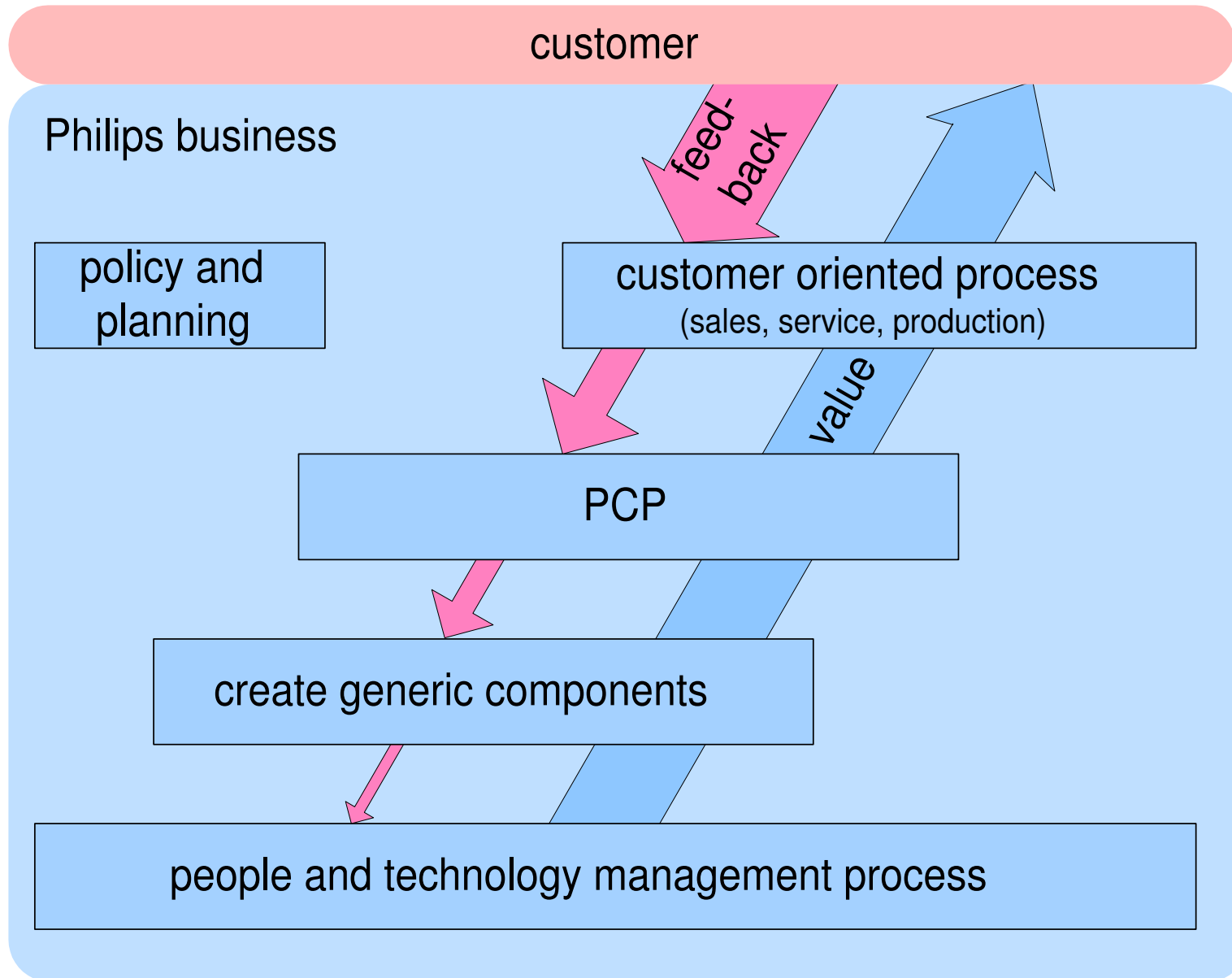
Modified Simplified Process Decomposition



