Lecture slides course Modeling and Analysis

by *Gerrit Muller*
Embedded Systems Institute

Abstract

The Modeling and Analysis course is part of a broader Systems Architecting Curriculum. It addresses the skills and methods needed to model and analyse systems in their context. The purpose of modeling is to gain insight in the domain and potential solutions to facilitate the architecting process.

July 3, 2023
status: planned
version: 0.6

Module Modeling and Analysis course info

by *Gerrit Muller* HSN-NISE
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract
This module provides the information about the “Modeling and Analysis” course.

July 3, 2023
status: planned
version: 0.2

Modeling and Analysis Overview Content

goal of this module

Provide overview and context for complete course.

Understand and experience the connection between problem and solution.

content of this module

Positioning of Modeling and Analysis (M&A)

Why, what and how of M&A

Program of the complete course

Overview of M&A approach

exercise

Quick scan of one case

Electronic Patient Record, Video on Demand, or Health Care Archive

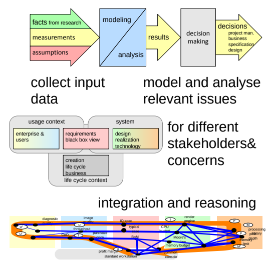
Modeling and Analysis Overview

by *Gerrit Muller* HSN-NISE
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

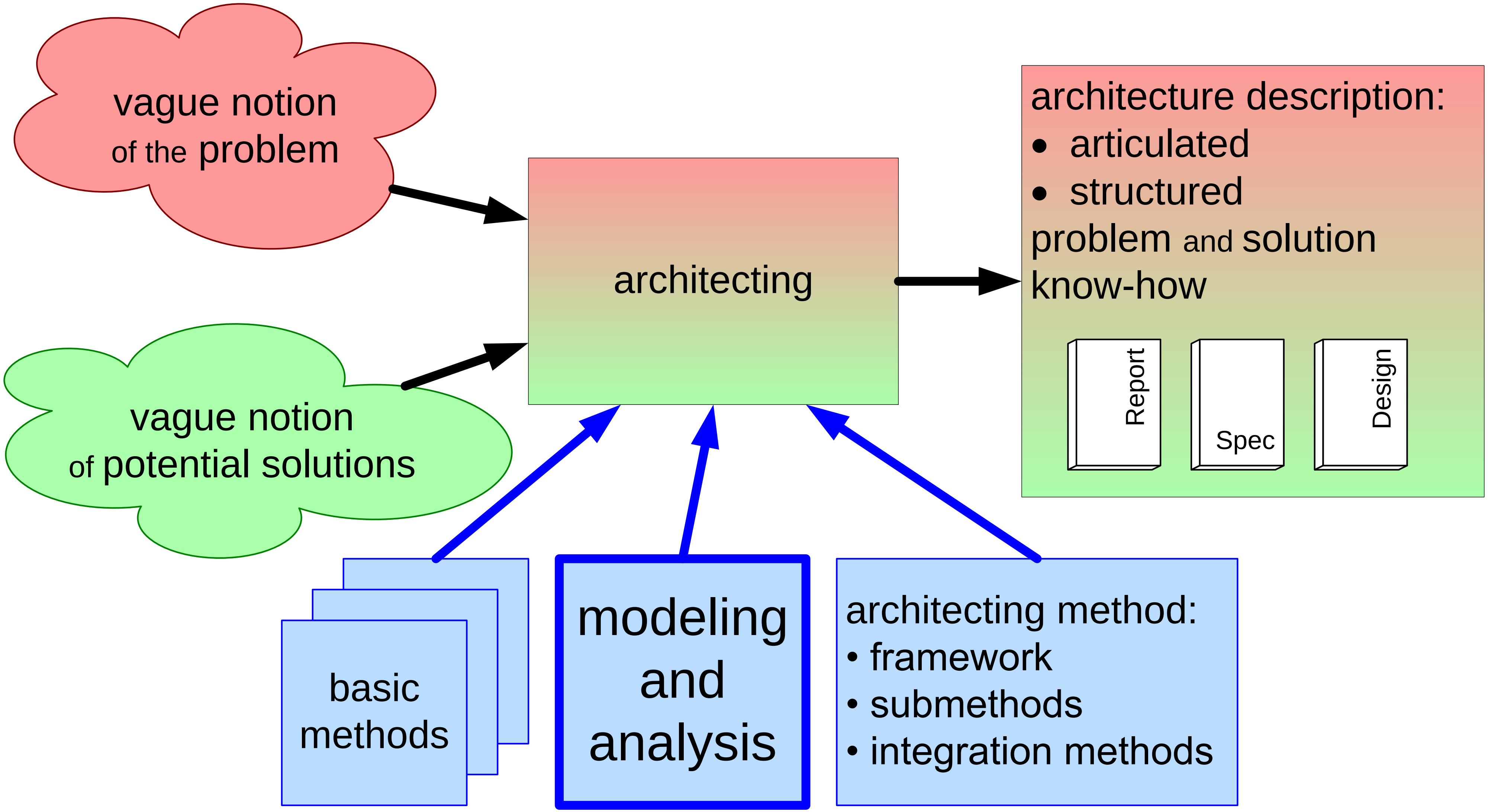
Abstract

The course Modeling and Analysis is described. The program consists of 10 modules. The course format, iterating theory, illustration and interaction is explained. The course heavily emphasizes the practical application of the method. This presentation shows the overview of the modeling and analysis approach and the methods and techniques that will be elaborated in the rest of the course.

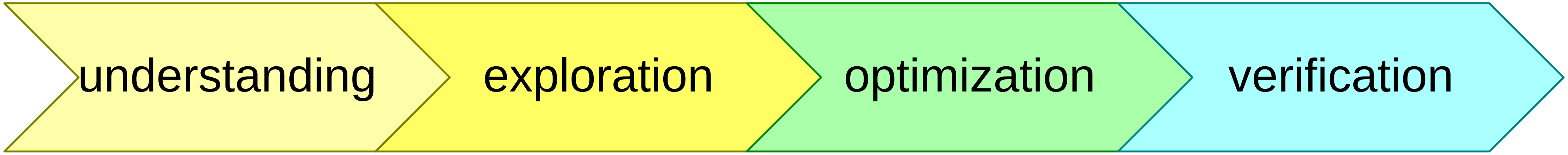
July 3, 2023
status: preliminary draft
version: 1.0



Positioning Modeling and Analysis in Architecting



Modeling and Analysis supports:



