

# Status of IT Architecting: Progression or Regression?

by *Gerrit Muller* Buskerud University College and Embedded Systems  
Institute

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

`www.gaudisite.nl`

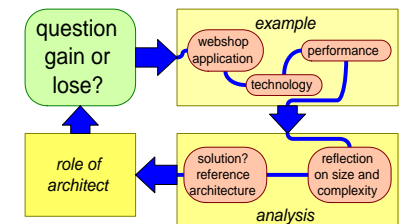
## Abstract

Today's IT capabilities are seemingly limitless. From the point of view of last century we have amazing functionality available to consumers, businesses, governments et cetera. Technology advances have made this possible. At the same time we suffer from unwanted, unexpected incidents, ranging from slow or no response to loss or theft of sensitive data. The growth of systems and its complexity play a role. We will look at the role of the human creators of these systems and the available technology to discuss our concurrent progression and regression, and we will look at the role of the architect in particular.

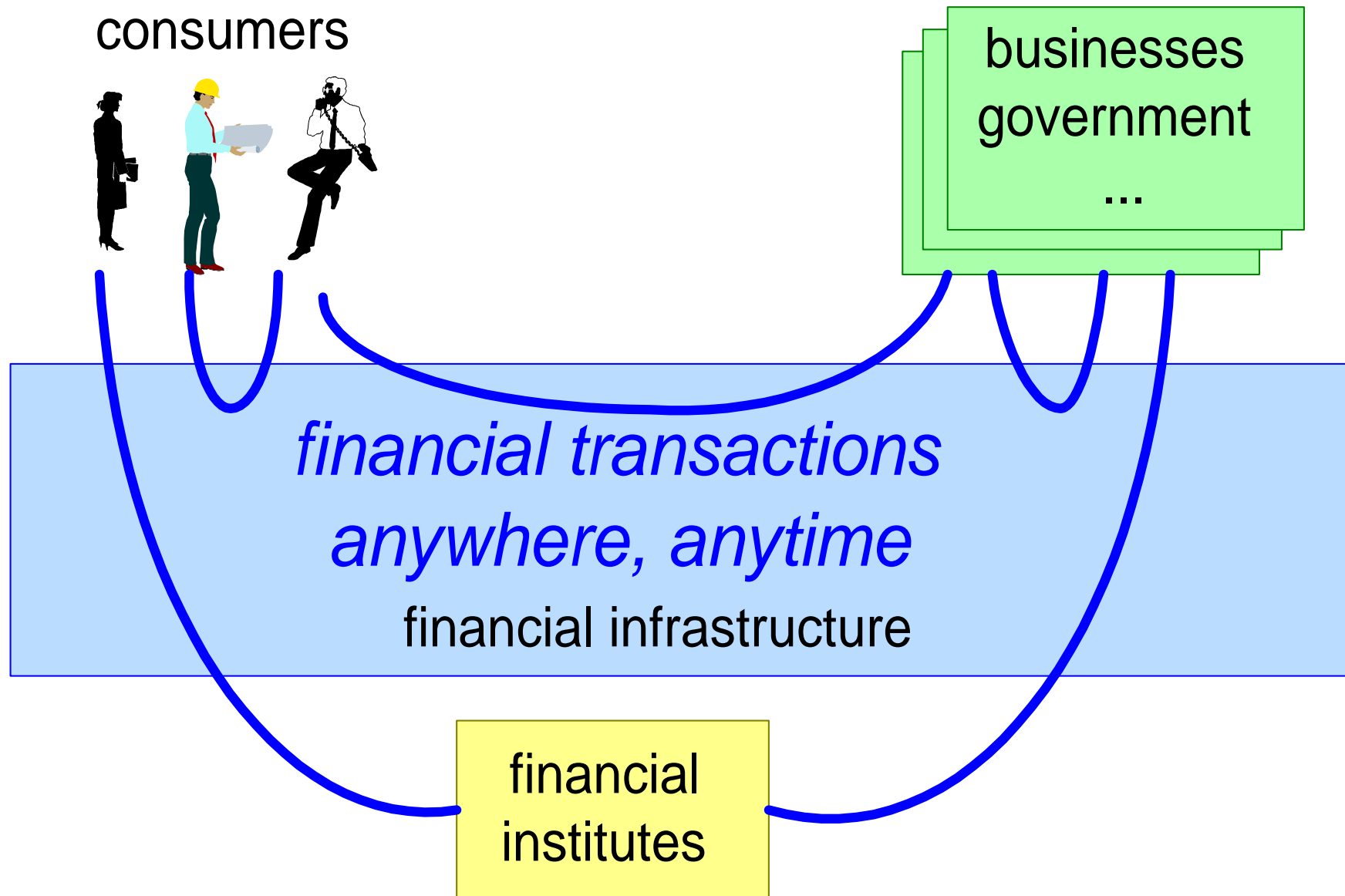
### Distribution

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 20, 2011  
status: preliminary  
draft  
version: 0



# Functionality is Limitless



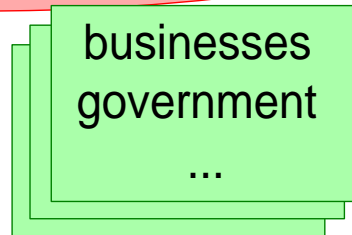
# But Problems seem to be Pervasive

slow response, outages, human-less helpdesks, silly excuses (the computer could not...), identity-theft, lost privacy

consumers



businesses  
government  
...



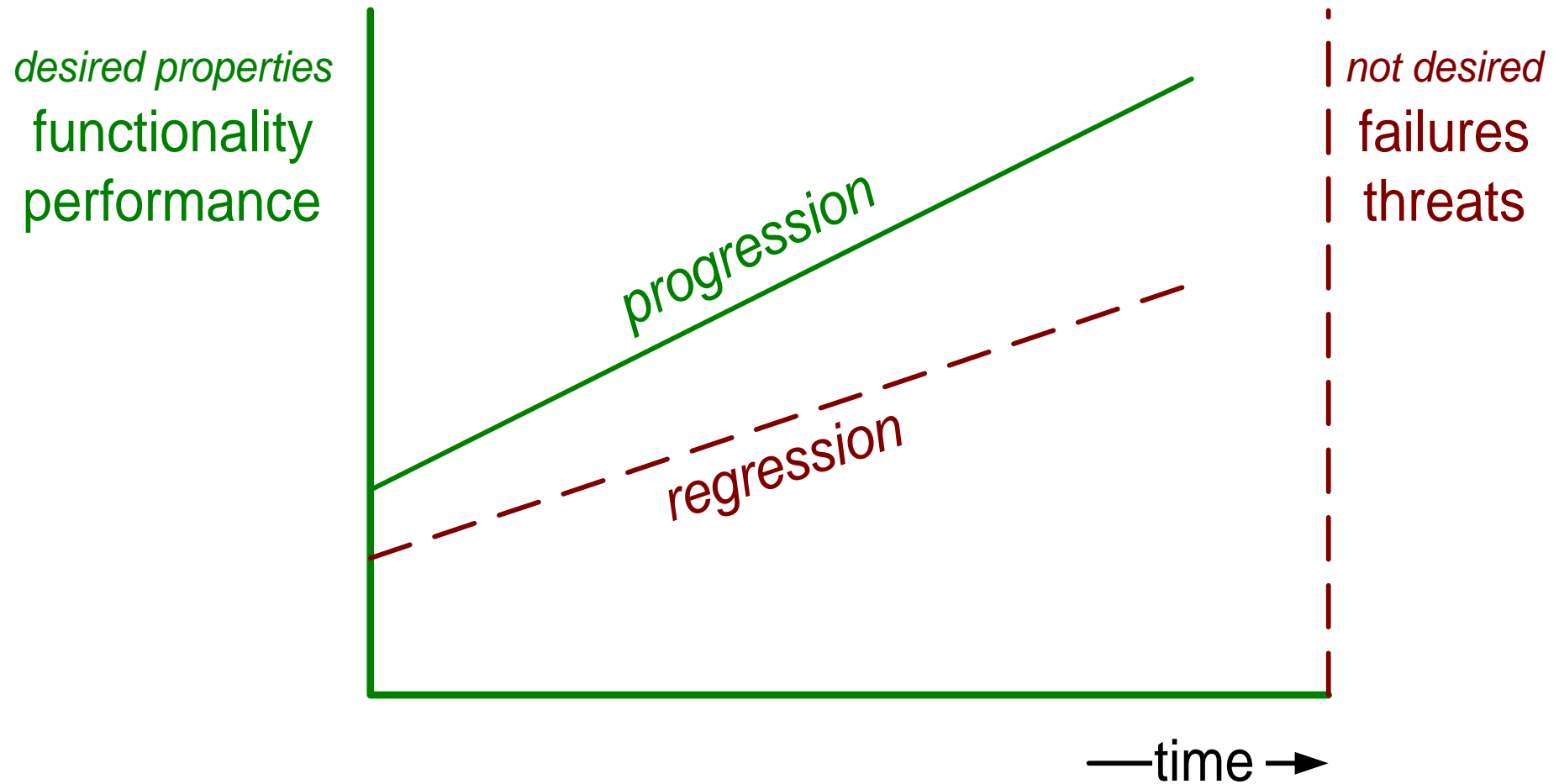
*financial transac*  
*anywhere*  
financial in

late delivery of new products,  
poor scaling of new services,  
interference of features,  
...

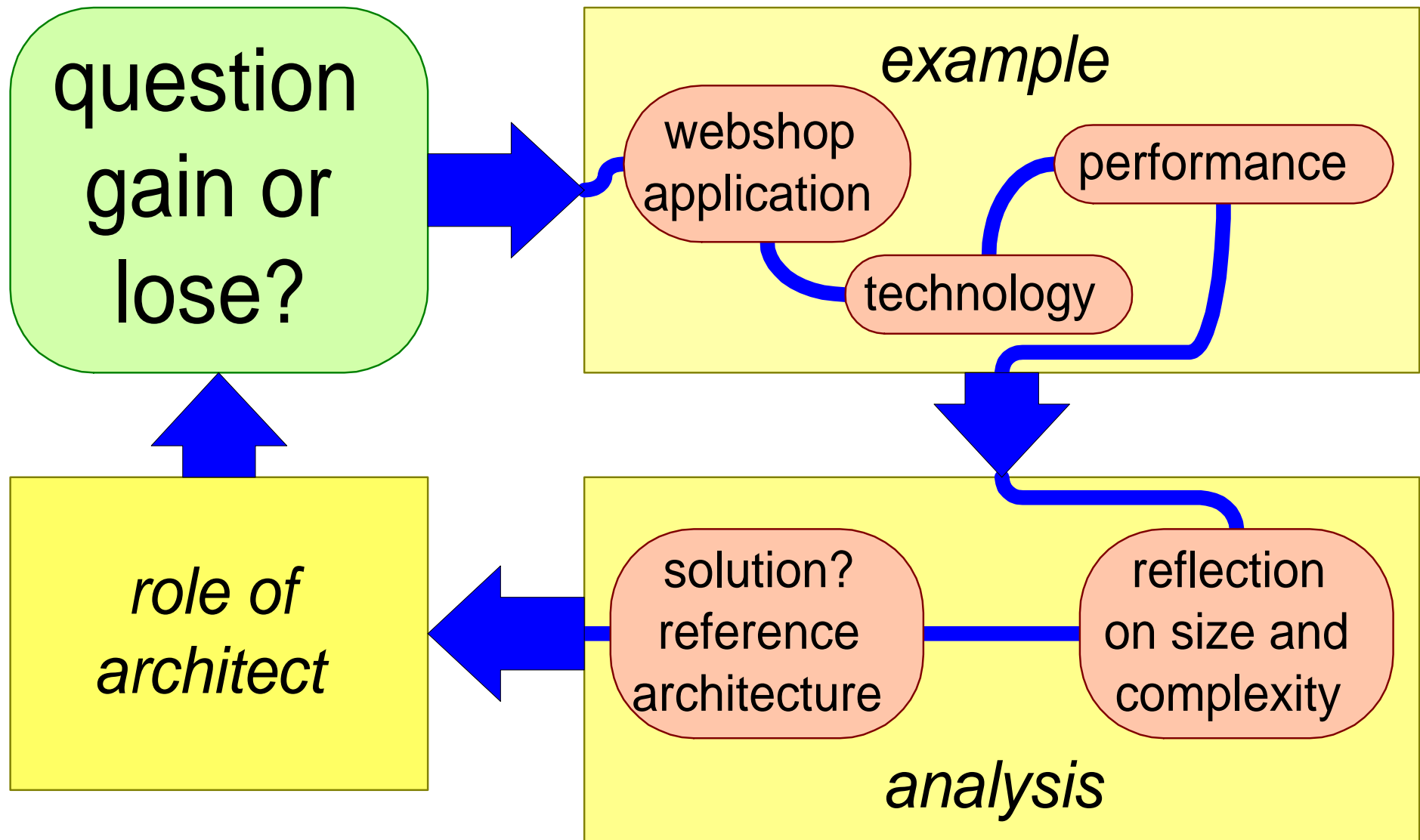
"entrepreneurial"  
employees

financial  
institutes

# Do we Gain or do we Lose?



# Figure Of Contents™



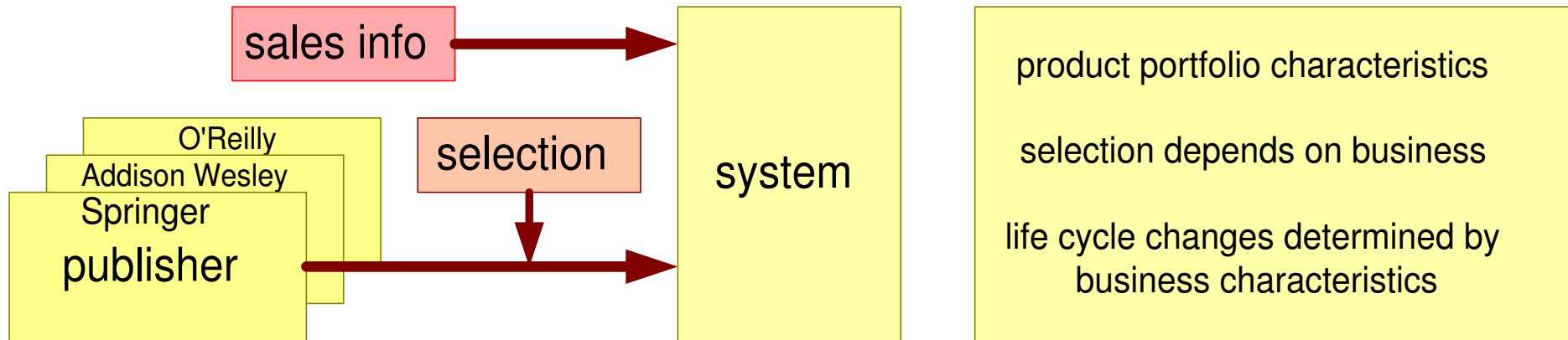
# Example, Case Webshop

The image shows a screenshot of the Amazon.com website in a Mozilla Firefox browser window. Several yellow callout boxes are overlaid on the page to highlight specific features:

- main access through search**: Points to the search bar at the top of the page.
- personalization**: Points to the text "Hello. Sign in to get personalized recommendations. New customer? Start here." with a red arrow.
- catalogue entries**: A vertical box on the left side pointing to the "Browse" menu.
- styling: frequently updated, fashion!**: A box on the right side pointing to the promotional banners and product images.
- other advertisements**: A vertical box on the right side pointing to the "Extreme Savings on..." and "Free Stand with Bow..." banners.
- Up-to-date information: Bestsellers**: A box pointing to the "Books Bestsellers" section.
- What Other Customers Are Looking At Right Now**: A box pointing to the "What Other Customers Are Looking At Right Now" section.
- standard boilerplate**: A box at the bottom pointing to the footer area containing links like "Directory of All Stores", "Canada", "United Kingdom", etc.

snapshot of  
www.amazon.com

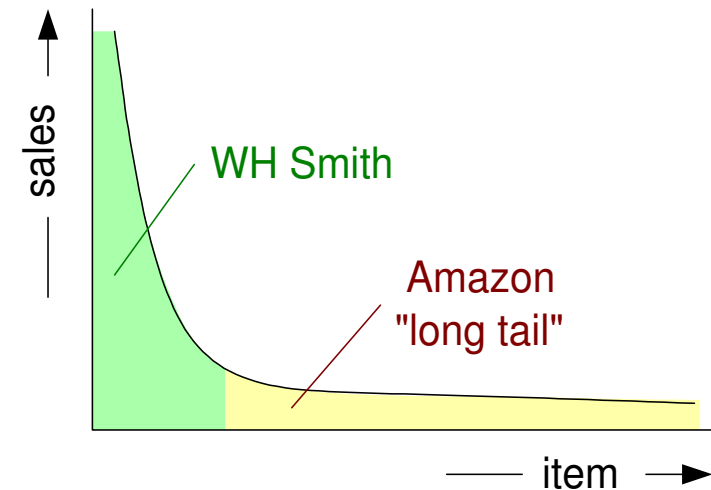
# Some Numbers: New Books per Year



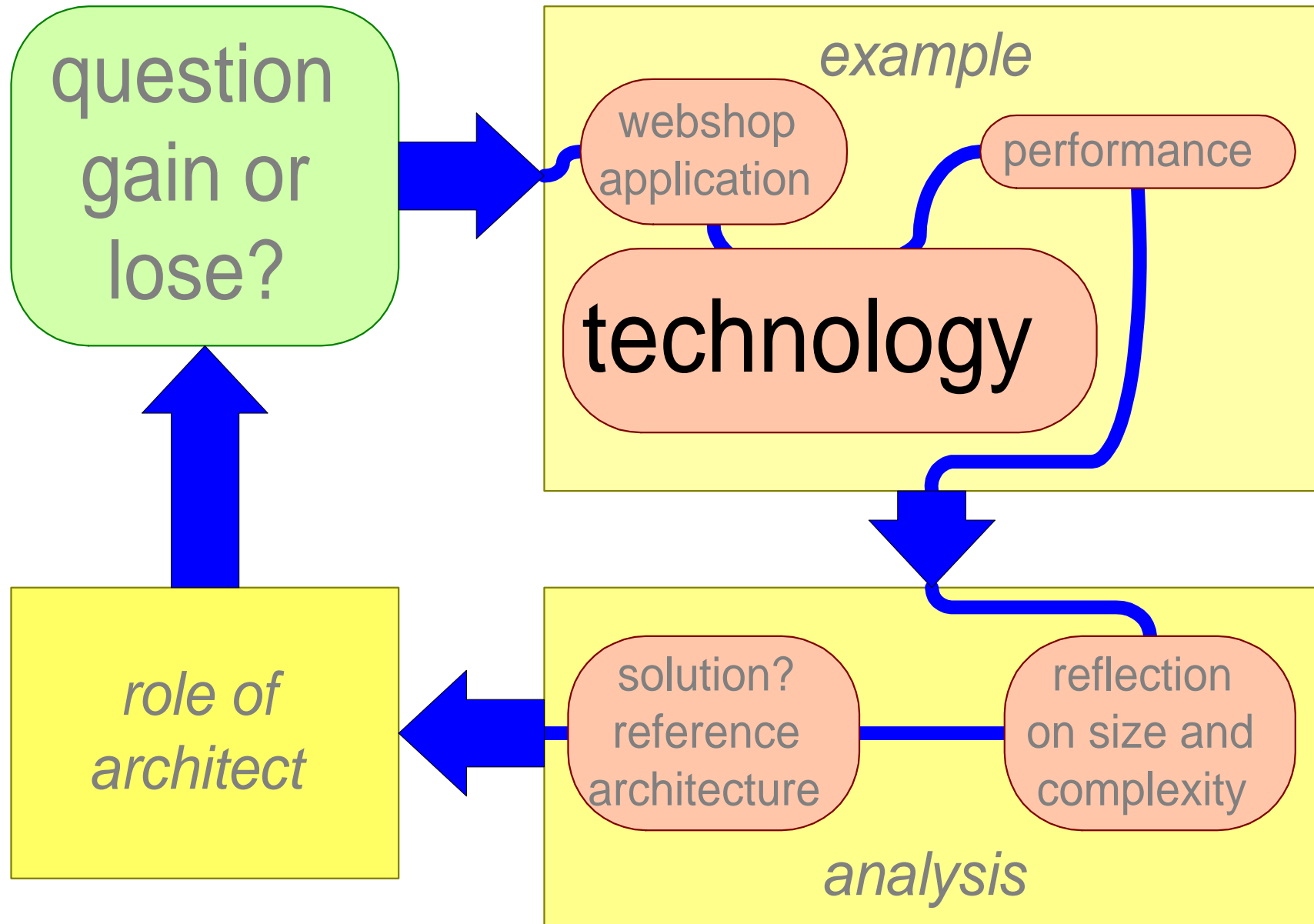
source: [http://en.wikipedia.org/wiki/Long\\_tail](http://en.wikipedia.org/wiki/Long_tail)

## new books per year

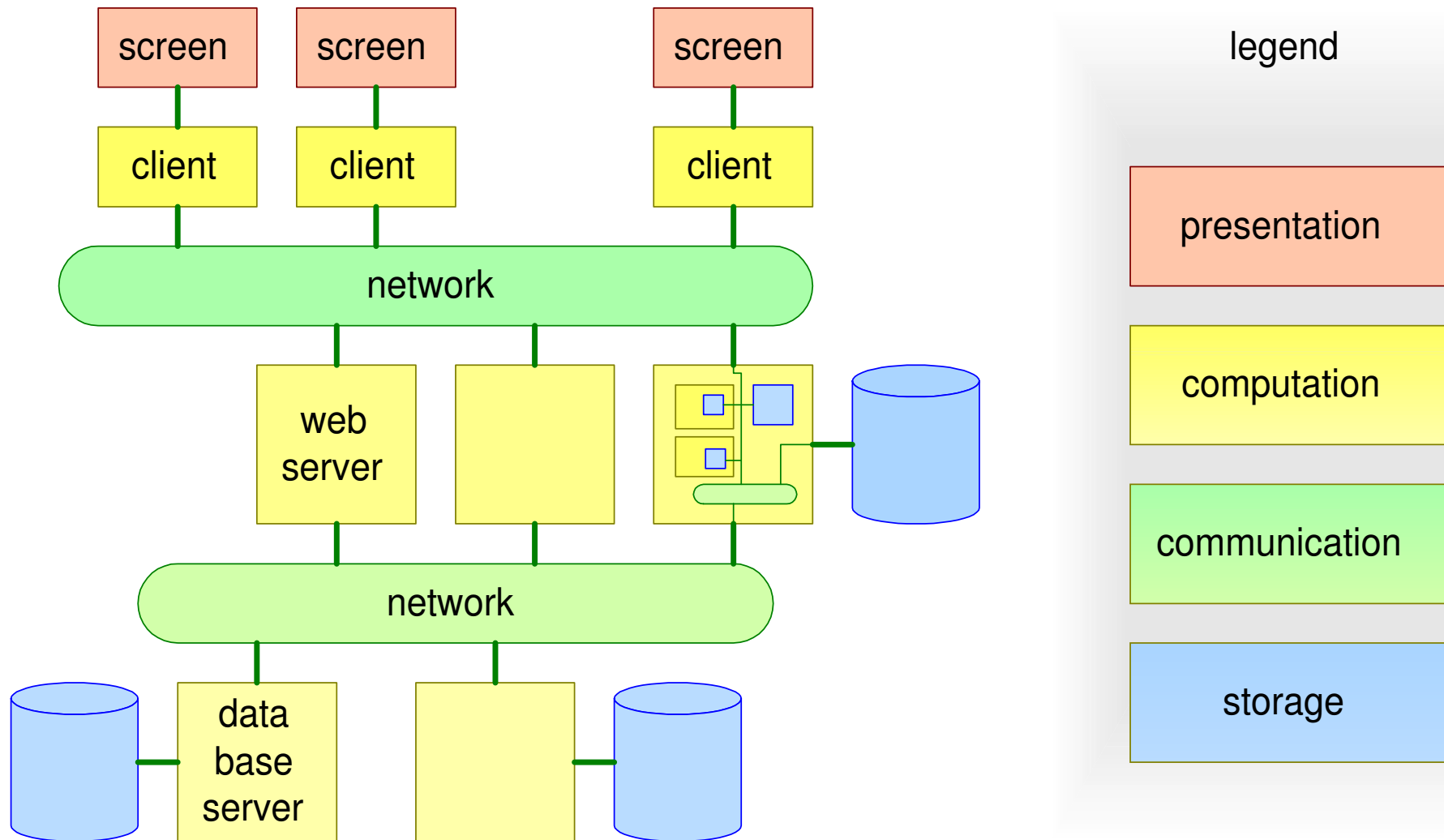
|           |             |             |
|-----------|-------------|-------------|
| UK (1)    | 206k (2005) | 107k (1996) |
| USA(2)    | 172k (2005) | 68k (1996)  |
| China(3)  |             | 101k (1994) |
| India(21) |             | 12k (1996)  |



source: [http://en.wikipedia.org/wiki/Books\\_published\\_per\\_country\\_per\\_year](http://en.wikipedia.org/wiki/Books_published_per_country_per_year)



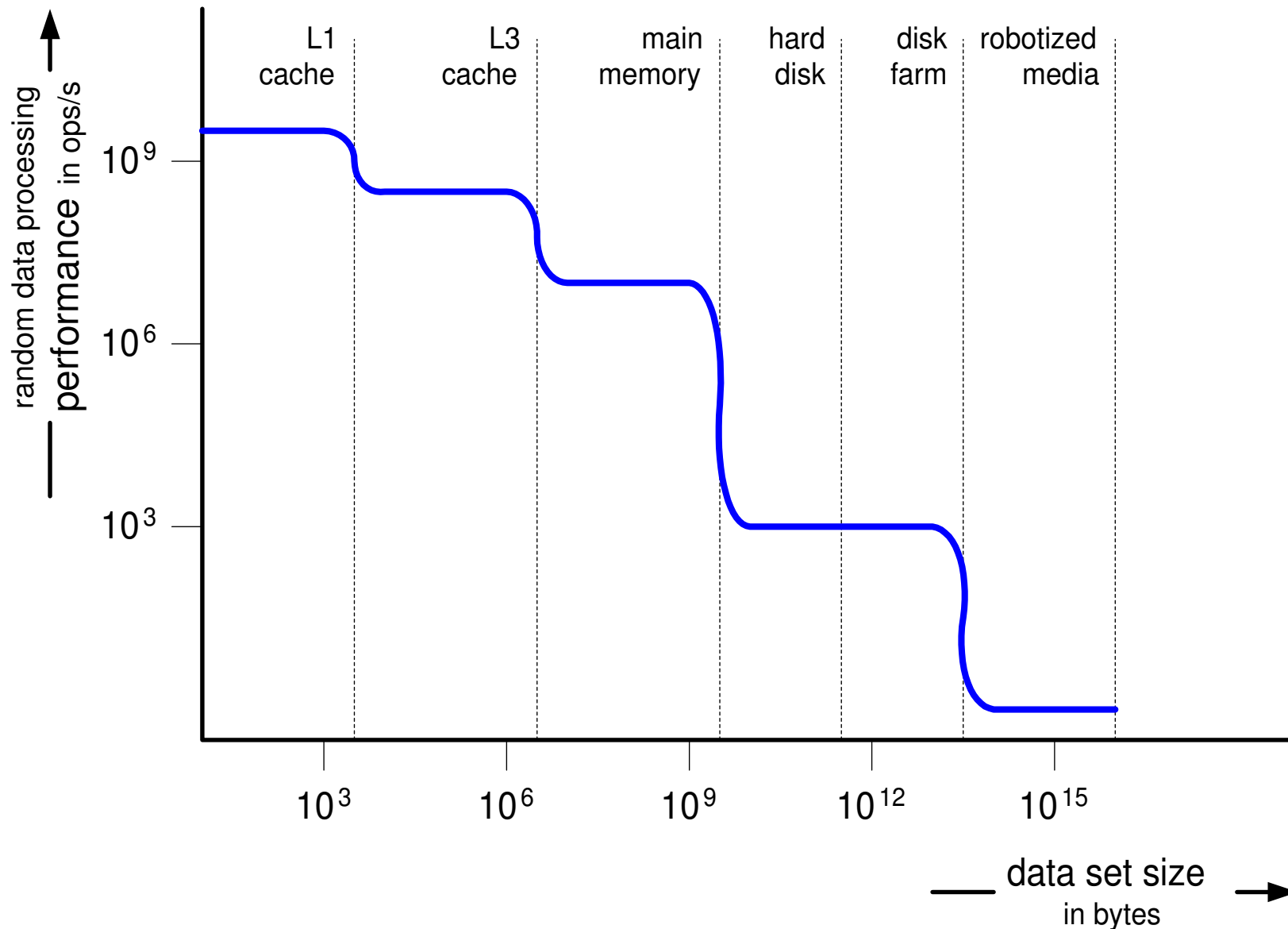
# Typical Block Diagram and Typical Resources