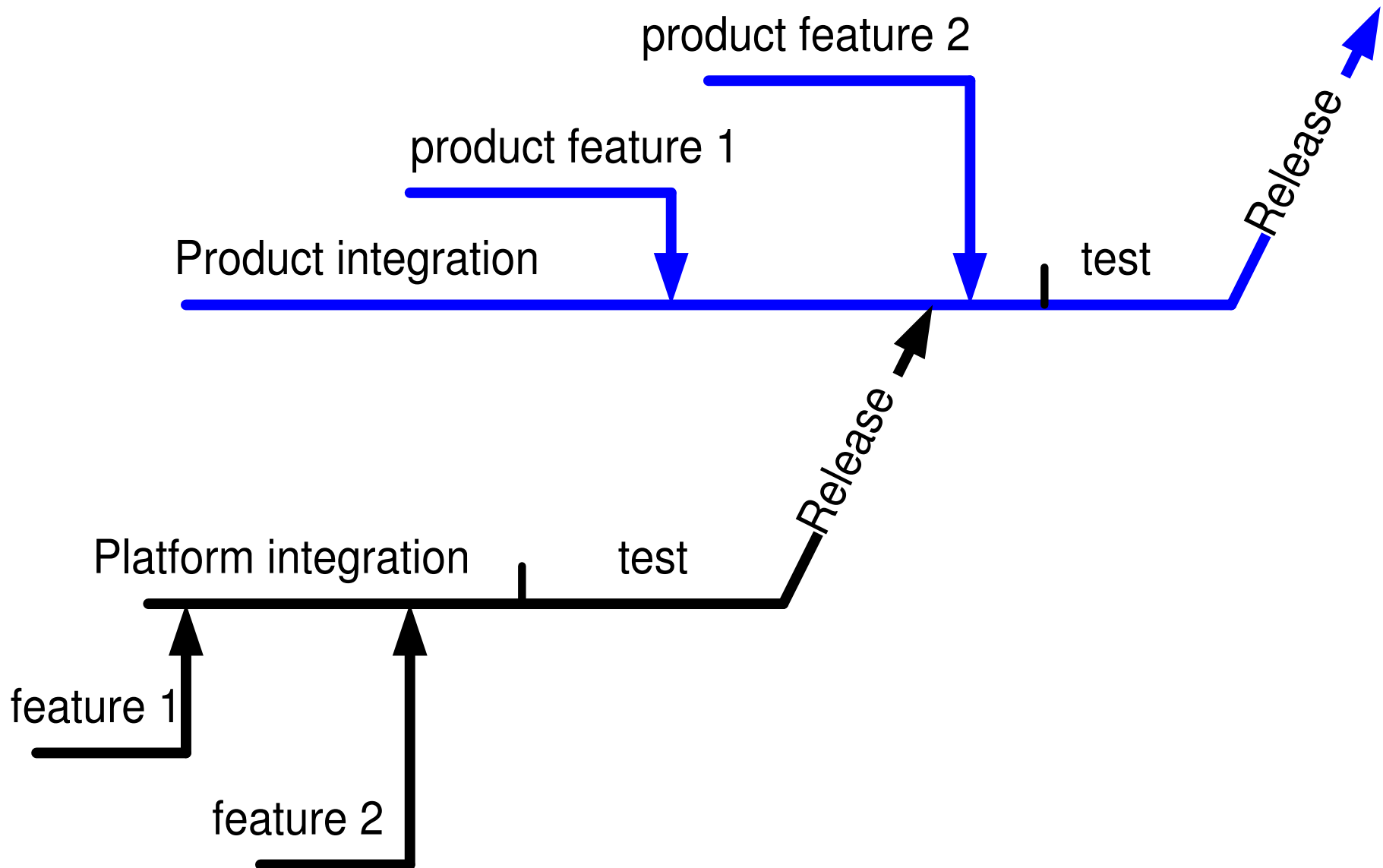
Financial View on Process Decomposition



Value and Feedback Flow



Propagation Delay Platform Feature to Market



Sources of Failure in Platform Developments

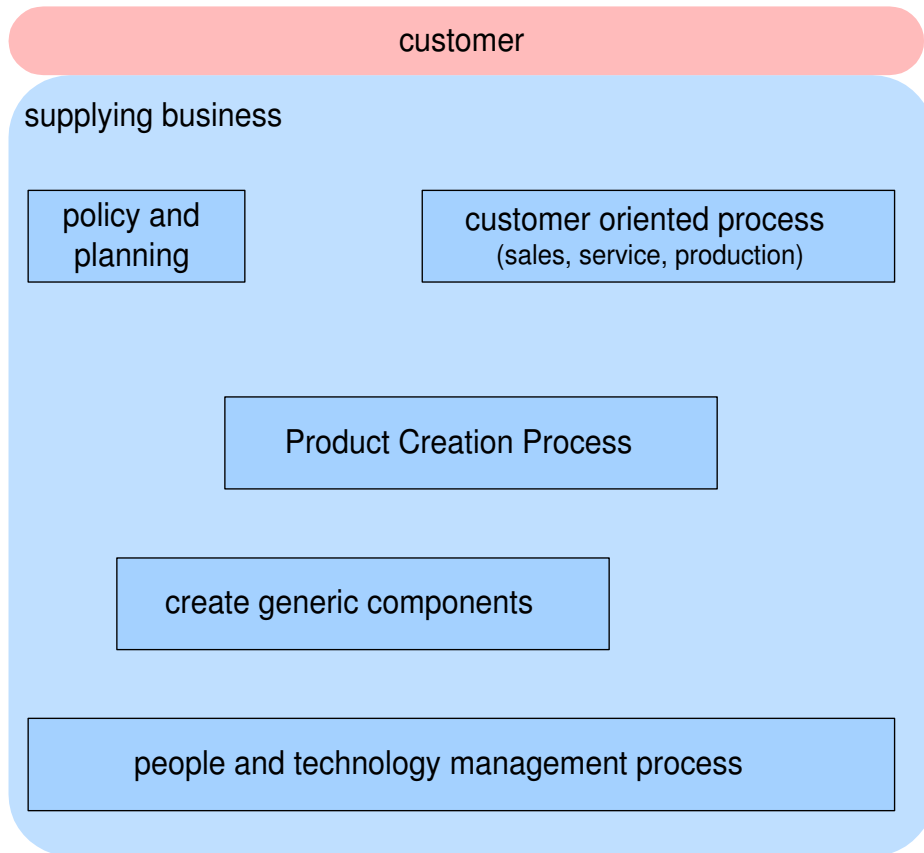
Technical

- Too generic
- Innovation stops
(stable interfaces)
- Vulnerability

Process/People/Organization

- Forced cooperation
- Time platform feature to market
- Unrealistic expectations
- Distance platform developer to customer
- No marketing ownership
- Bureaucratic process (no flexibility)
- New employees, knowledge dilution
- Underestimation of platform support
- Overstretching of product scope
- Nonmanagement, organizational scope increase
- Underestimation of integration
- Component/platform determines business policy
- Subcritical investment

Models for Platform Development



lead customer

direct feedback
too specific?

carrier product

product feedback
product specific?