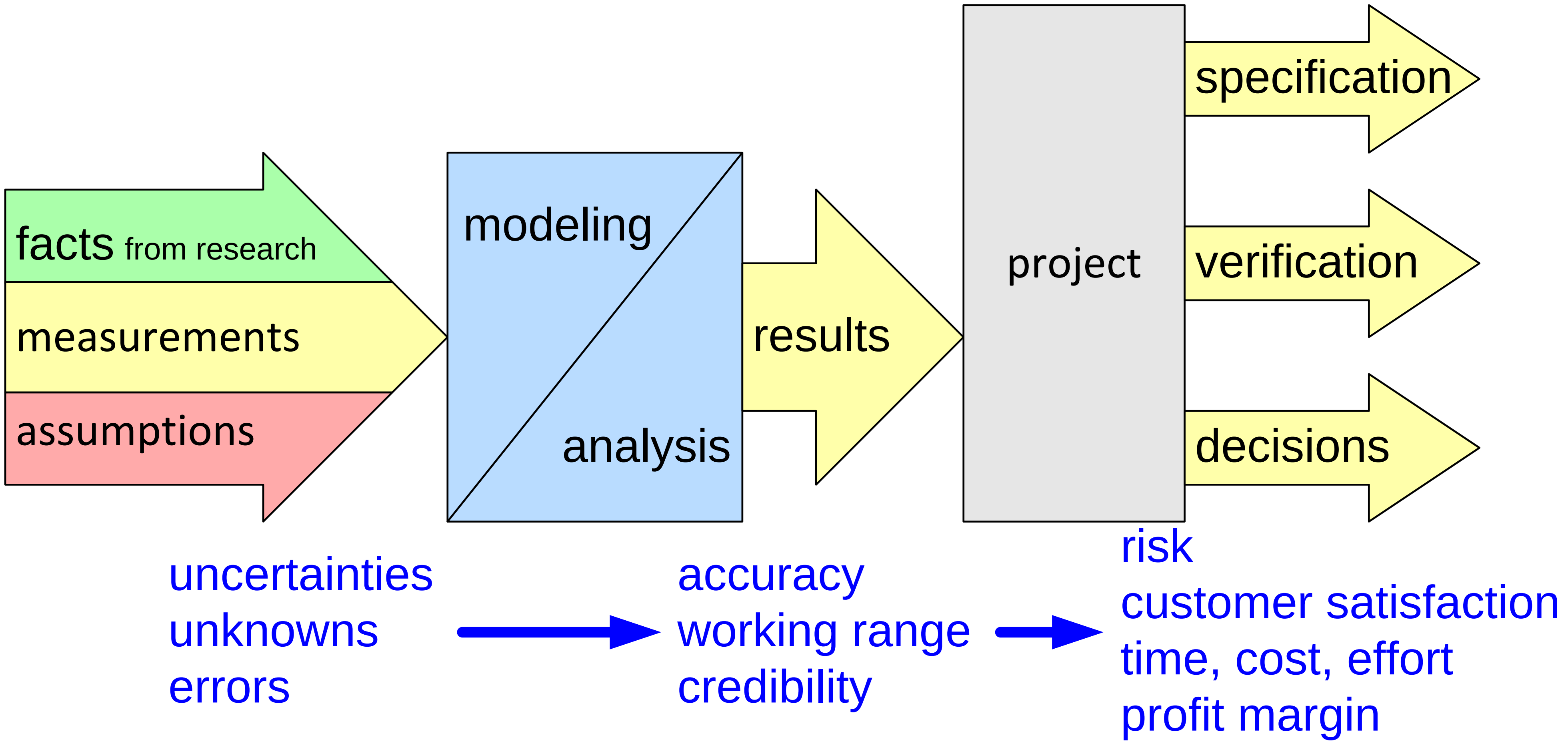
Type of model depends on project phase

Models have a goal

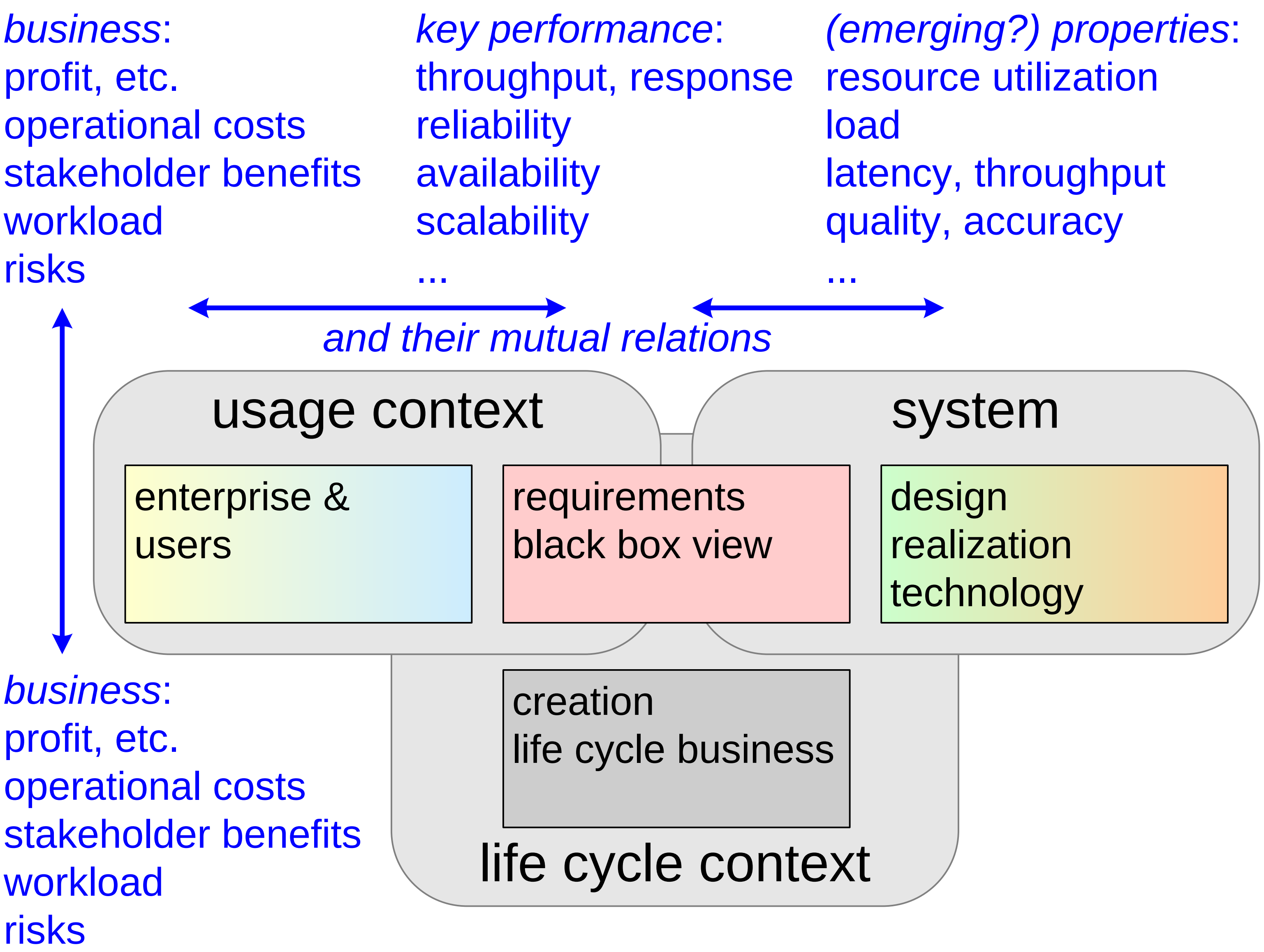
Goals evolve and models evolve

Techniques are used to reach this goal

Purpose of Modeling



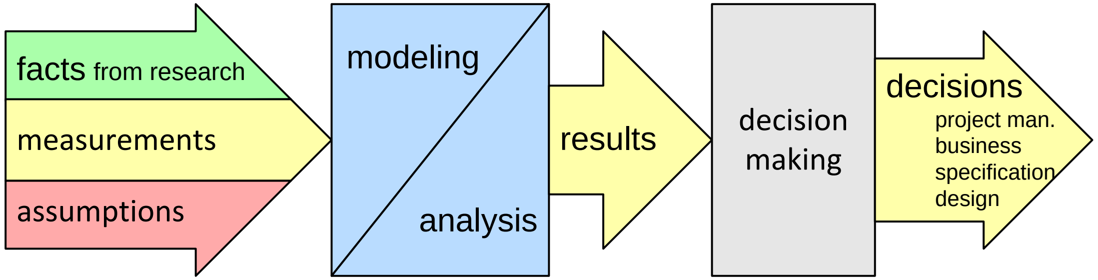
What to Model?



Program of Modeling and Analysis Course

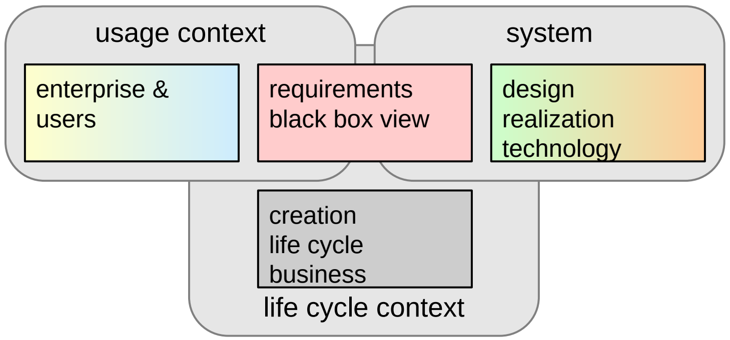
day 1	1. overall approach intro, overall approach, exercise overall approach
	2. input facts, data, uncertainties quantification, measurements, modeling, validation, technology background, lifecycle and business input sources
day 2	3. system modeling purpose, approaches, patterns, modularity, parametrization, means, exploration, visualization, micro-benchmarking, characterization, performance as example
	4. application, life-cycle modeling reiteration of modeling approach (see module 3), applied on customer application and business, and life cycle
day 3	5. integration and reasoning relating key driver models to design models, model based threads of reasoning, FMEA-like approach, modeling in project life-cycle
	6. analysis, using models sensitivity, robustness, worst case, working range, scalability, exceptions, changes