# Hierarchy of Storage Technology Figures of Merit

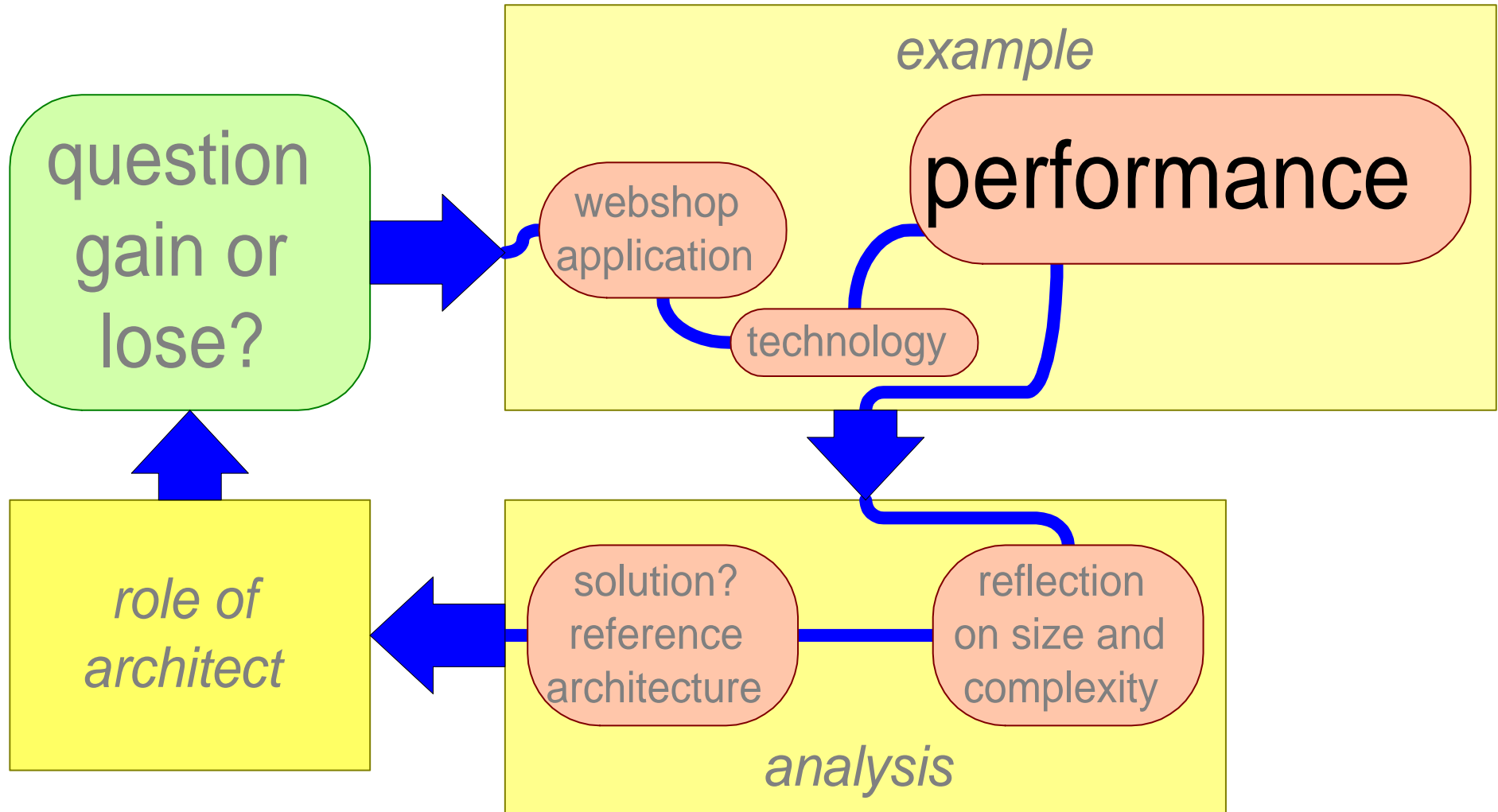
|                  |   | latency | capacity |
|------------------|---|---------|----------|
| processor cache  | <i>L1 cache</i>                             | sub ns  | n kB     |
|                  | <i>L2 cache</i>                             |         |          |
|                  | <i>L3 cache</i>                             | ns      | n MB     |
| fast<br>volatile | <i>main memory</i>                          | tens ns | n GB     |
| persistent       | <i>disks</i>                                |         | n*100 GB |
|                  | <i>disk arrays</i>                          | ms      |          |
|                  | <i>disk farms</i>                           |         | n*10 TB  |
| archival         | <i>robotized<br/>optical media<br/>tape</i> | >s      | n PB     |

# Performance as Function of Data Set Size

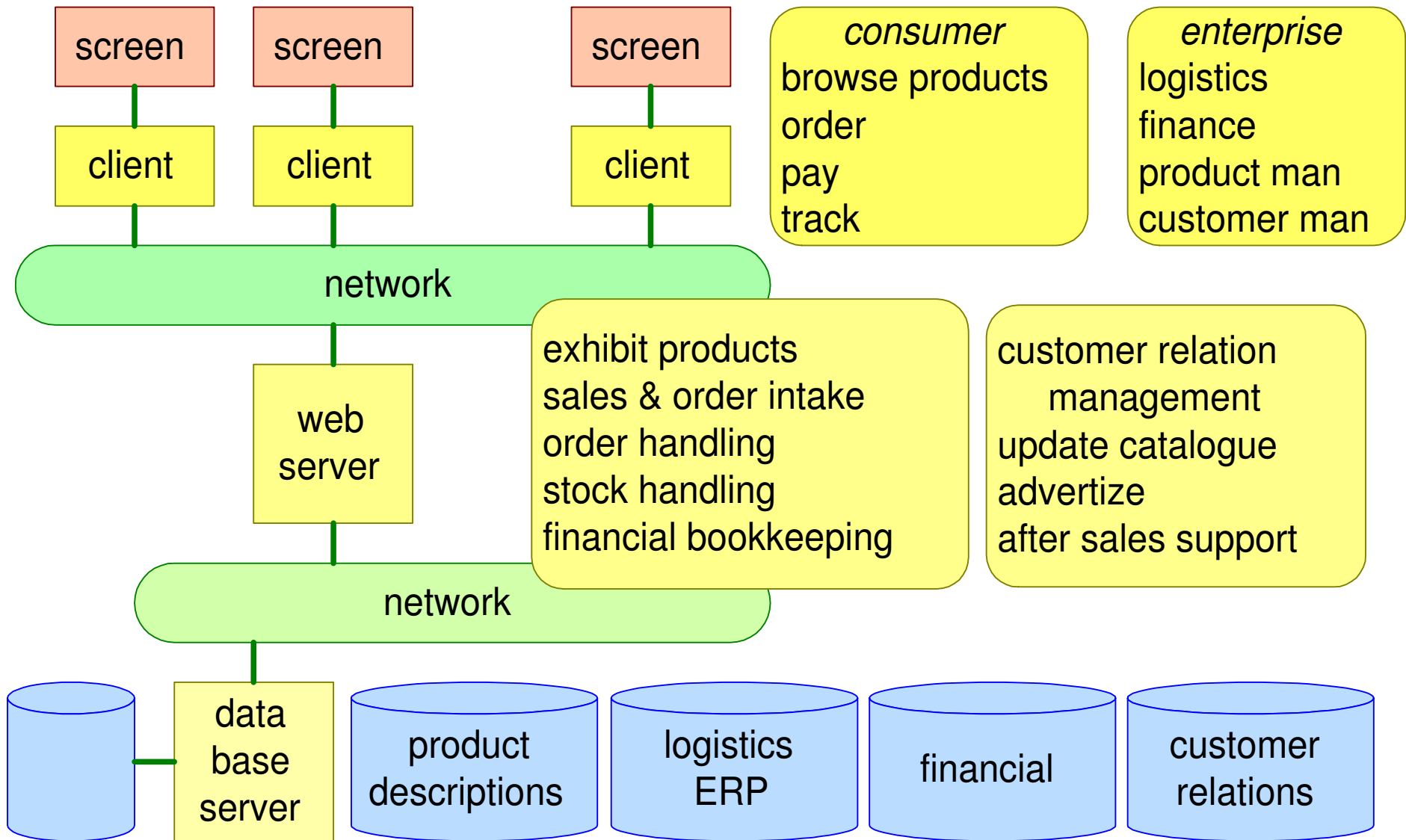


|            |                   | latency | frequency | distance |
|------------|-------------------|---------|-----------|----------|
| on chip    | <i>connection</i> | sub ns  | n GHz     | n mm     |
|            | <i>network</i>    | n ns    | n GHz     | n mm     |
| PCB level  |                   | tens ns | n 100MHz  | n cm     |
| Serial I/O |                   | n ms    | n 100MHz  | n m      |
| network    | <i>LAN</i>        | n ms    | 100MHz    | n km     |
|            | <i>WAN</i>        | n 10ms  | n GHz     | global   |