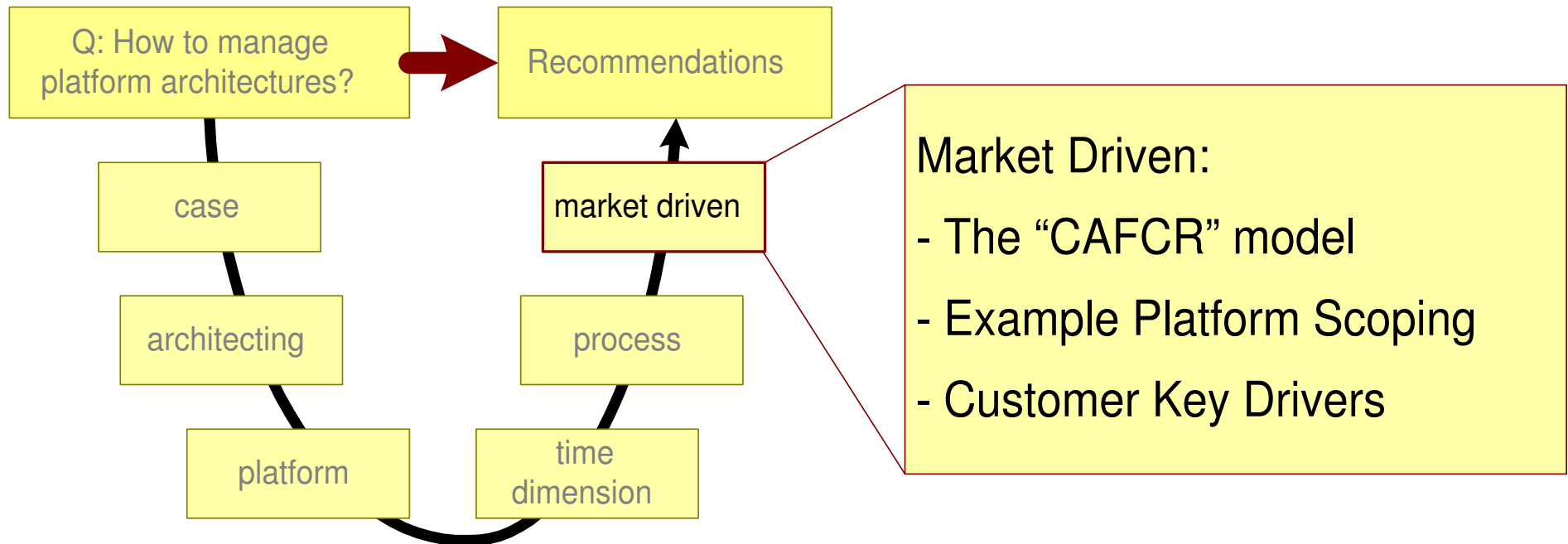
platform

feedback problem
too generic

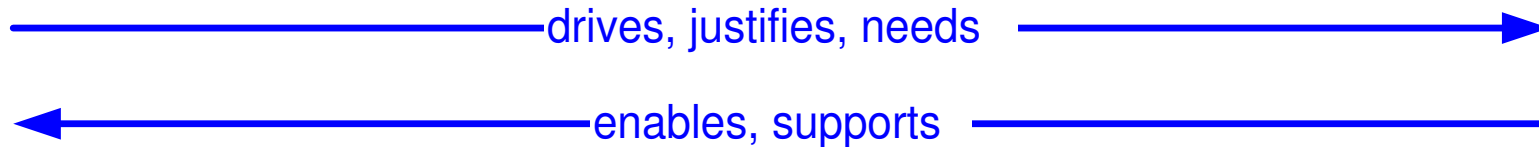