Overview of Approach



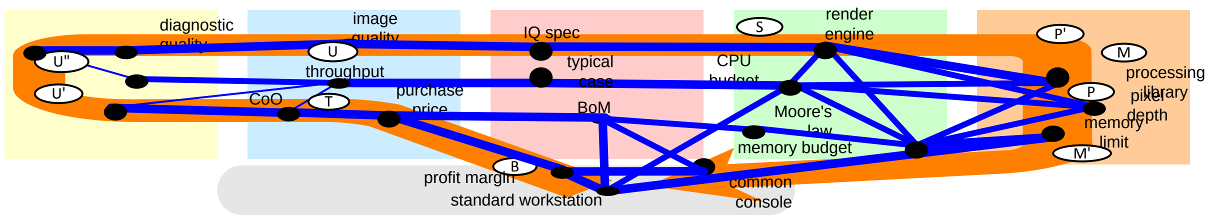
collect input data

model and analyse relevant issues

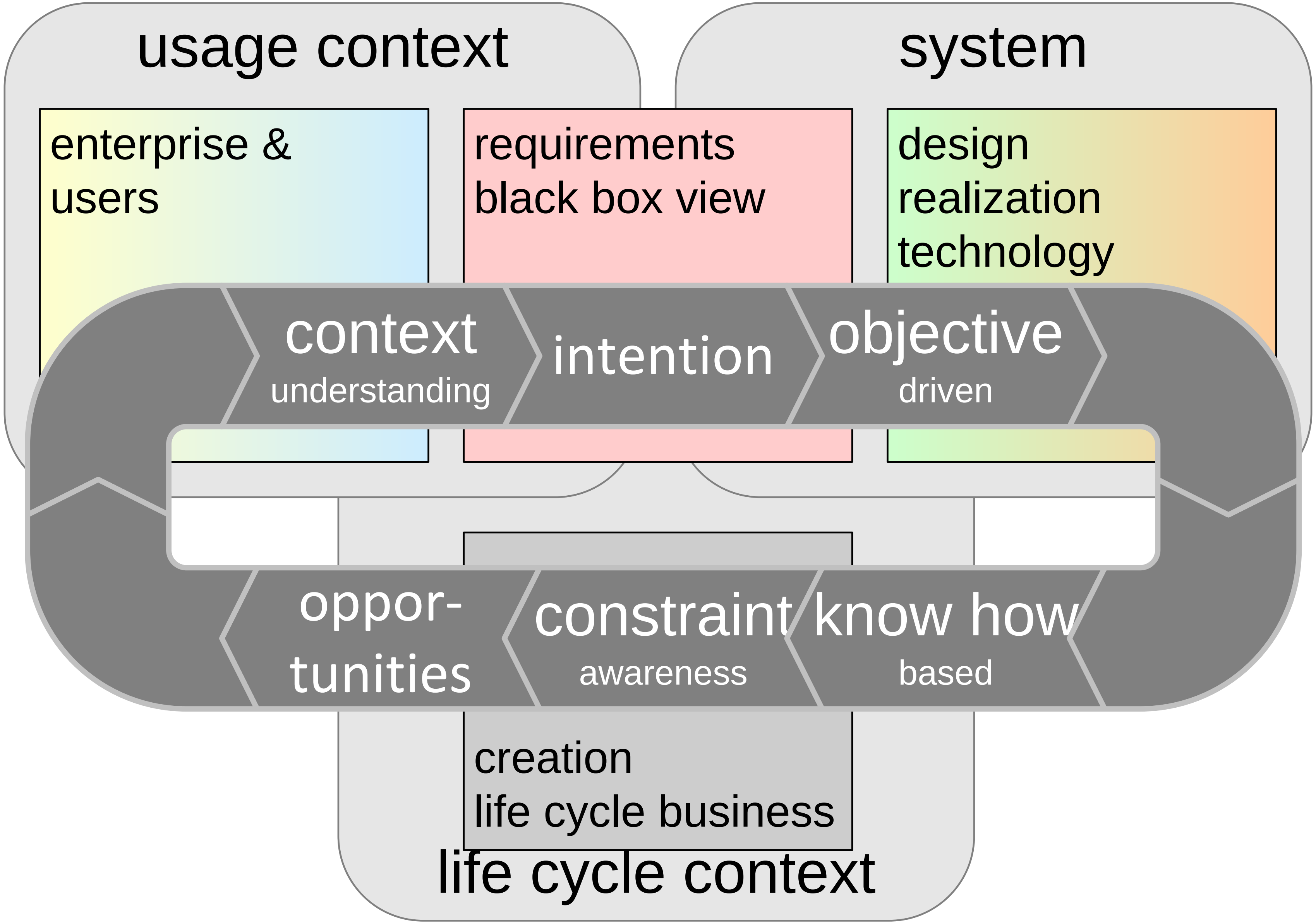


for different stakeholders& concerns

integration and reasoning



Iteration over viewpoints



Modeling and Analysis: Background of the Course

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

Abstract

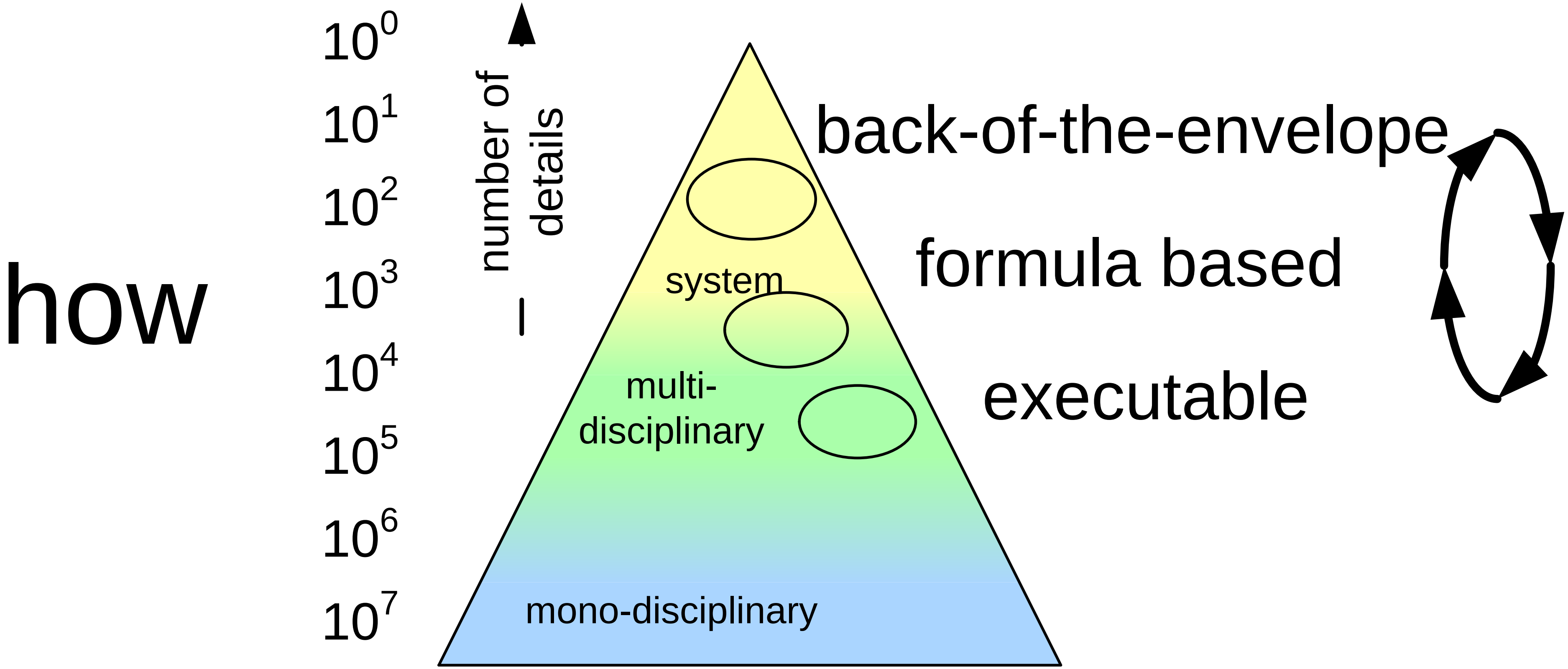
The background ideas of the Modeling and Analysis course are collected in a number of diagrams. These diagrams are provided solely as background and probably should not be shown during the course itself.