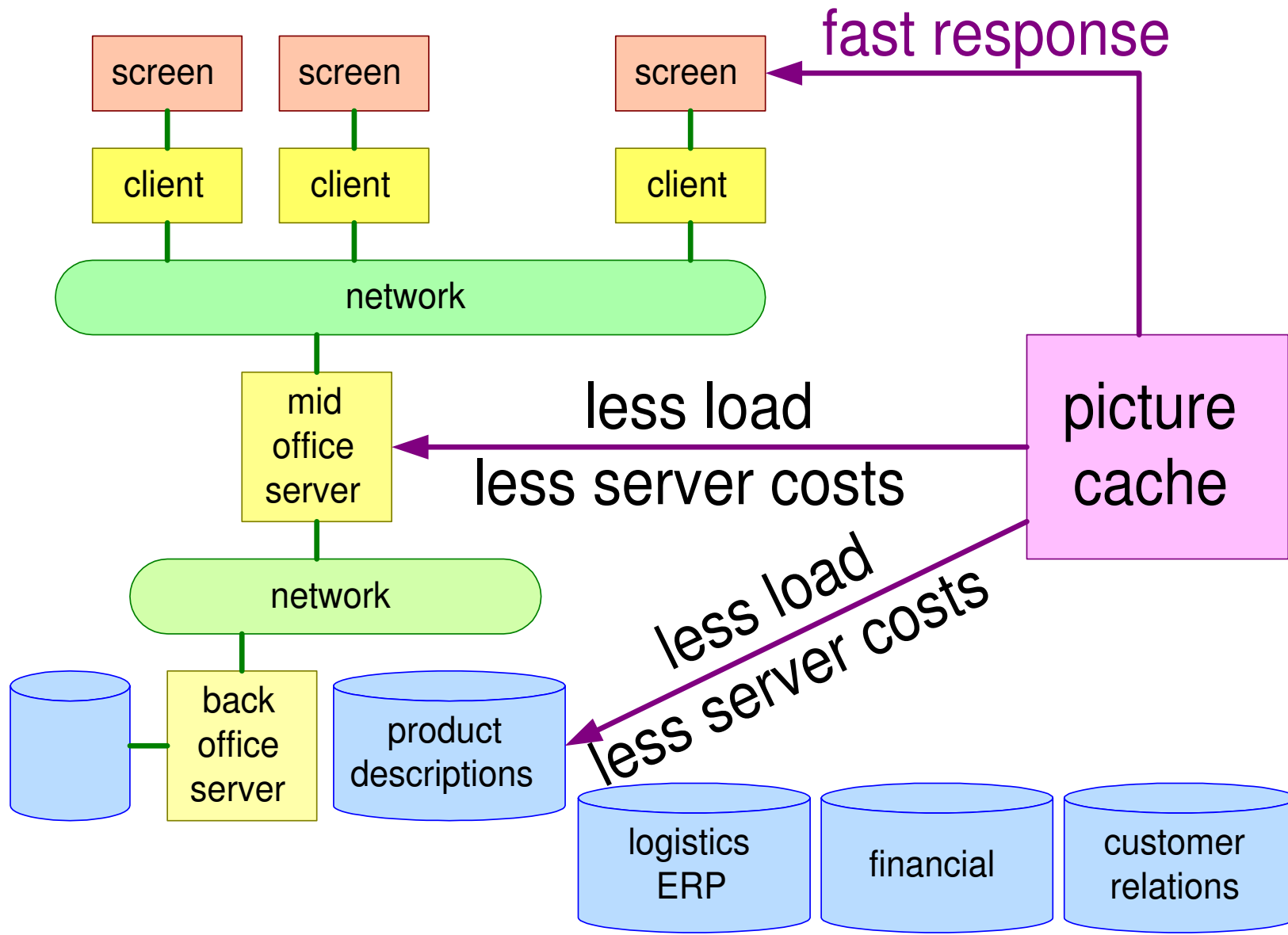
# Performance



# Example Web Shop

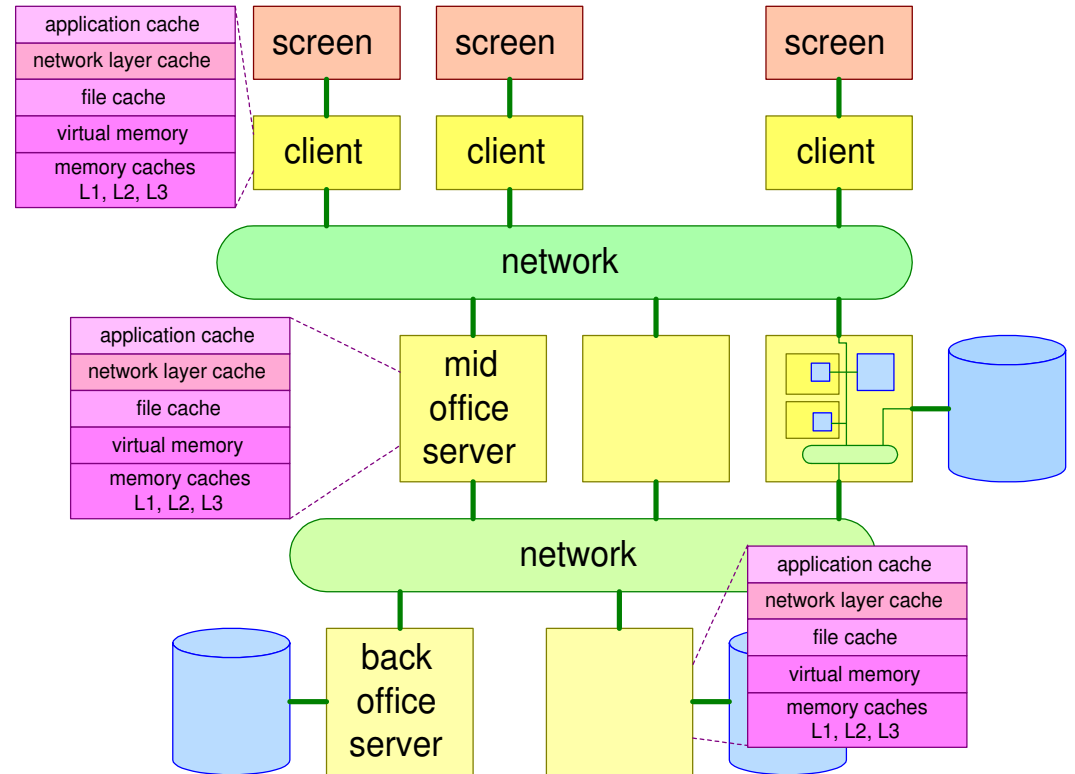


# Impact of Picture Cache

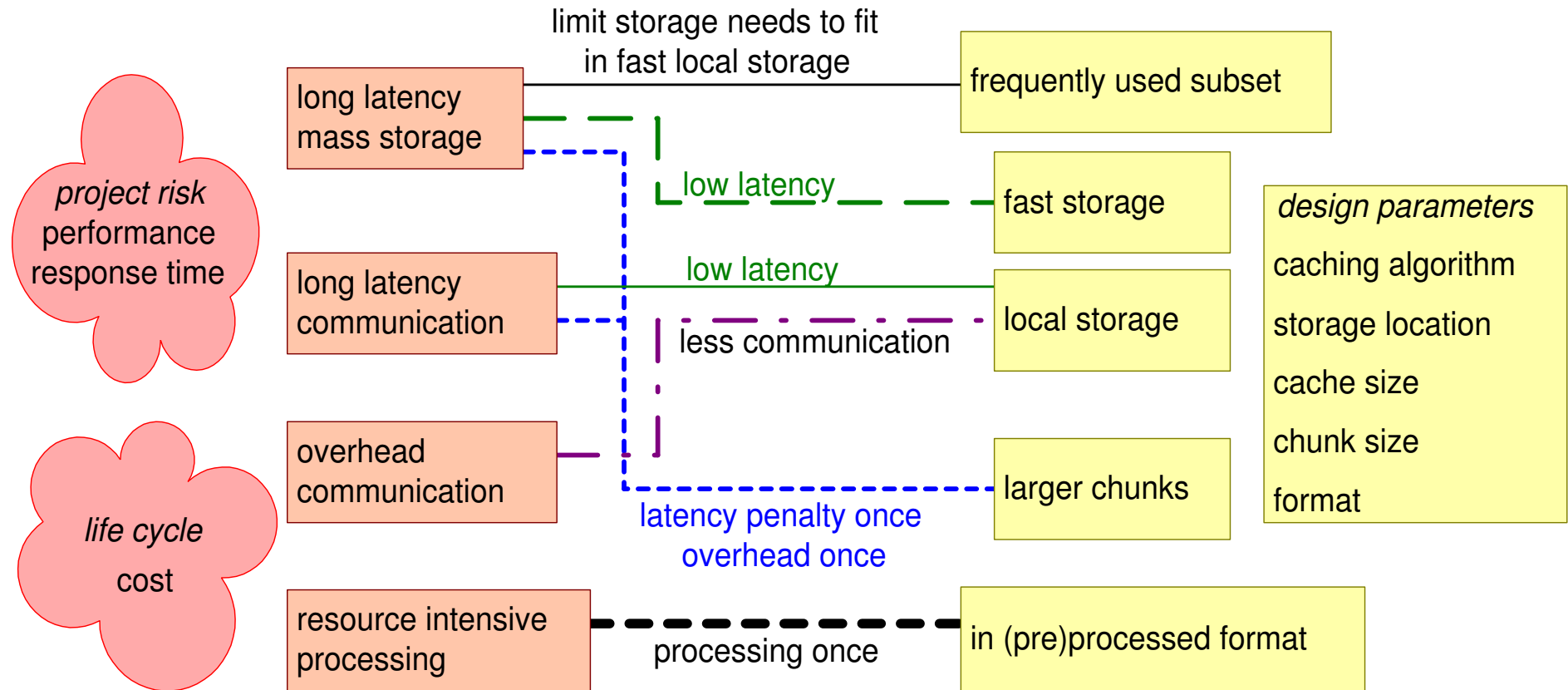


# Multiple Layers of Caching

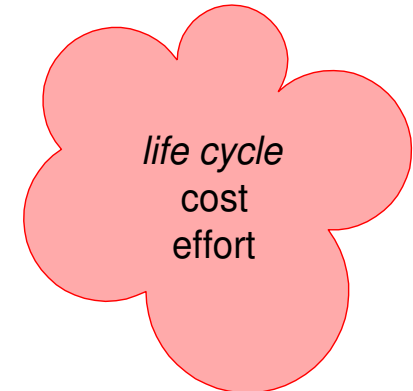
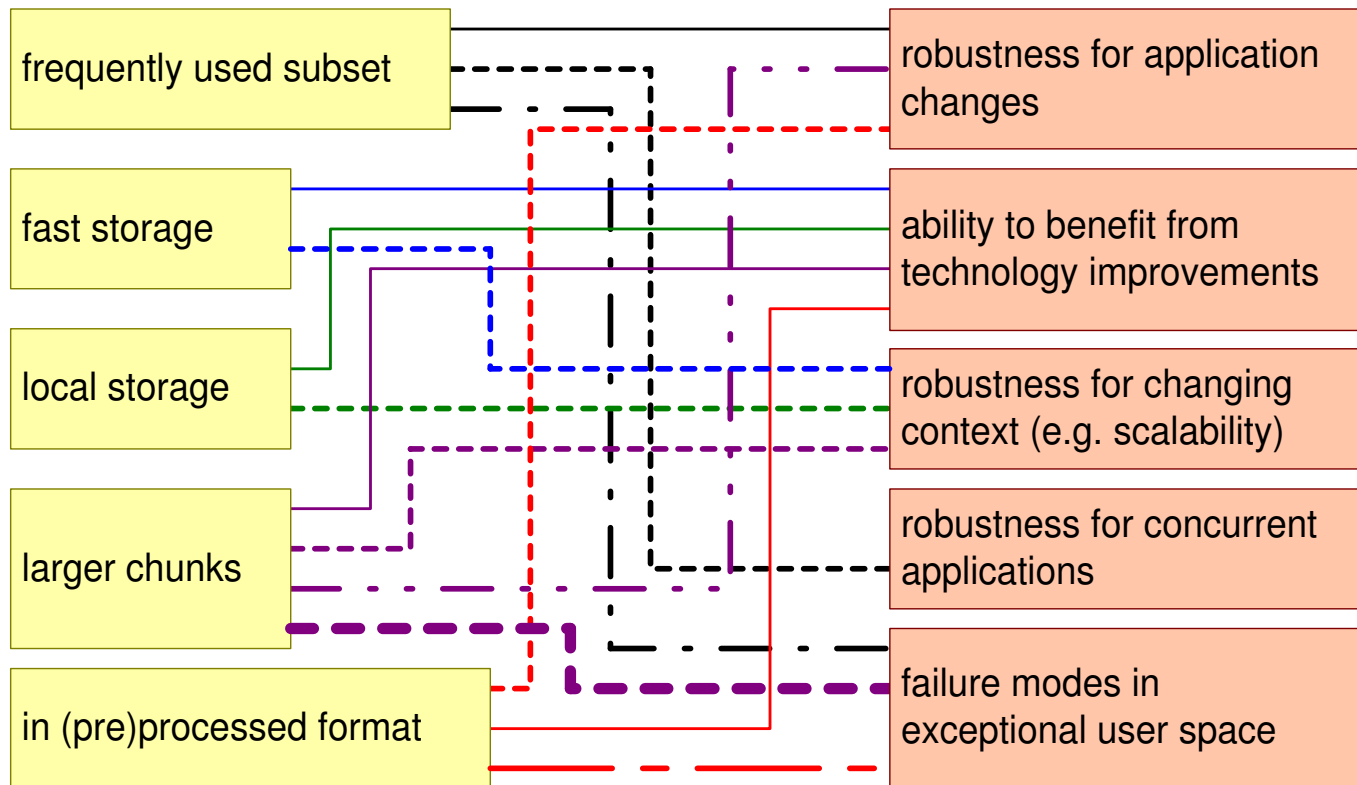
|                          | cache miss penalty | cache hit performance |
|--------------------------|--------------------|-----------------------|
| application cache        | 1 s                | 10 ms                 |
| network layer cache      | 100 ms             | 1 ms                  |
| file cache               | 10 ms              | 10 us                 |
| virtual memory           | 1 ms               | 100 ns                |
| memory caches L1, L2, L3 | 100 ns             | 1 ns                  |



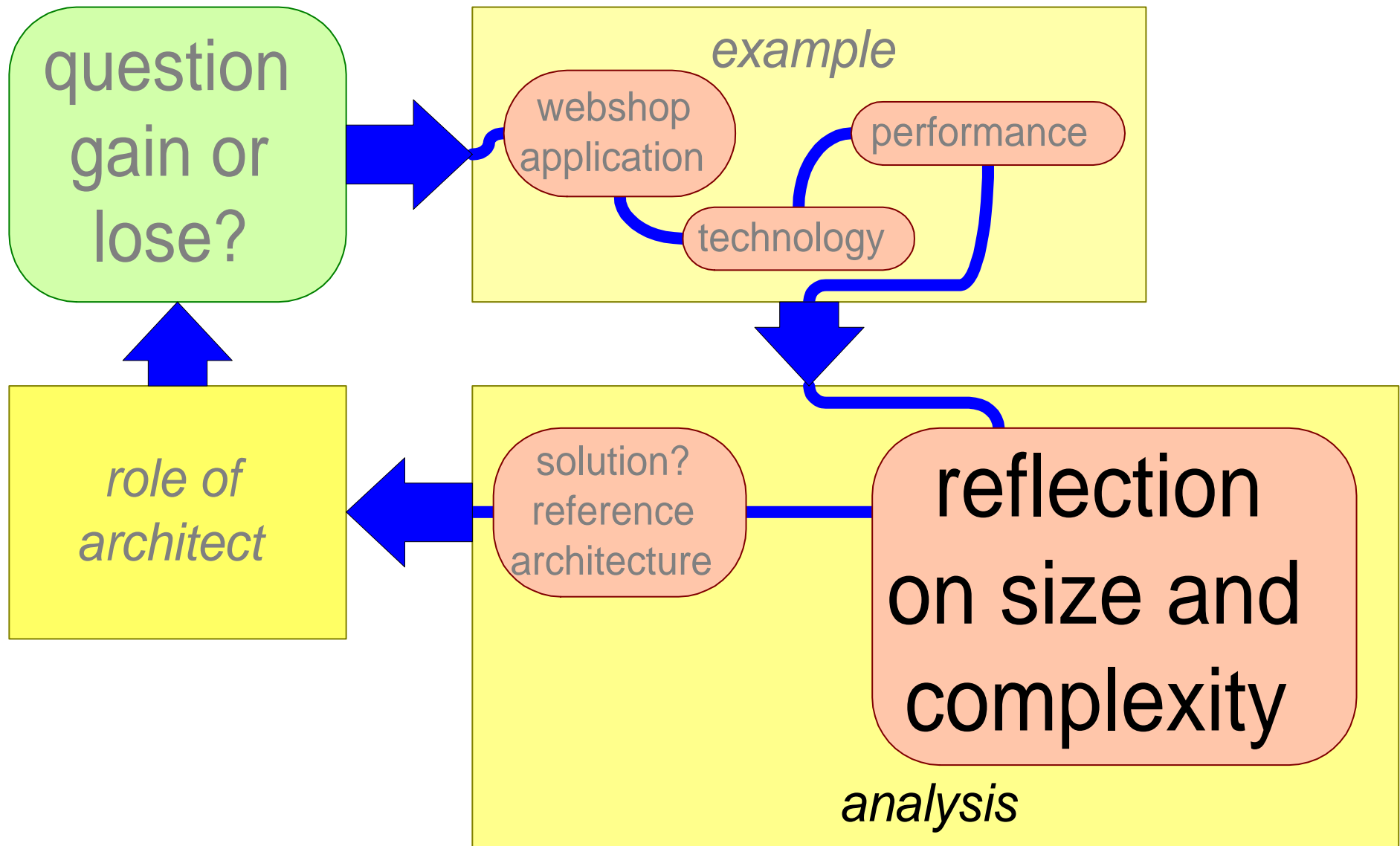
# Why Caching?