technology push

no feedback

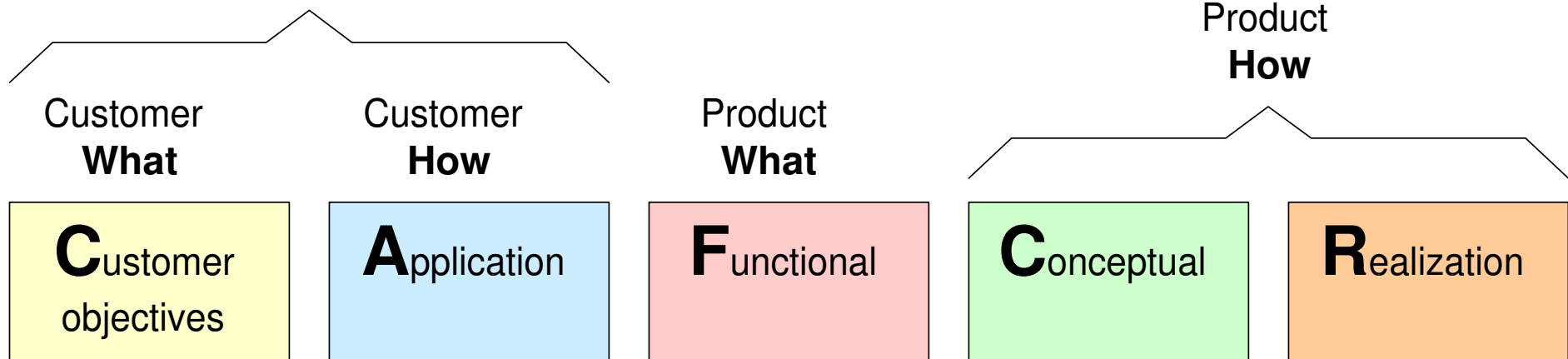
Market Driven Outline



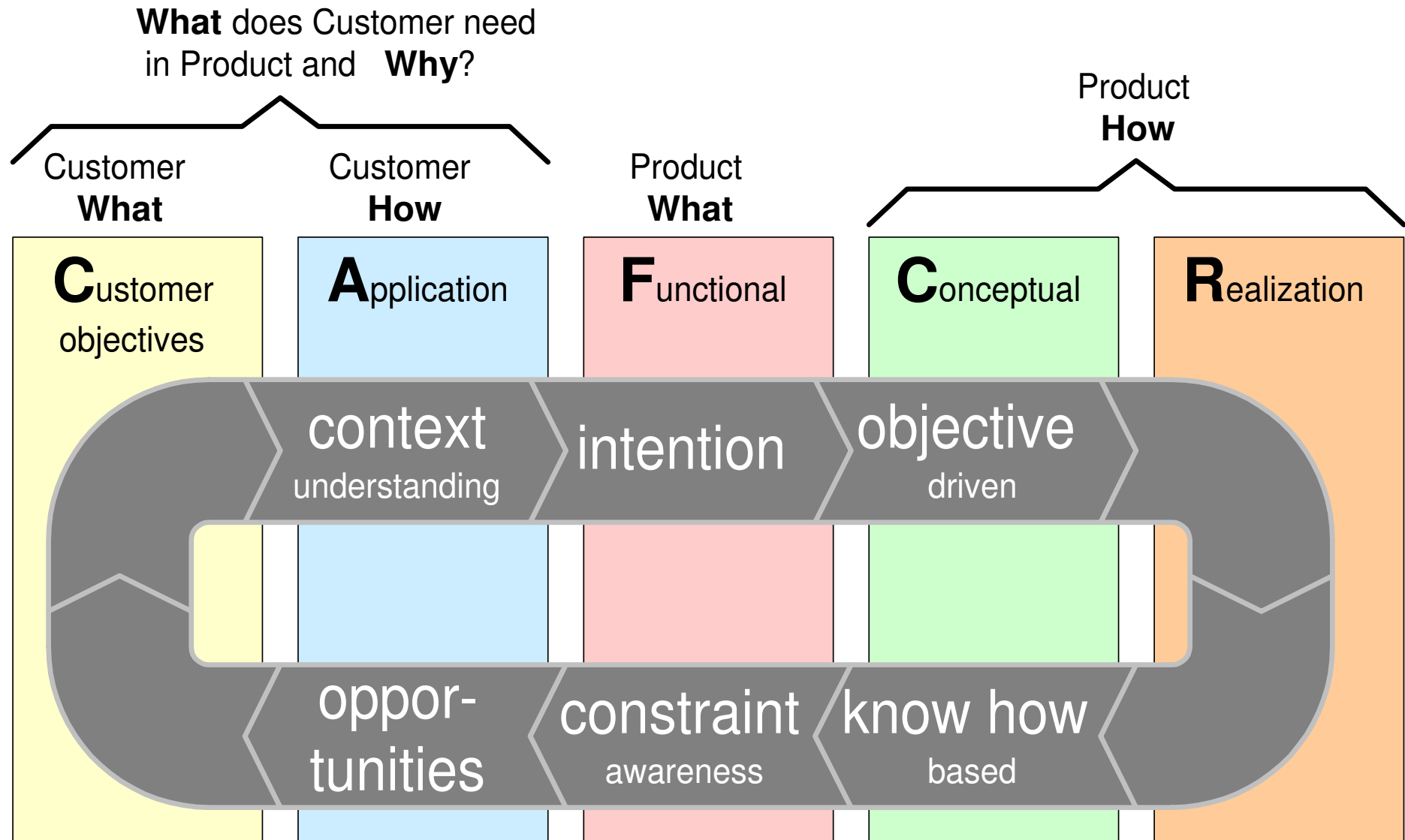