Distribution

July 3, 2023
status: planned
version: 0.1

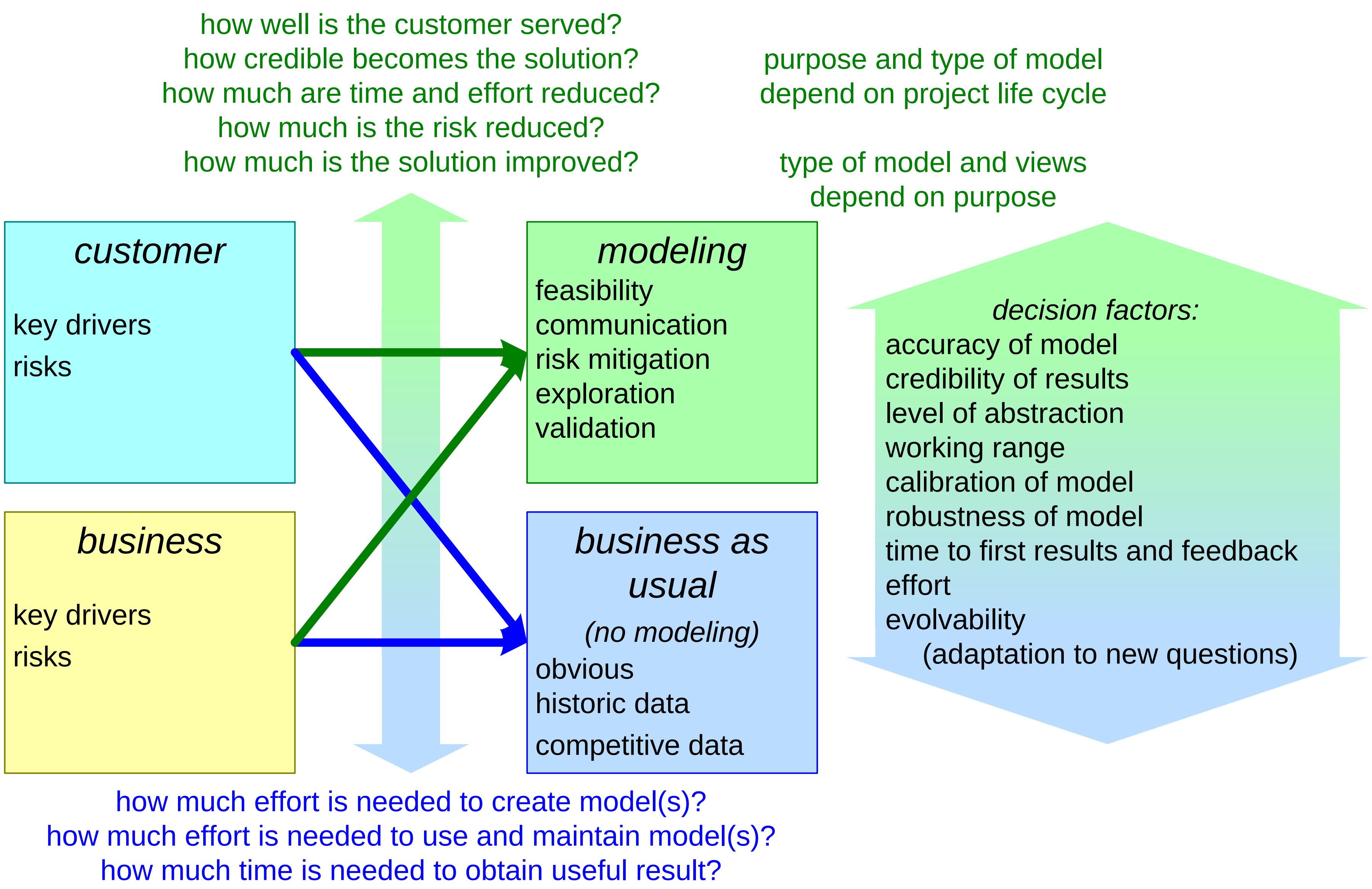
logo
TBD

How to Model?