# Risks of Caching

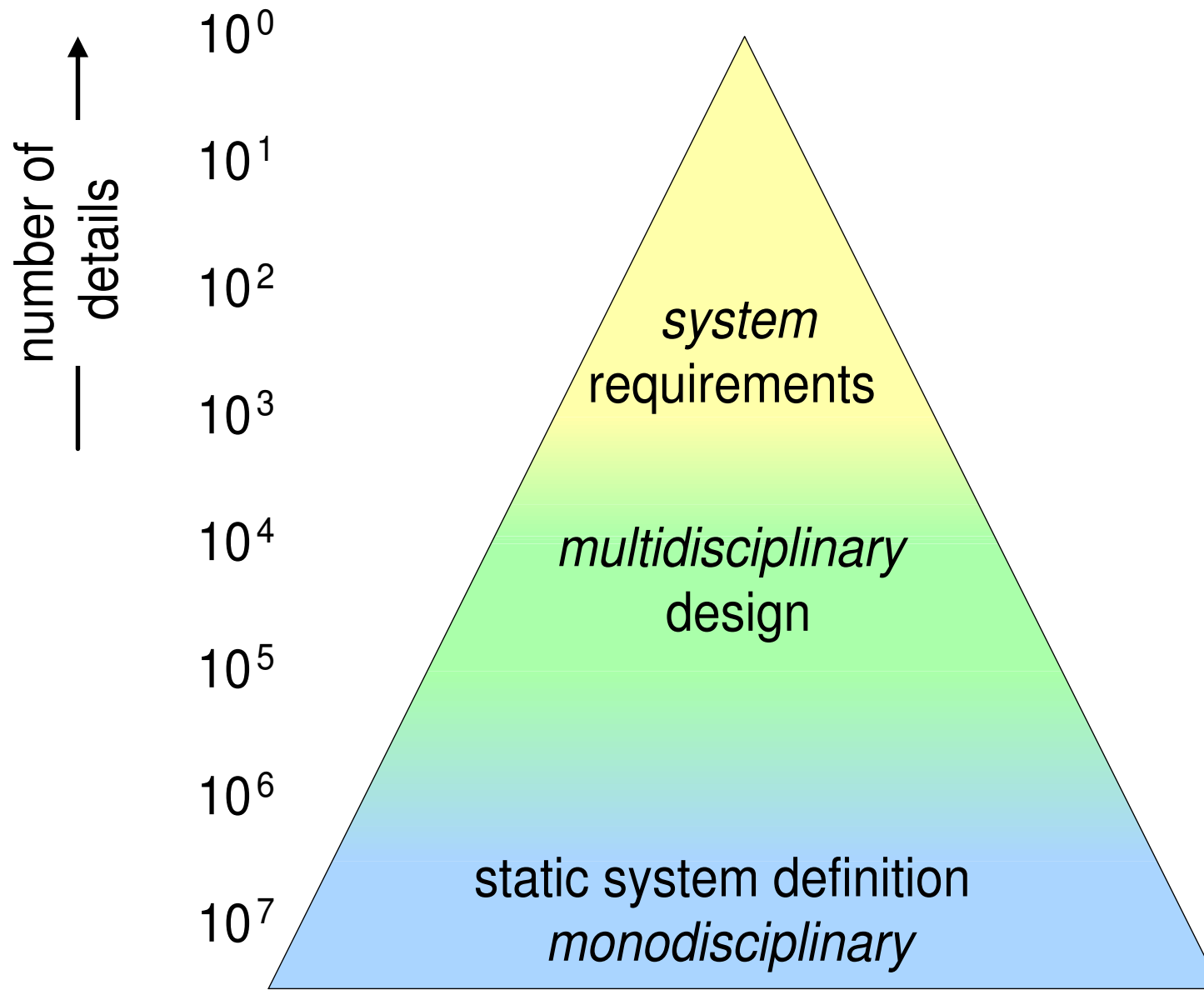


# Size and Complexity

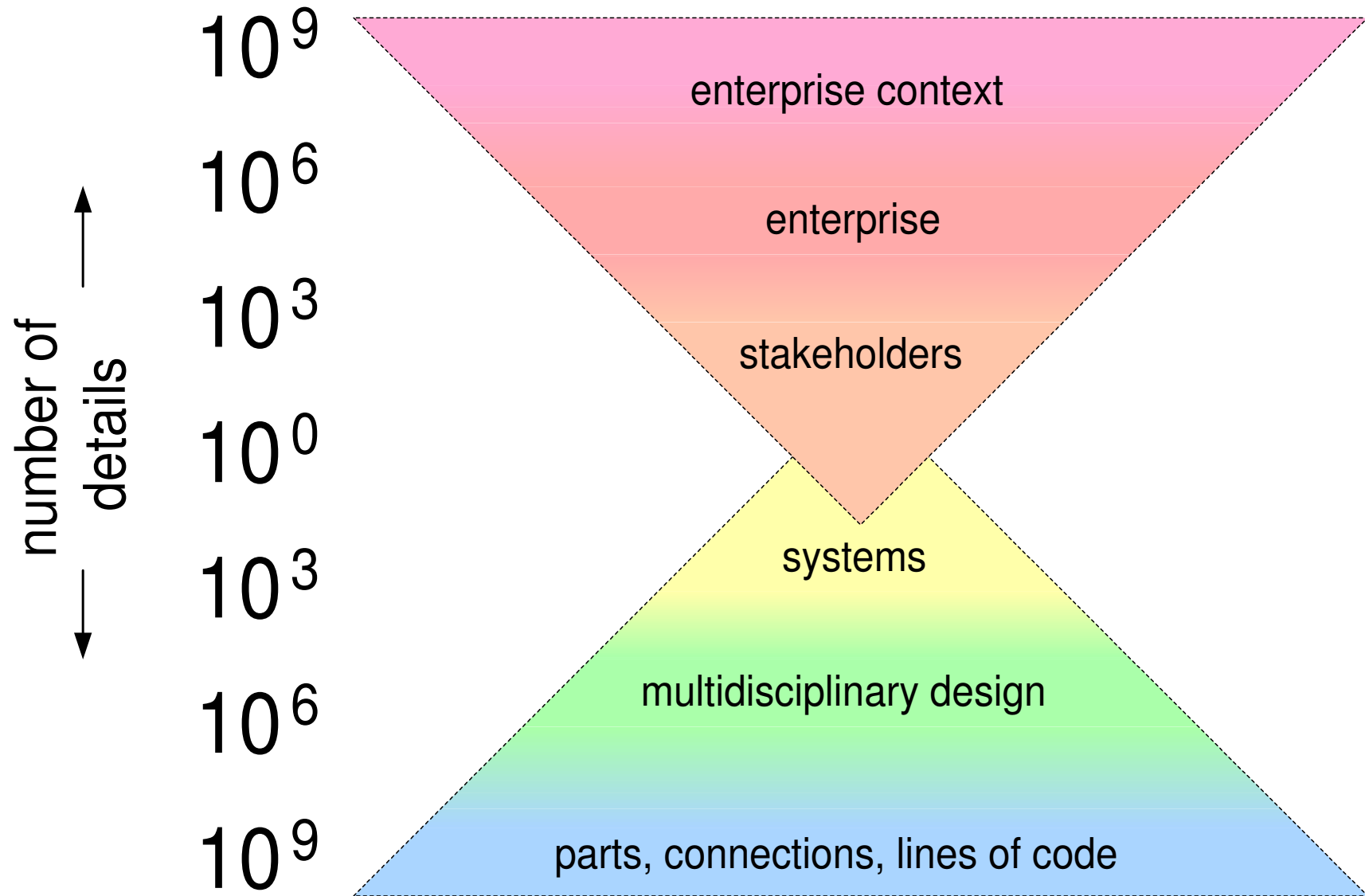


# Level of Abstraction Single System

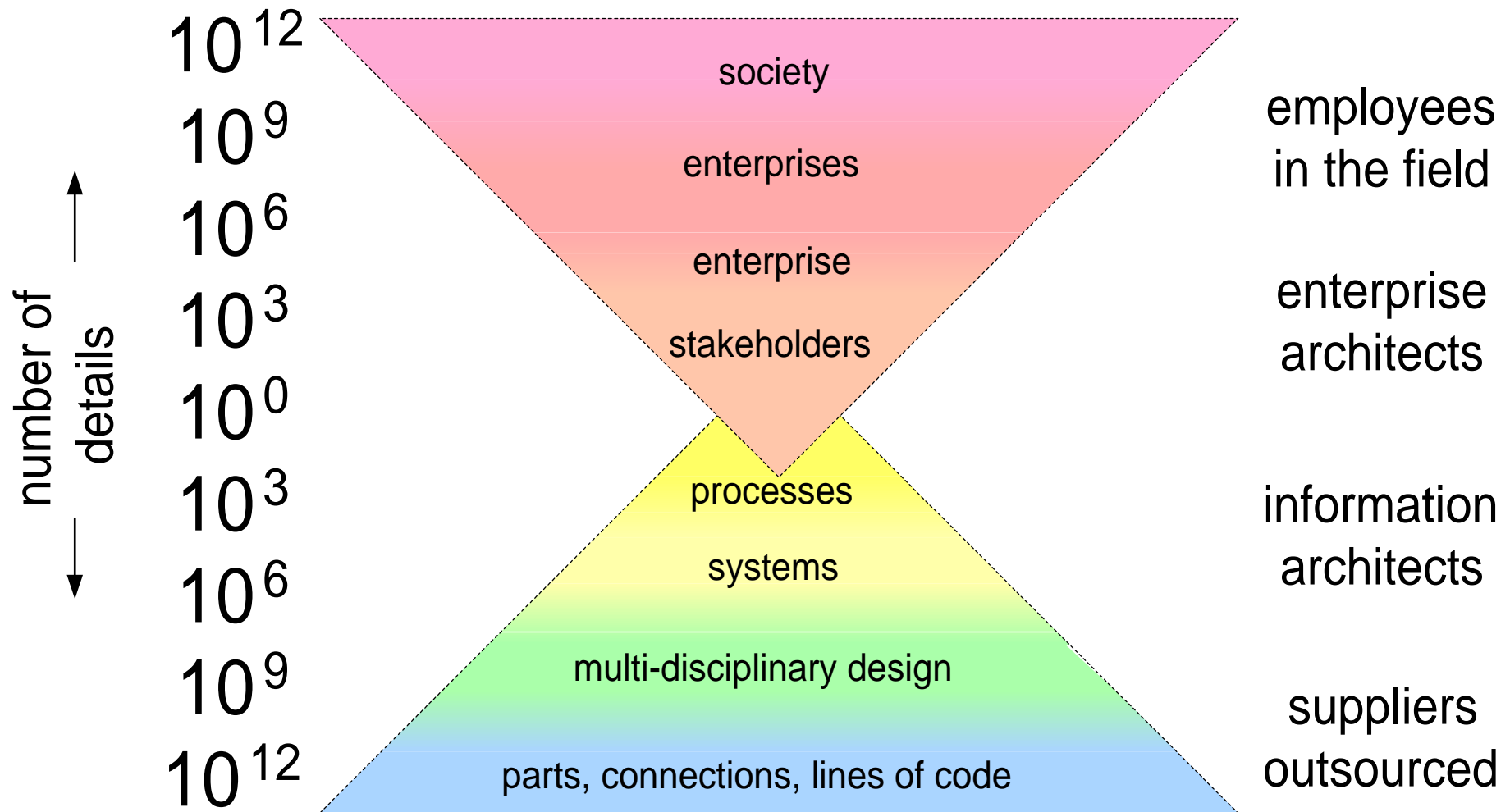
---



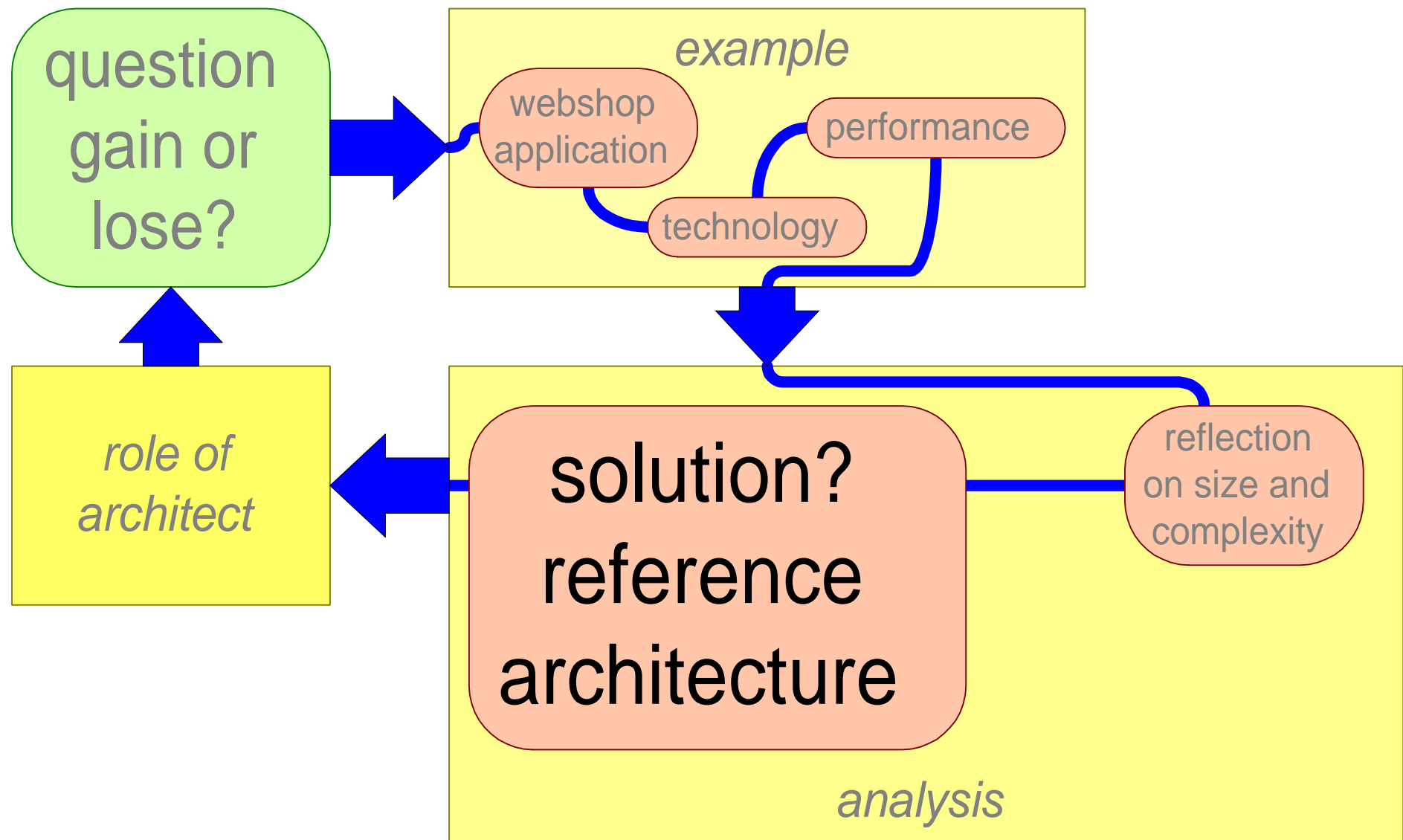
# Product Family in Context



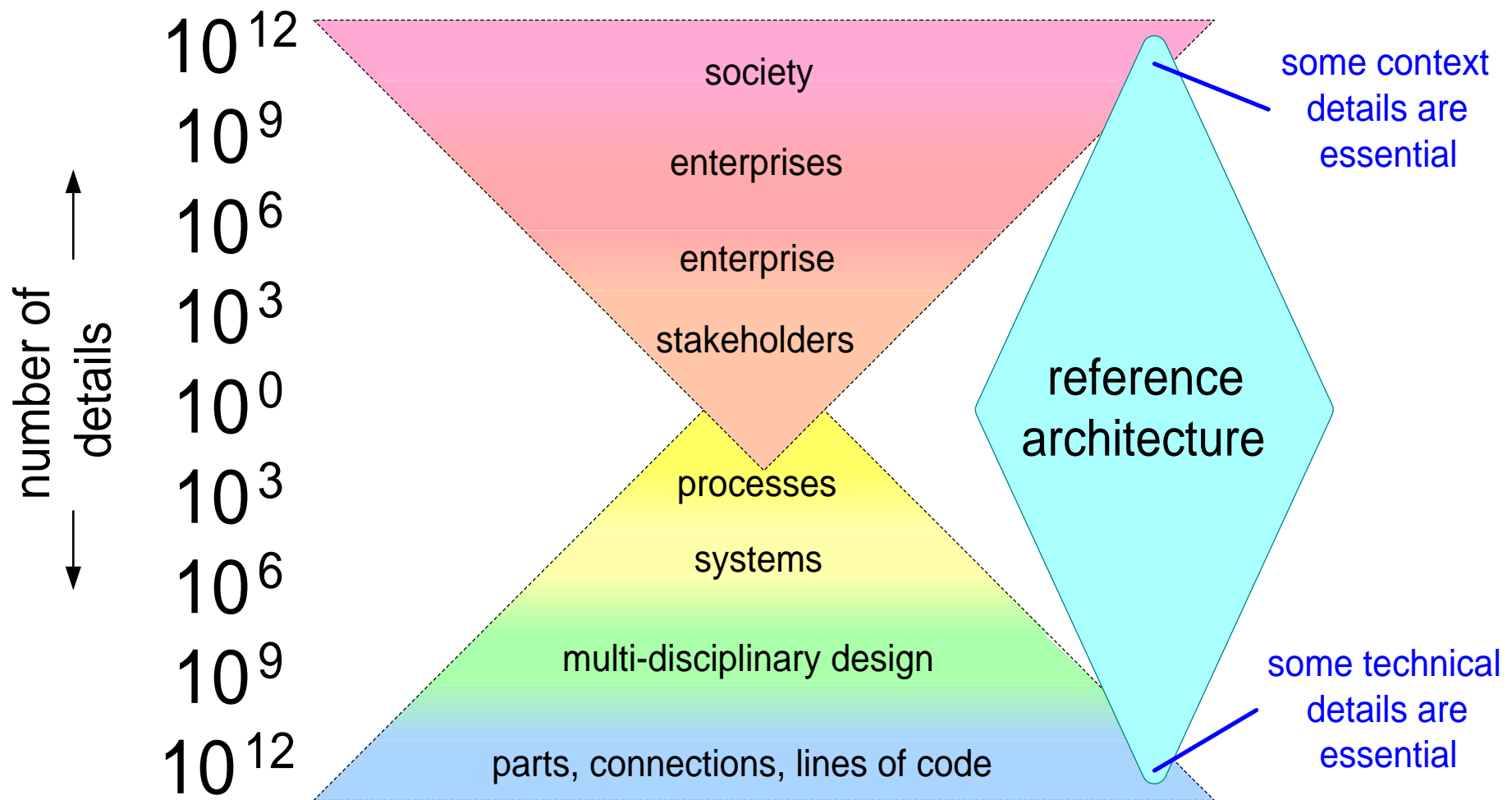
# Number of Details in Today's Services



# Reference Architecture



# Reference Architecture as Solution?



1.1 One of several prerequisites for architecture creative synthesis is the definition of **5-7 specific key drivers** that are critical for success, along with the rationale behind the selection of these items