The "CAFCR" model



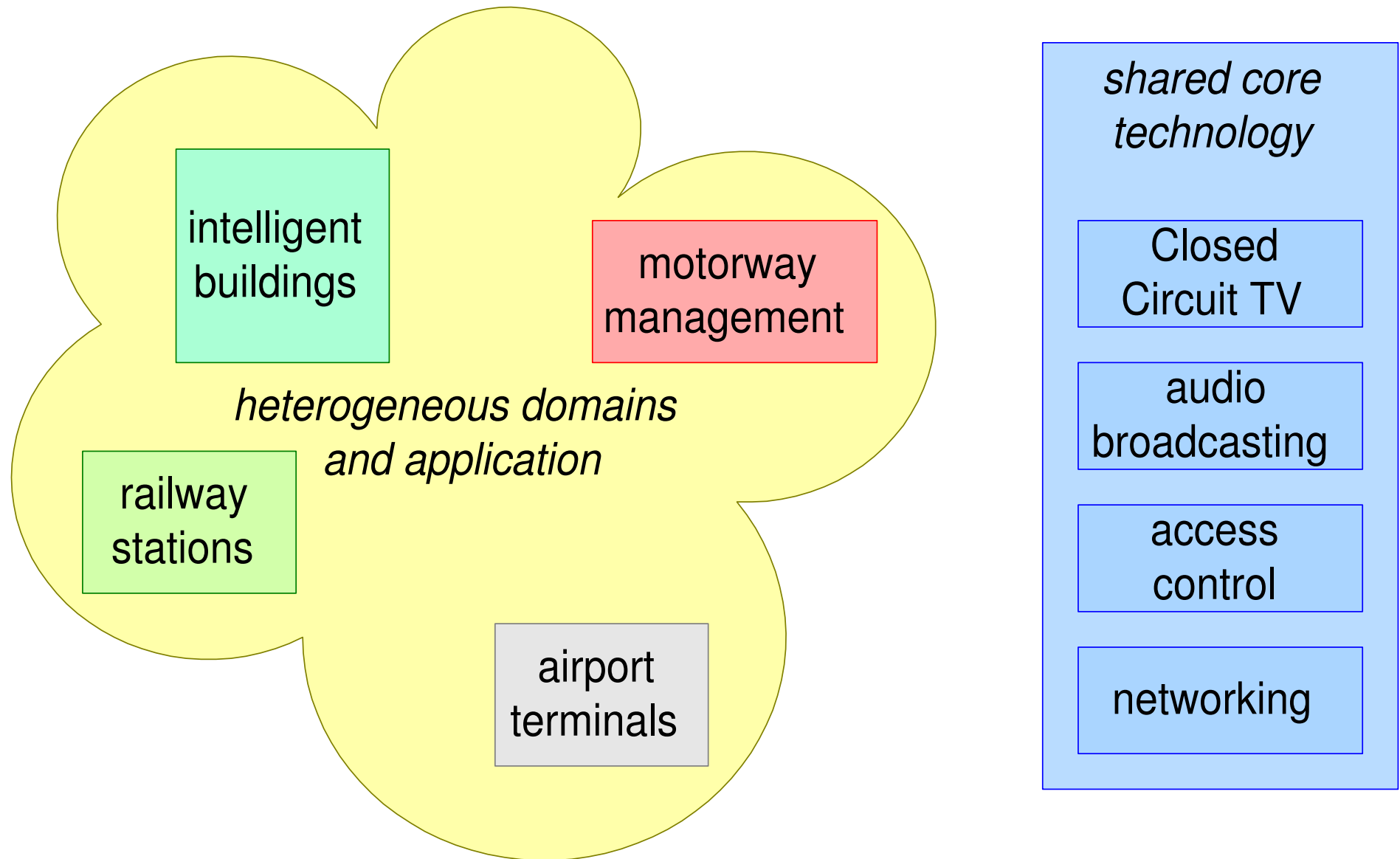
What does Customer need
in Product and **Why?**