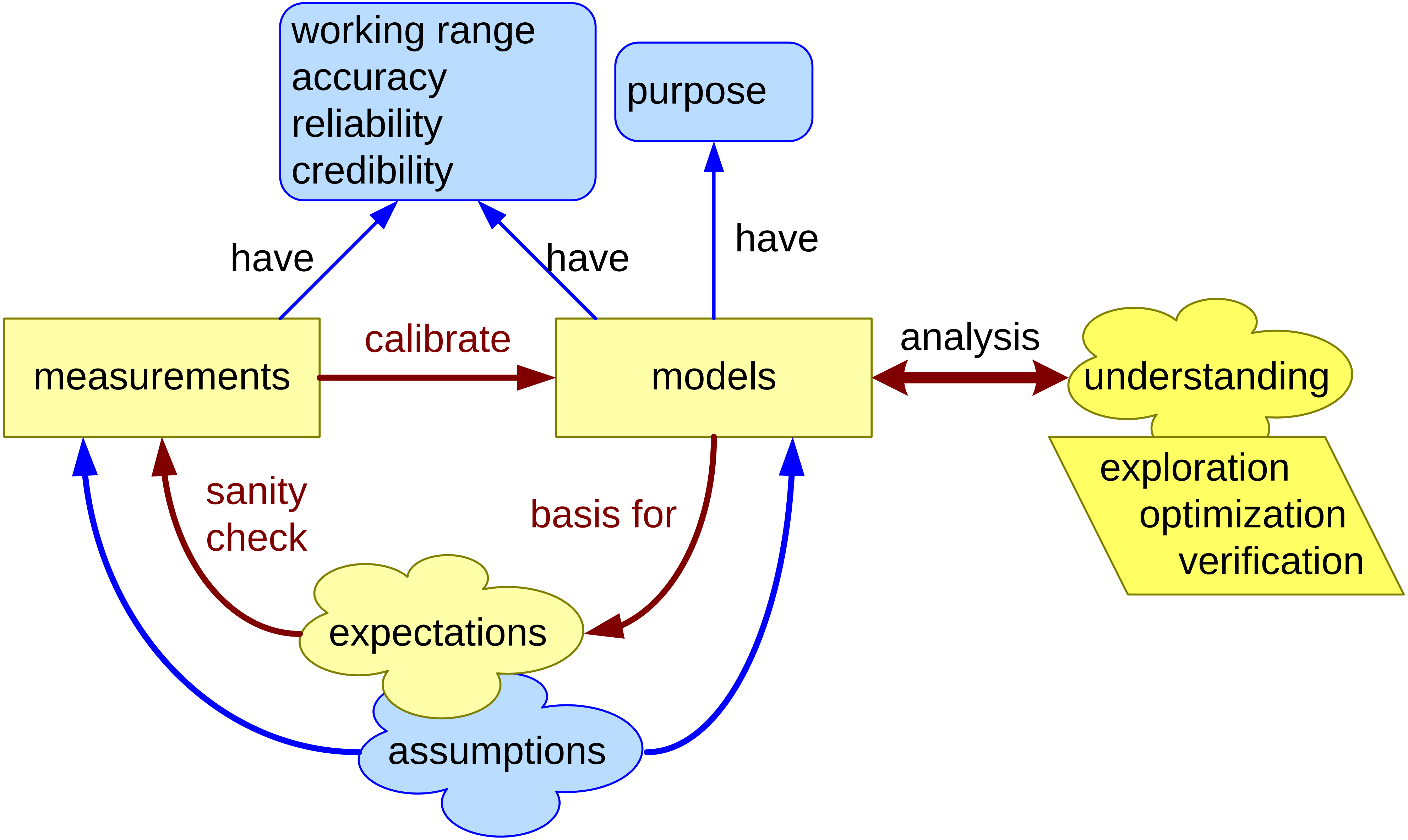


small, simple, goal-driven models

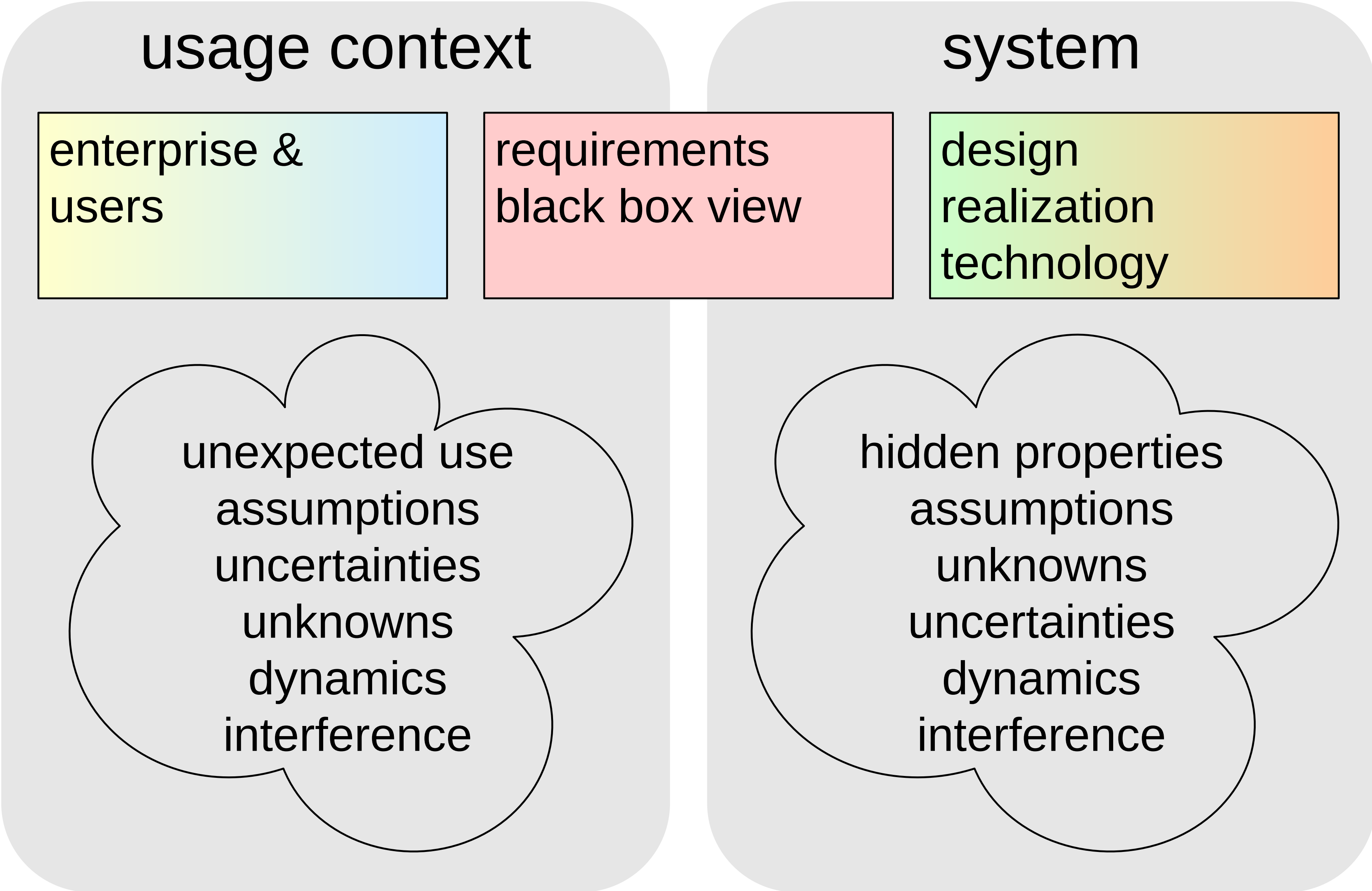
What and Why to Model



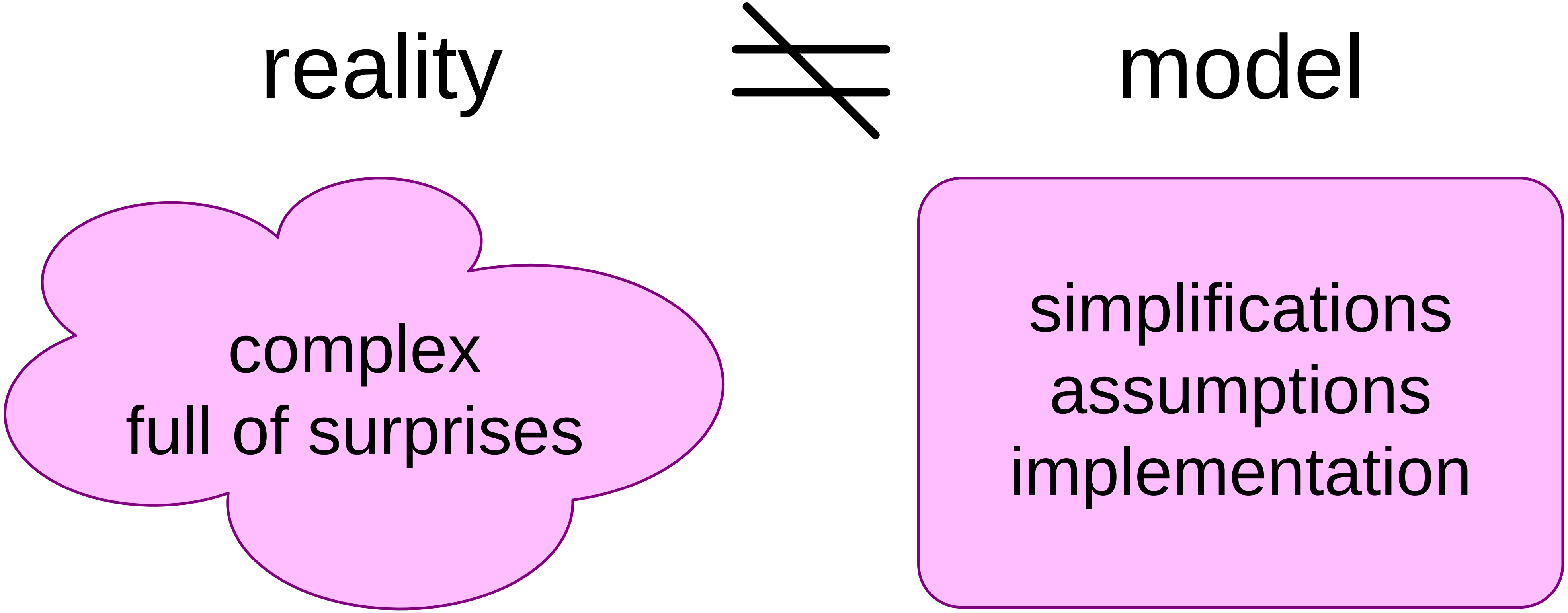
Models, Measurements, Expectations and Assumptions



Unknowns, Uncertainties, ...



Model versus Reality



Starting Points of the Course

practical, immediately applicable in day-to-day work

(inter)active: daily hands-on exercises on case(s)

target: understanding, insight; way-of-working

method, tool, language and domain agnostic

Modeling and Analysis Questions

1. Why do we model? - what are indicators that modeling and analysis beyond "business as usual" architecture is needed. What questions trigger M & A.