2.1. The essence of a system can be captured in about **10 models/views**

2.2. A **diversity** of architecture descriptions and models is needed: languages, schemata and the degree of formalism.

2.3. The level of **formality** increases as we move closer to the implementation level.

from <http://www.architectingforum.org/bestpractices.shtml>

# Possible useful visualizations



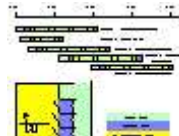
COVmotorwayManagementKeyDrivers



LWValueChain



COVsuppliers



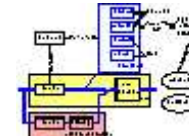
AVdynamicsURF



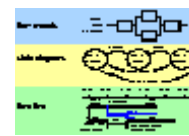
AVstakeholders



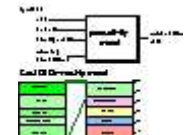
AVcontextMotorwayManagement



AVsimpleTVmodel



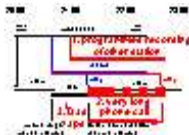
AVdynamicModels



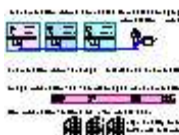
AVcostBenefitModels



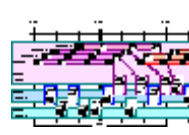
SHTexampleStoryLayout



ETexampleTimeShiftingWhatIf



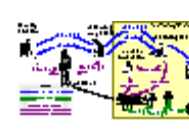
MICAftypicalCase



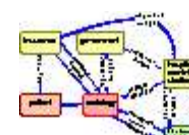
MICAftypicalTiming



MICAclinicalInfoFlow



MICArequestFlow



MICAfinancialContext



MICAfSystemLayers



MICAfReferenceModel



MICAfmarketSegmentation



MICAfInformationLayers



FVcommercialTree



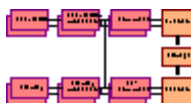
FVfeatureMatrix



FVinformationModel



FVdatamodel



CVfunctionalDecomposition



CVconstructionDecomposition



CVinformationModel



CVprocessDecomposition



CVreconstructionPerformanceModel



CVstartUp



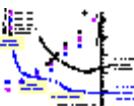
CVworkBreakdown



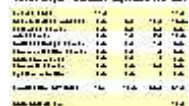
MAFTexampleWebShop



CVintegrationPlan



RVperformanceCostEffort



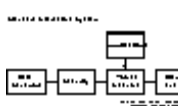
RVmemoryBudgetTable



ASMLoverlayBudget



MICVpresentationPipeline



FFTSstandardInteractiveSystemAnnotated



EBMImemoryTimingARM

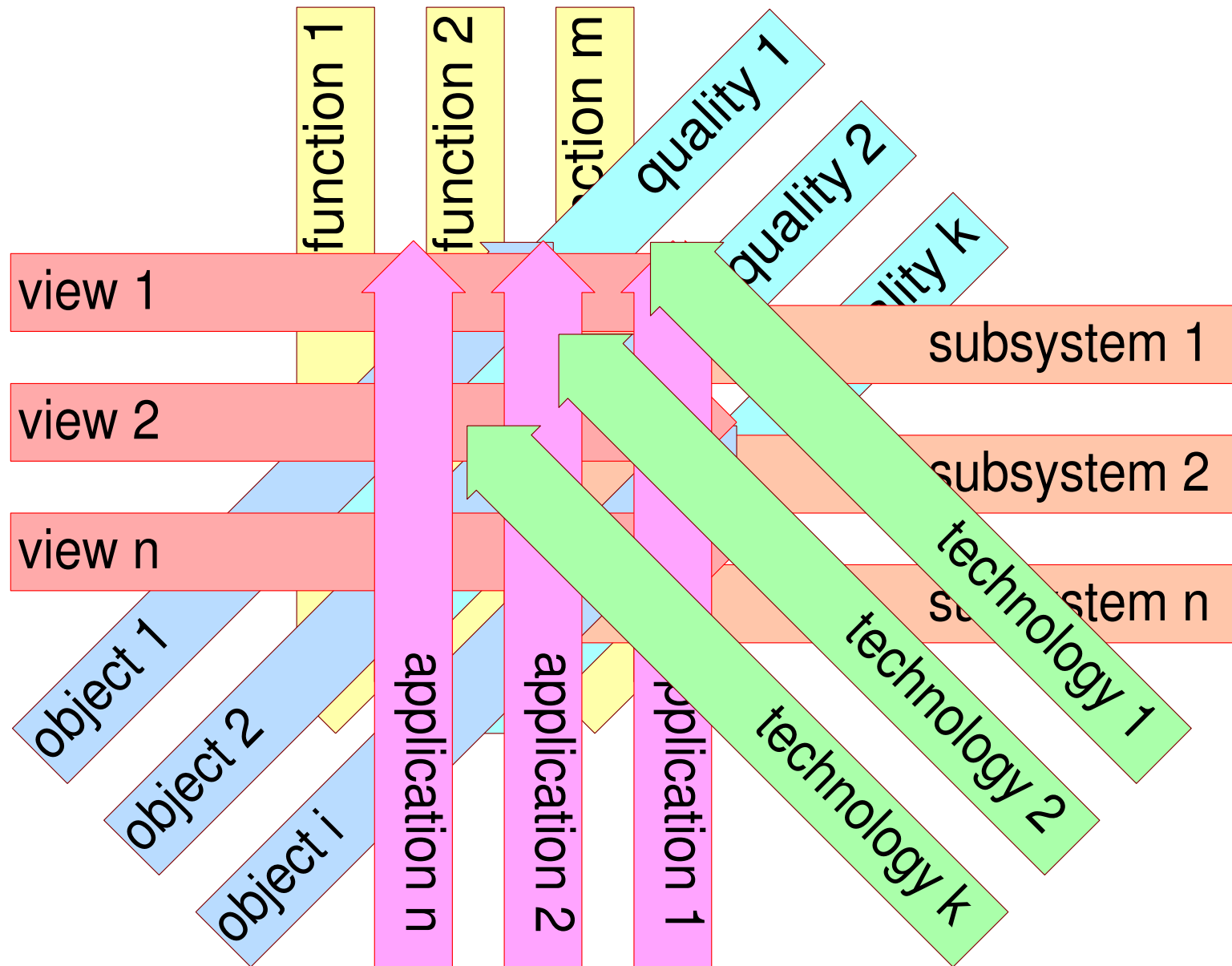


MAFTstoragePerformance

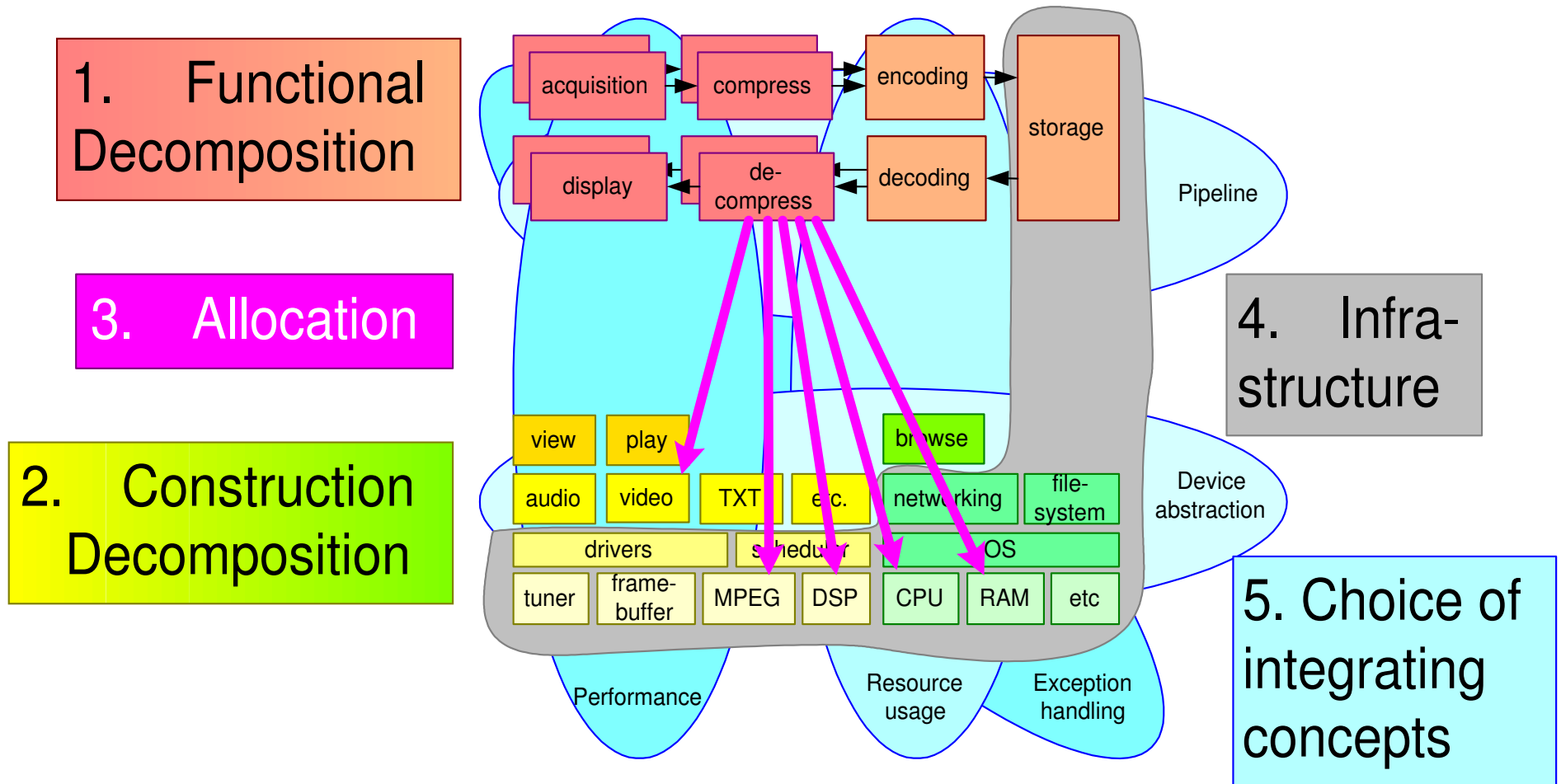
actual figures and references to their use at

<http://www.gaudisite.nl/figures/<name>.html>

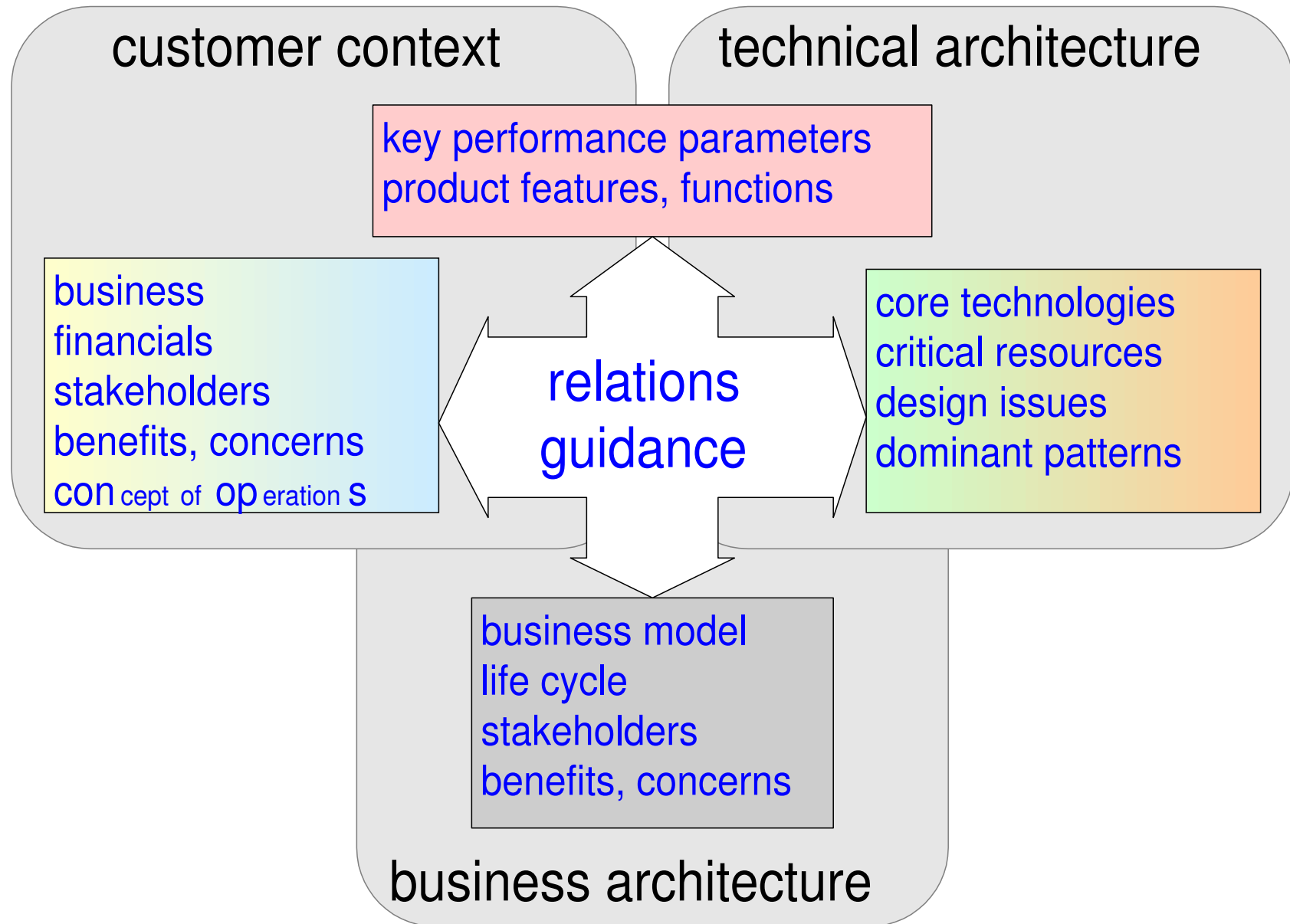
# Ideal Structure does not exist



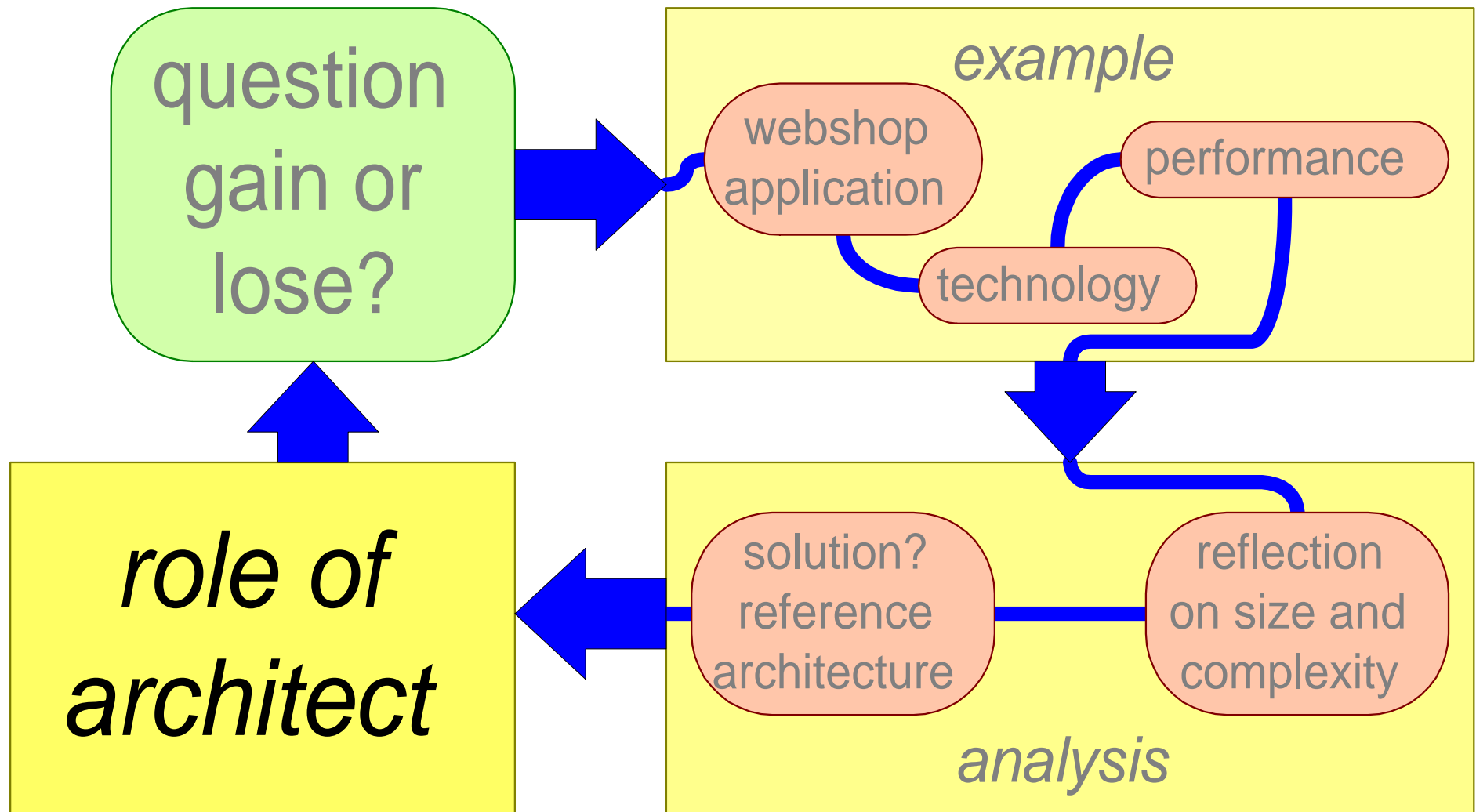
# Synthesis, Integration, Relation oriented