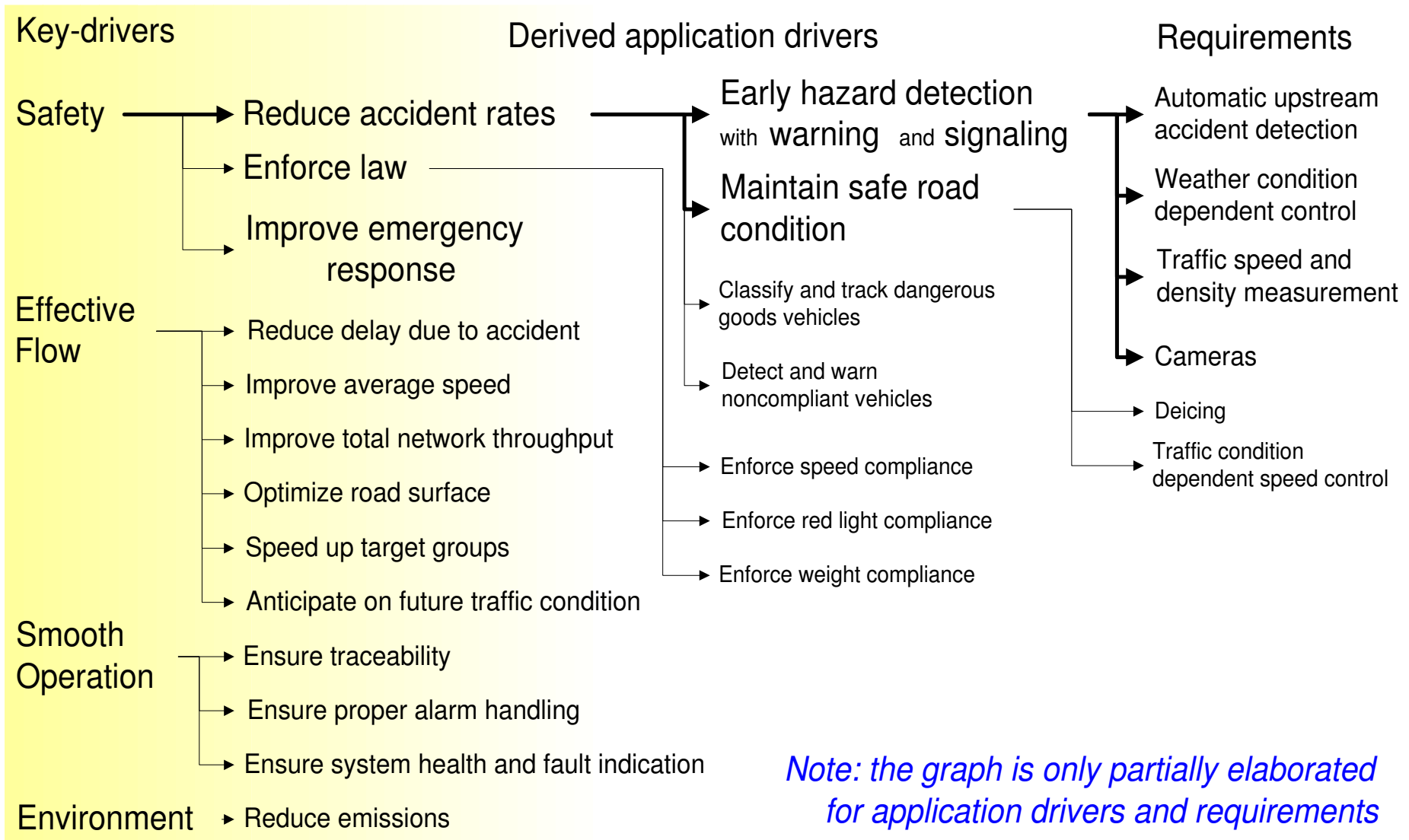
Five viewpoints for an architecture



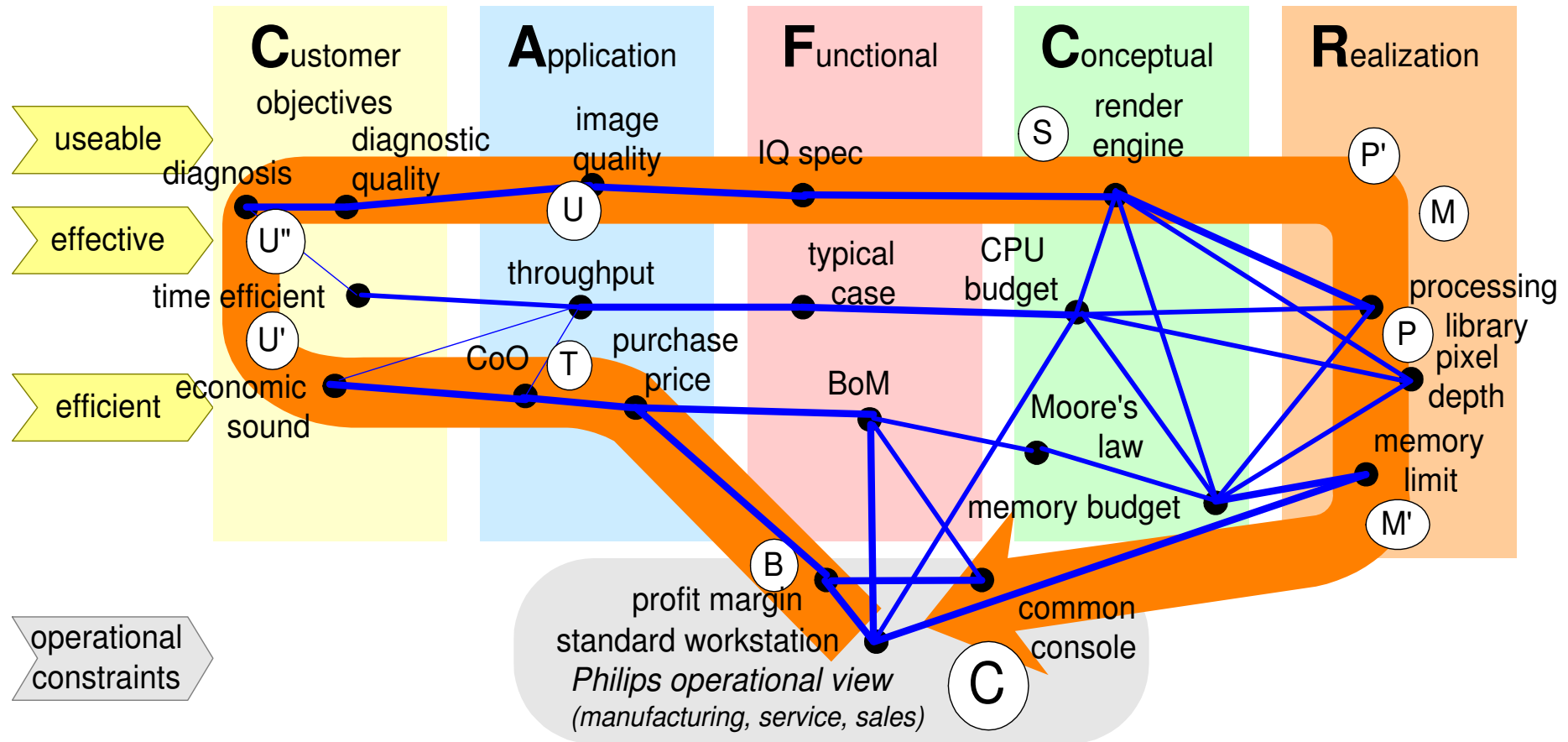
Example Platform Scoping



Customer Key Drivers Motorway Management



Finally All Design Decisions are Related to Market



cost revisited in context of clinical needs and realization constraints; note: original threads are significantly simplified