2. What do we model? - what kinds of views do we need to consider (4+1, IBM GS Method, Zachman, CAFCR)

3. When do we model? - what models are needed at various points in the project lifecycle.

4. What is the appropriate type of model? - formula, visualization, executable, simulation

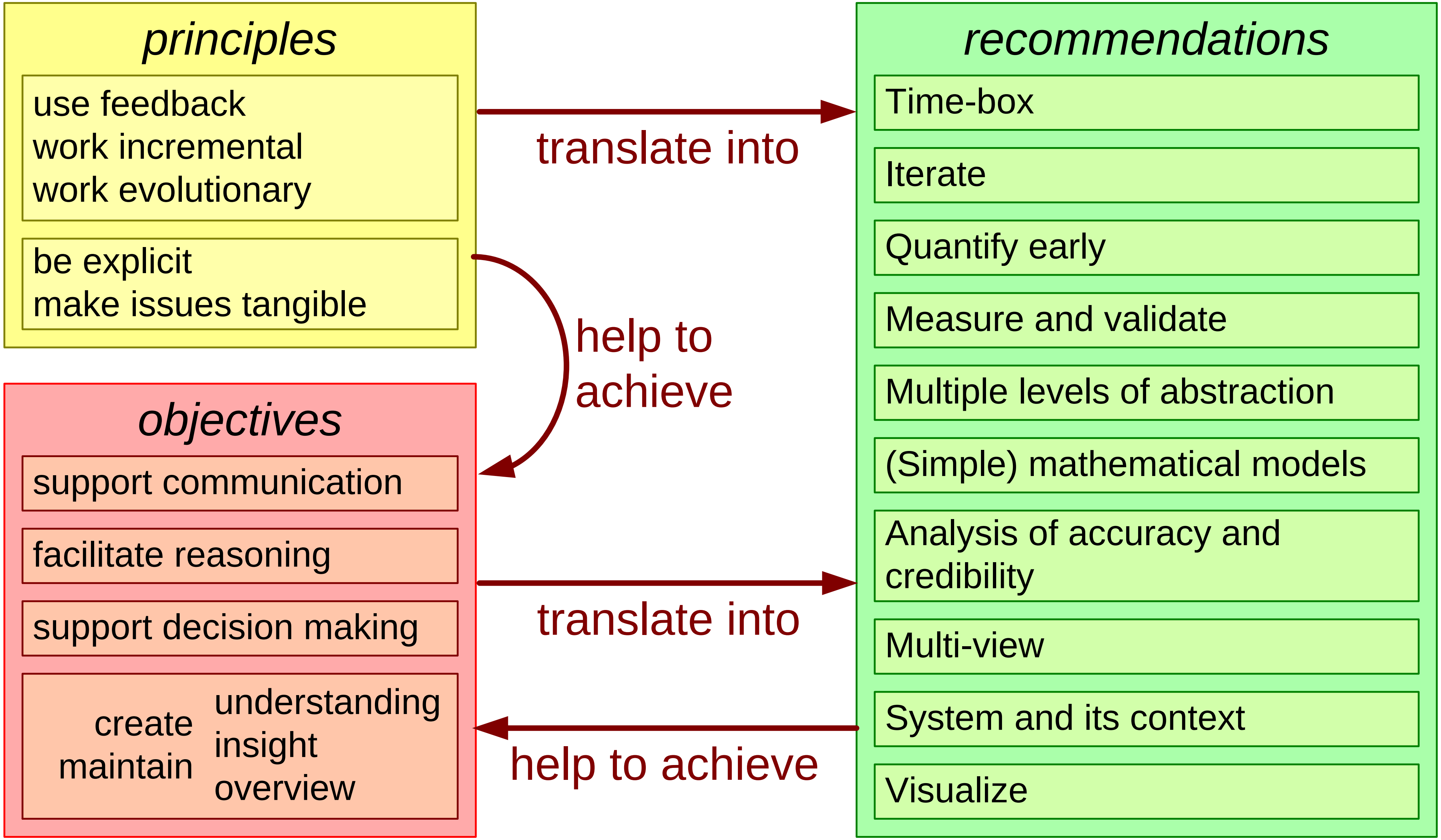
5. What is the required accuracy of the model? - when do we achieve the desired risk mitigation

6. What is the appropriate level of abstraction? - how much details have to be taken into account, versus how much effort can we afford

7. How to calibrate models? - models are based on facts and assumptions. The model outcome deppends strongly on these input data. Note again the tension between effort to make and calibrate versus the value in terms of risk mitigation.

8. How to use models?

Recommendations as Red Thread



Cases for exercise

Electronic Patient Record:

+ relevant health care related information available at the right place for the right person

Long Term Health Care Archive:

+ extreme robust, persistent, high availability archive for large chain of hospitals

Video on Demand Backoffice:

+ large scale content database with fast response download capability including billing, DRM et cetera

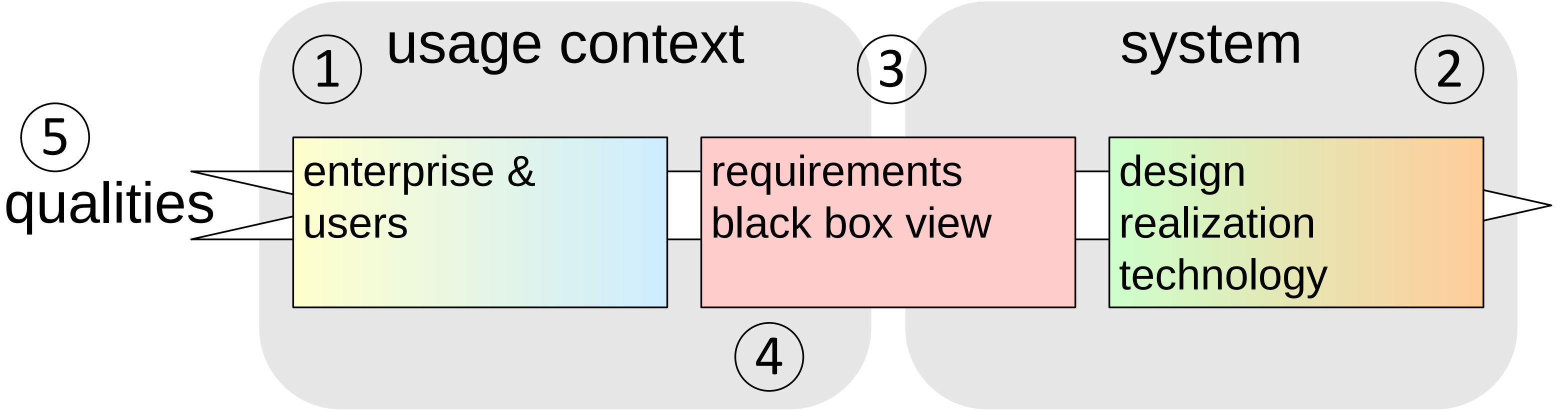
Modeling and Analysis Exercise

- make a quick scan over the following views:

0. what is this exercise about?
1. context: stakeholders, concerns, application
2. system design and realization
3. requirements
4. operational context
5. qualities

- use time boxes of 15 minutes per view
- show the most dominant decomposition(s) of that view, as diagram or as a list; quantify whenever possible

0



Reflection on Exercise

+ collectively we know quite a lot
+ broad overview in short amount of time

~ some "hot" issues appear to be less relevant

- #questions >> #answers

baseline for next refining steps

Modeling and Analysis Summary

Conclusions

Modeling and Analysis must provide more *in-depth* answers for questions that are *breadth* relevant.