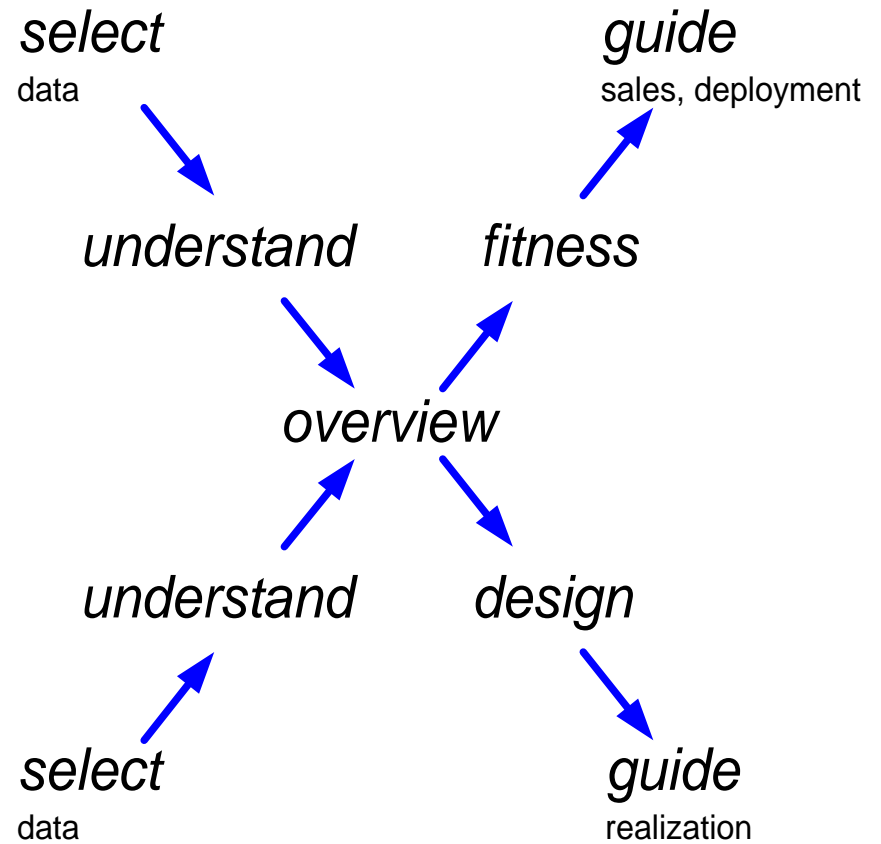
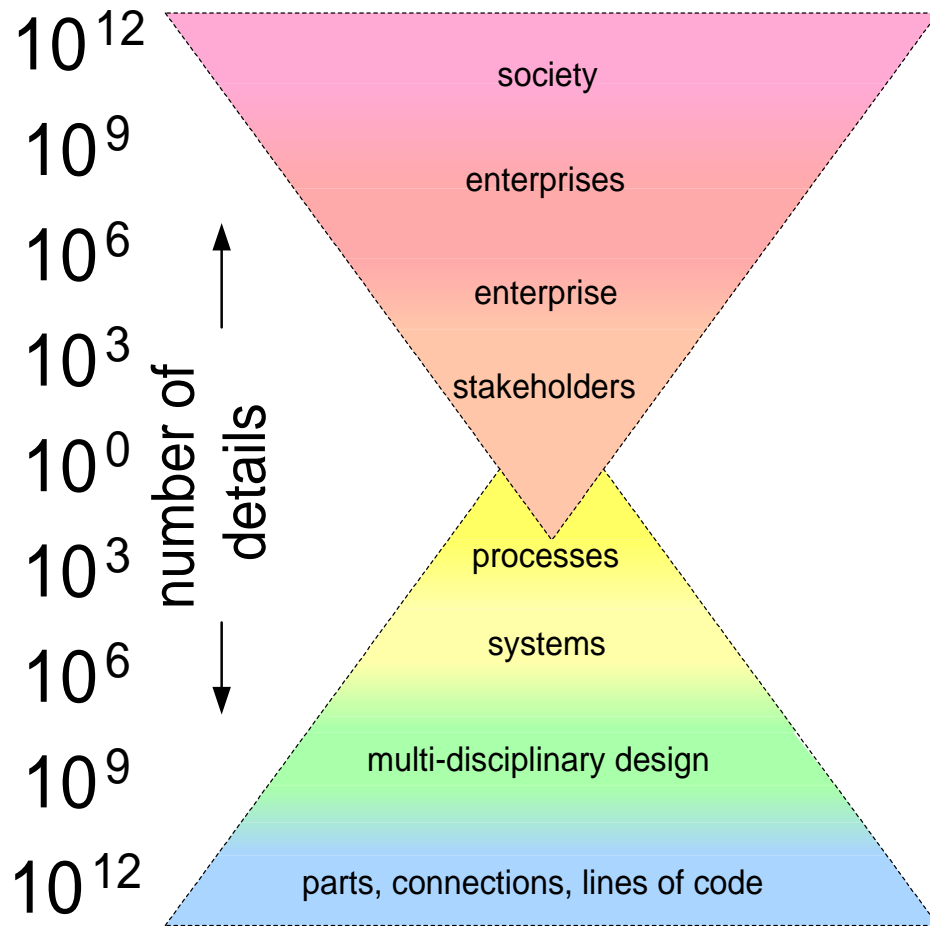
# Checklist for RA content



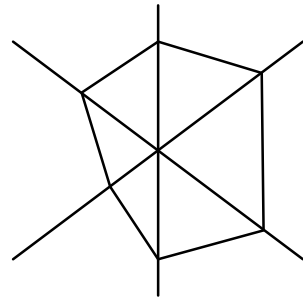
# Role of Architect



# Tasks of Architect

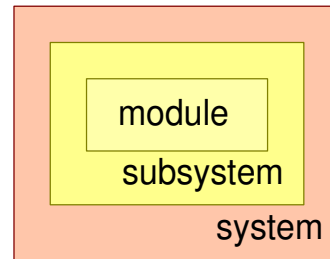


# Responsibilities



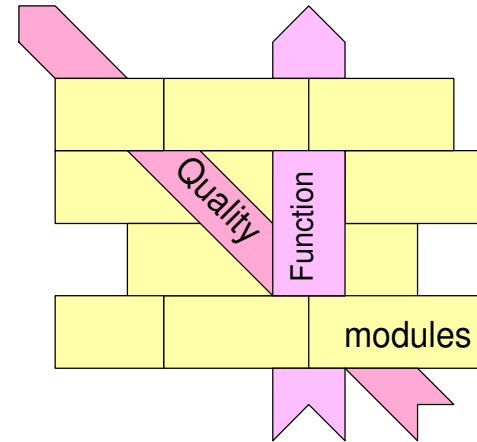
Balance

Requirement  
Spec  
Design  
Realization



Consistency

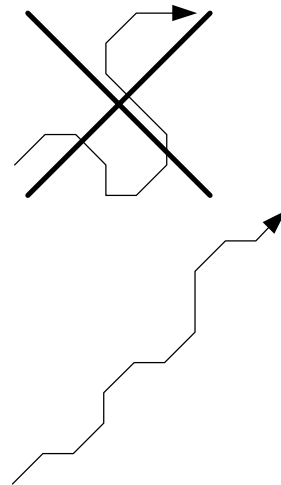
Decomposition  
Integration



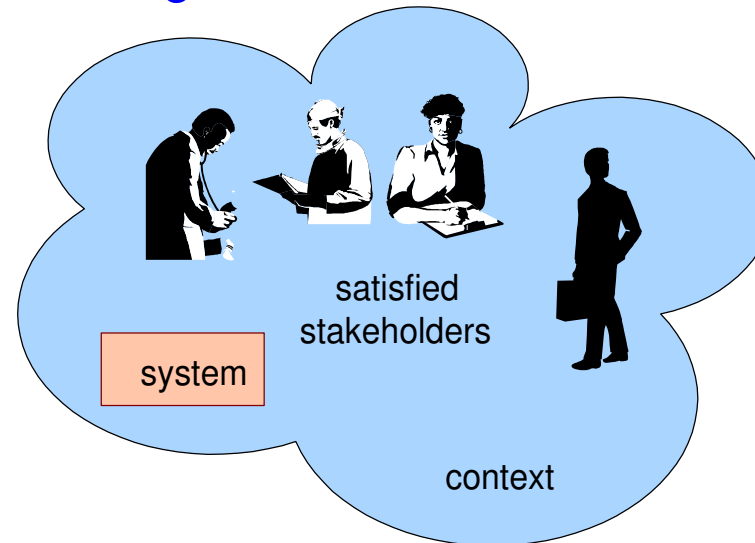
Overview

KISS

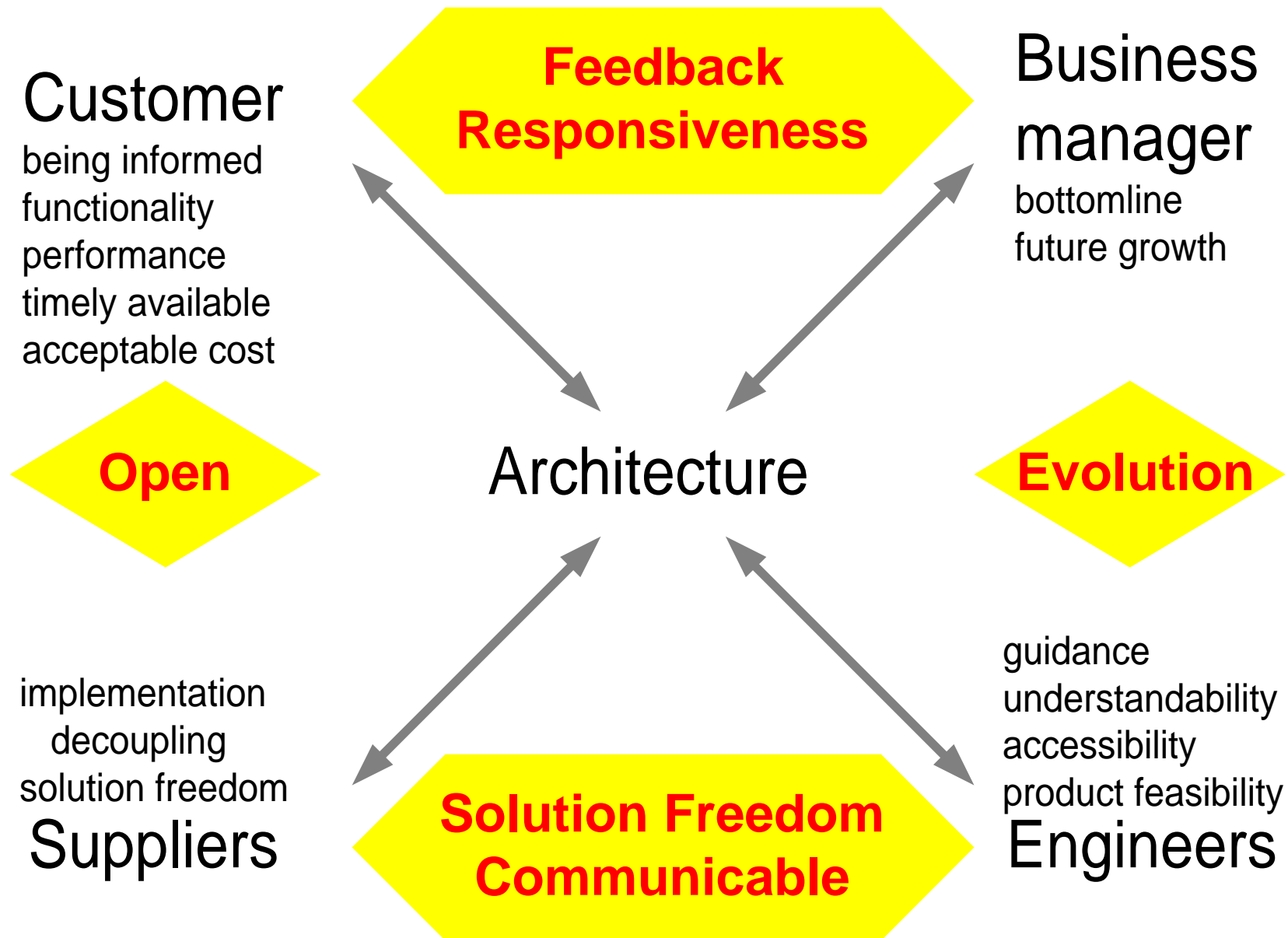
Elegance  
Simple



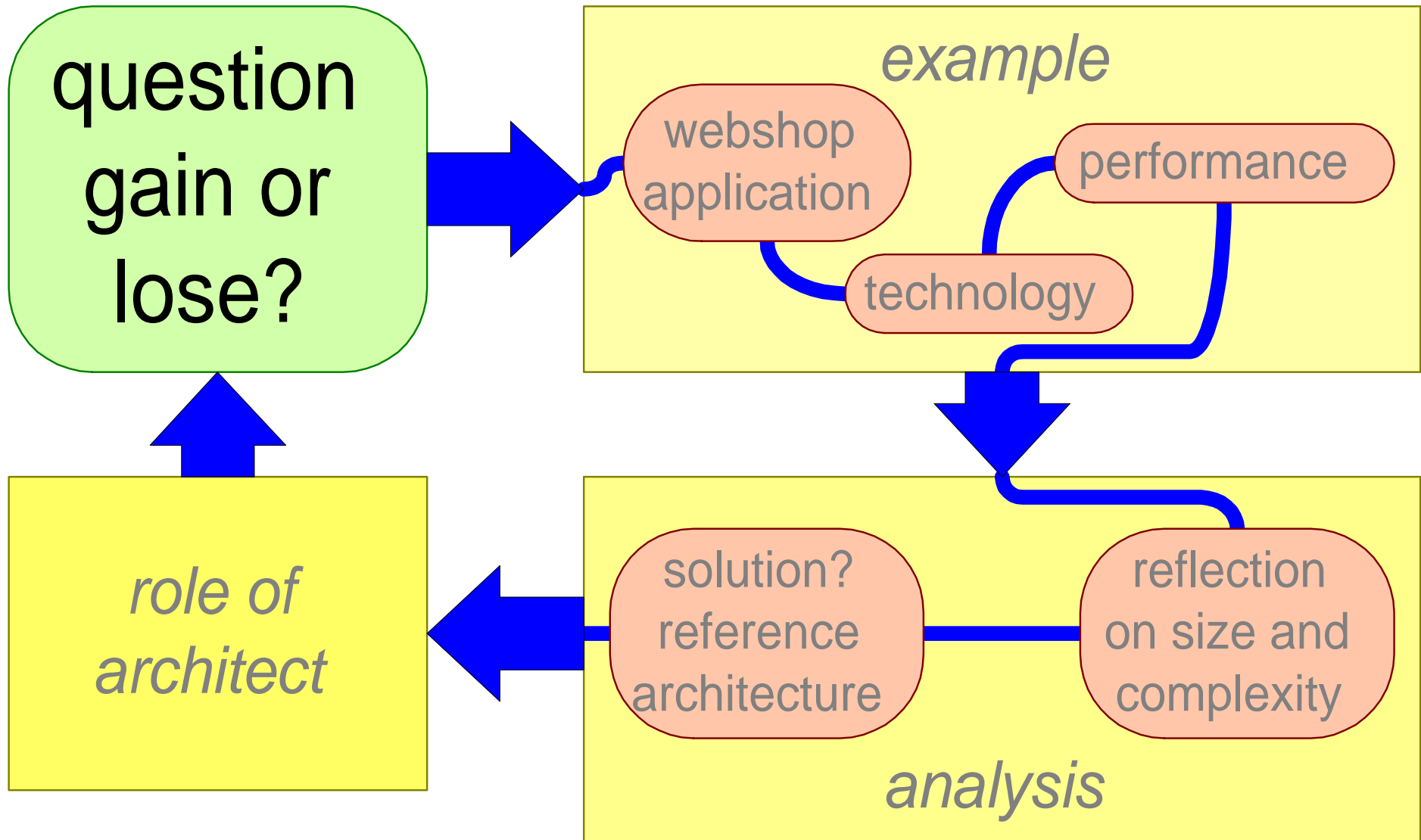
Integrity



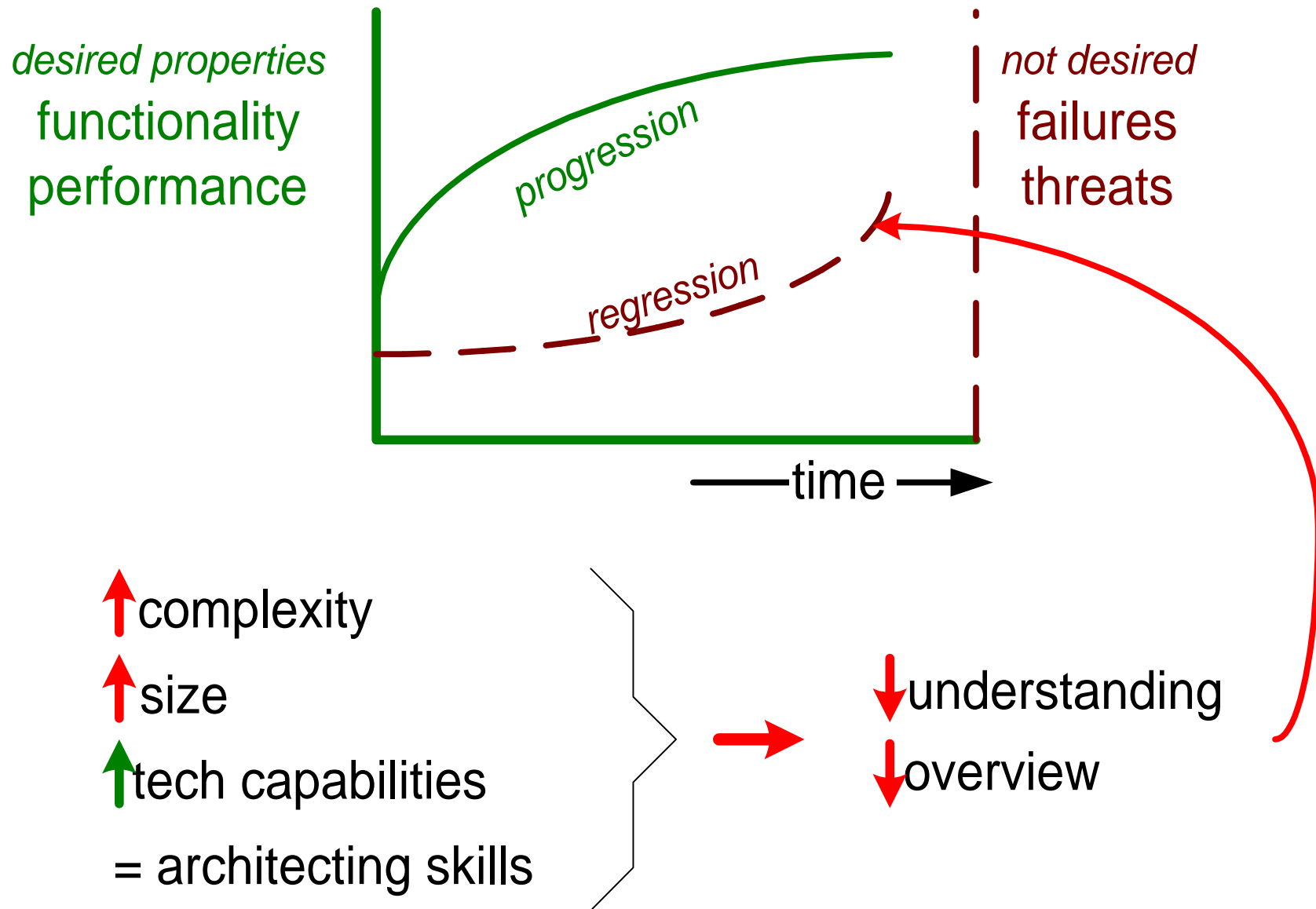
Fitting



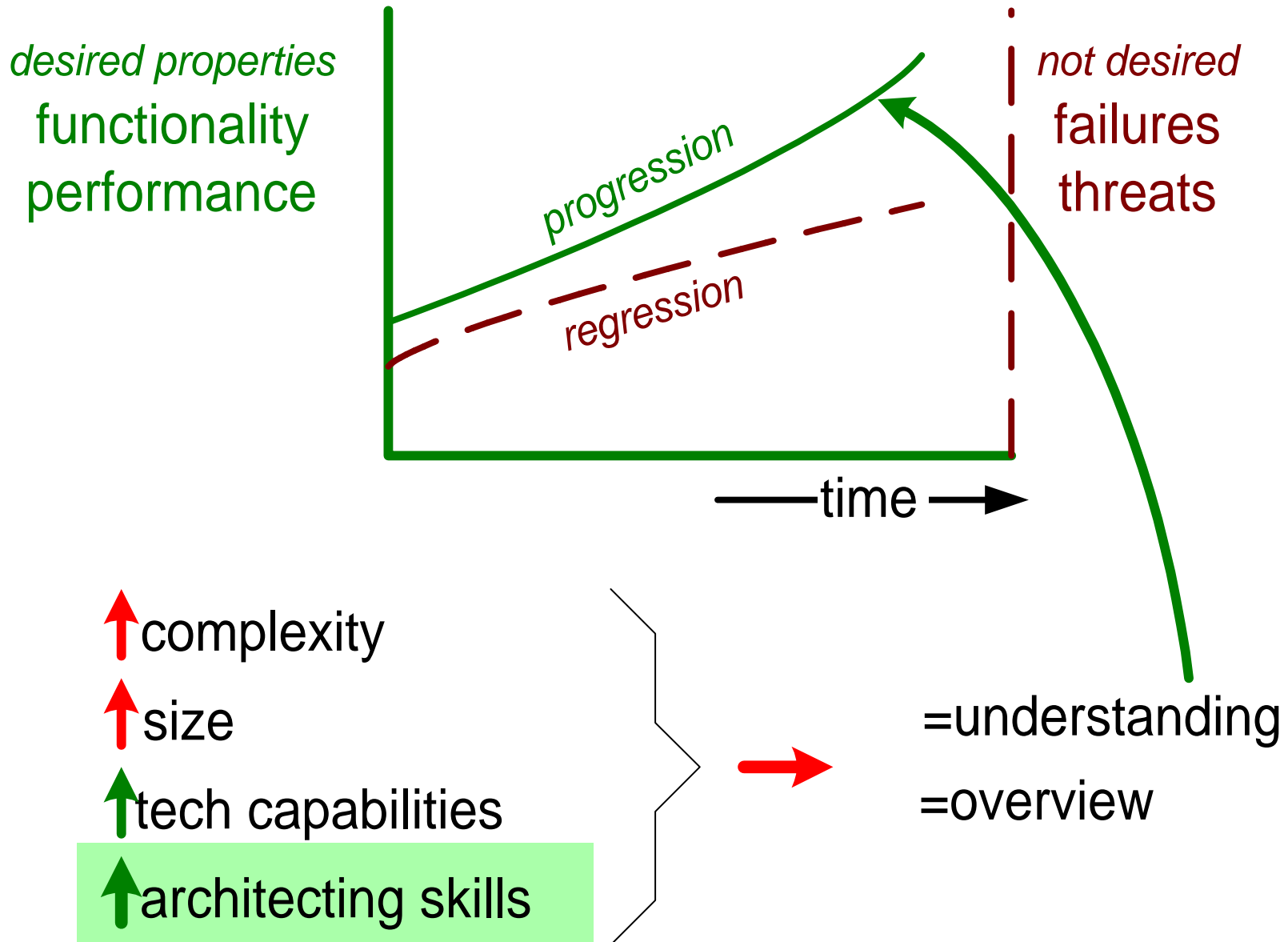
# Gain or Lose?



# Loss Scenario

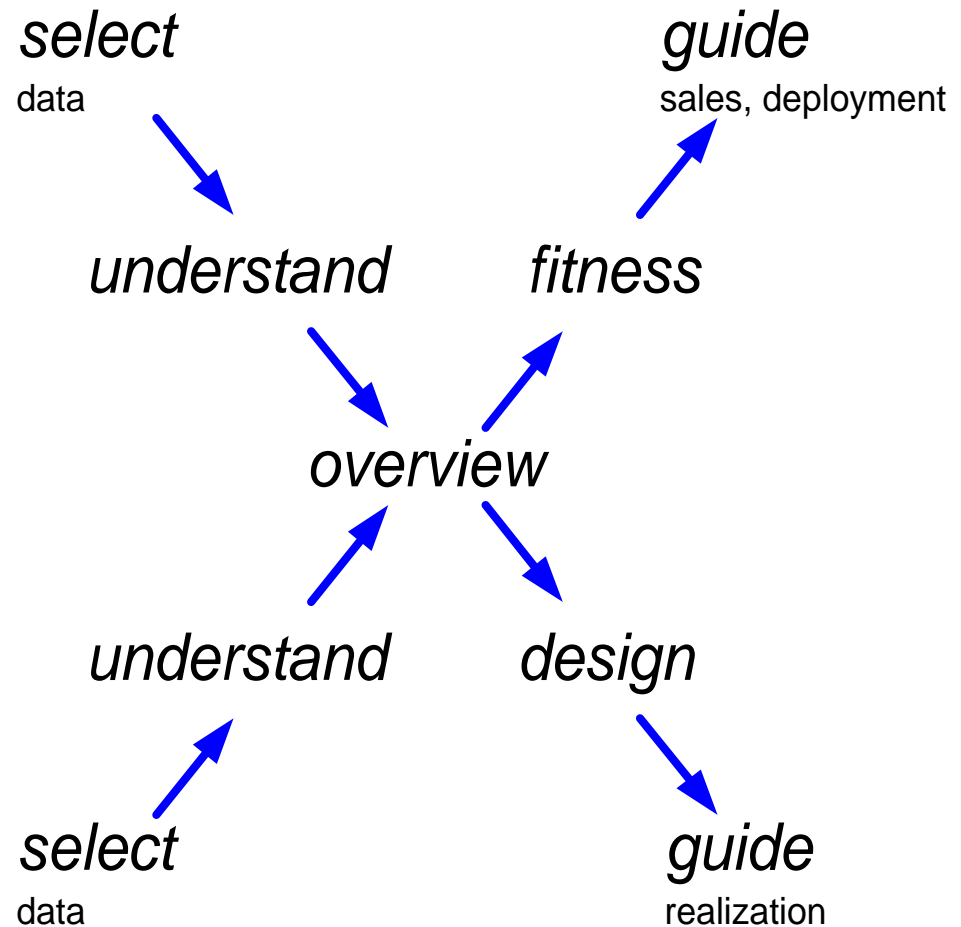
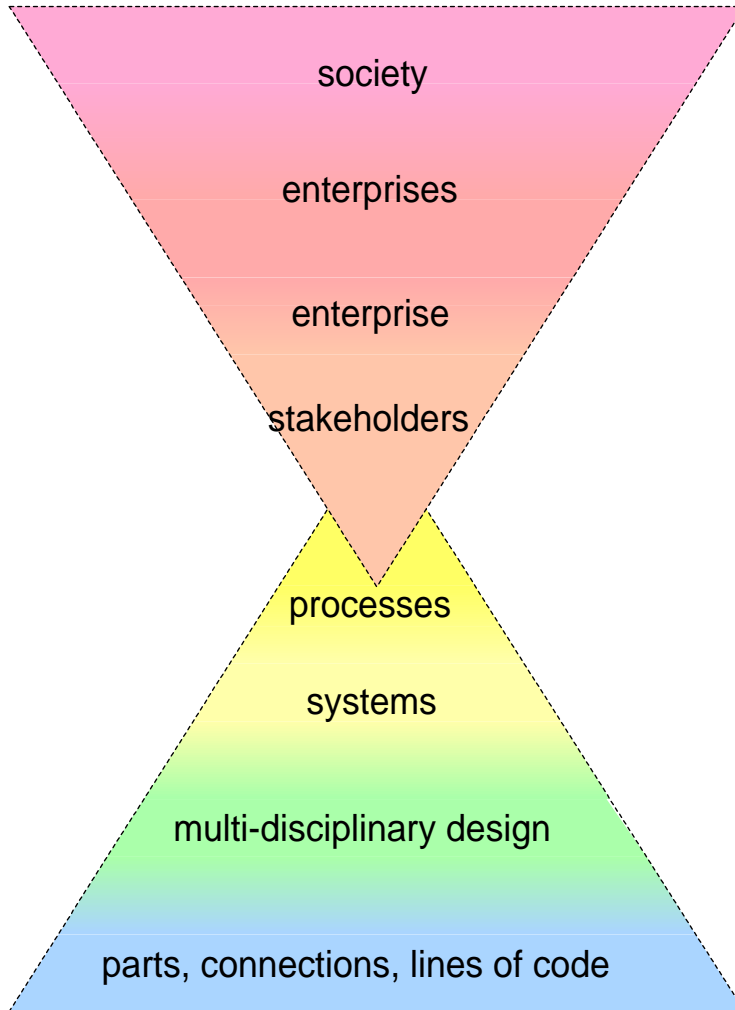


# Gain Scenario



# Conclusion

We need to improve architecting skills to gain.



# Read More at the Gaudisite

---

<http://www.gaudisite.nl/>

## Reference Architecture Primer

<http://www.gaudisite.nl/ReferenceArchitecturePrimerPaper.pdf>

## Webshop case is part of System Modeling and Analysis

<http://www.gaudisite.nl/SystemModelingAndAnalysisBook.pdf>

## All about Architecting: System Architecting

<http://www.gaudisite.nl/SystemArchitectureBook.pdf>