Modeling and Analysis is a means that supports *requirements* management, *architecting* and *project* management.

Modeling and Analysis ranges from *business* aspects to *technical* decisions.

Good models are *small*, *simple* and *goal-driven*.

Techniques, Models, Heuristics of this module

Context viewpoints

Fast iteration based on time-boxing

Module Modeling and Analysis: Inputs and Uncertainties

by *Gerrit Muller* LogoESIbuskerud
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

This module addresses Modeling and Analysis: Inputs and Uncertainties. The input for models comes from different sources: facts obtained from market and technology research, data from measurements, and assumptions. All these sources have uncertainties and may hide unknowns, or may even be wrong. We zoom in on commonly used technology.

July 3, 2023
status: planned
version: 0.6

Module Content

goal of this module

Provide foundation and figures of merit for technology modeling

Provide insight in the inputs of models

Provide measurement fundamentals

content of this module

problem statement

generic layering and block diagrams

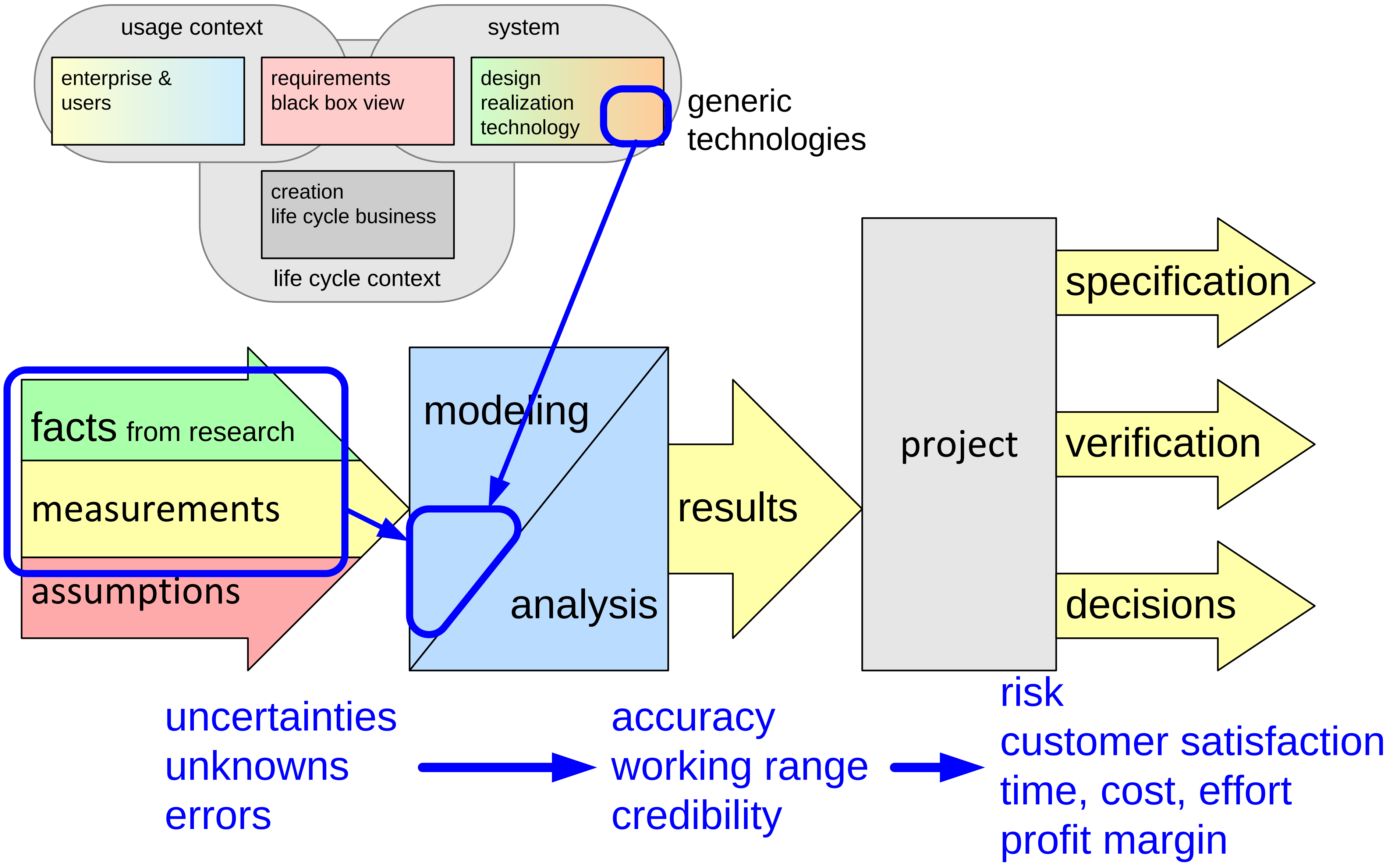
measuring HW and SW

exercise

measurement of loop and file open performance

participants may chose their own programming environment or Python

Where are we in the Course?



Introduction to System Performance Design

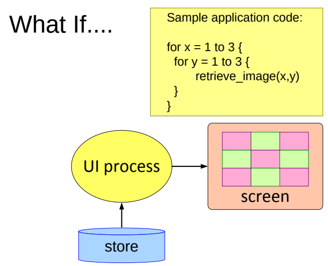
by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

What is System Performance? Why should a software engineer have knowledge of the other parts of the system, such as the Hardware, the Operating System and the Middleware? The applications that he/she writes are self-contained, so how can other parts have any influence? This introduction sketches the problem and shows that at least a high level understanding of the system is very useful in order to get optimal performance.

Distribution

July 3, 2023
status: preliminary draft
version: 0.5



Content of Problem Introduction

content of this presentation

Example of problem

Problem statements

Image Retrieval Performance

application need:

at event 3*3 show 3*3 images instanteneous

design

design

Sample application code:

```
for x = 1 to 3 {
   for y = 1 to 3 {
         retrieve_image(x,y)
   }
}
```

or

alternative application code:

event 3*3 -> show screen 3*3

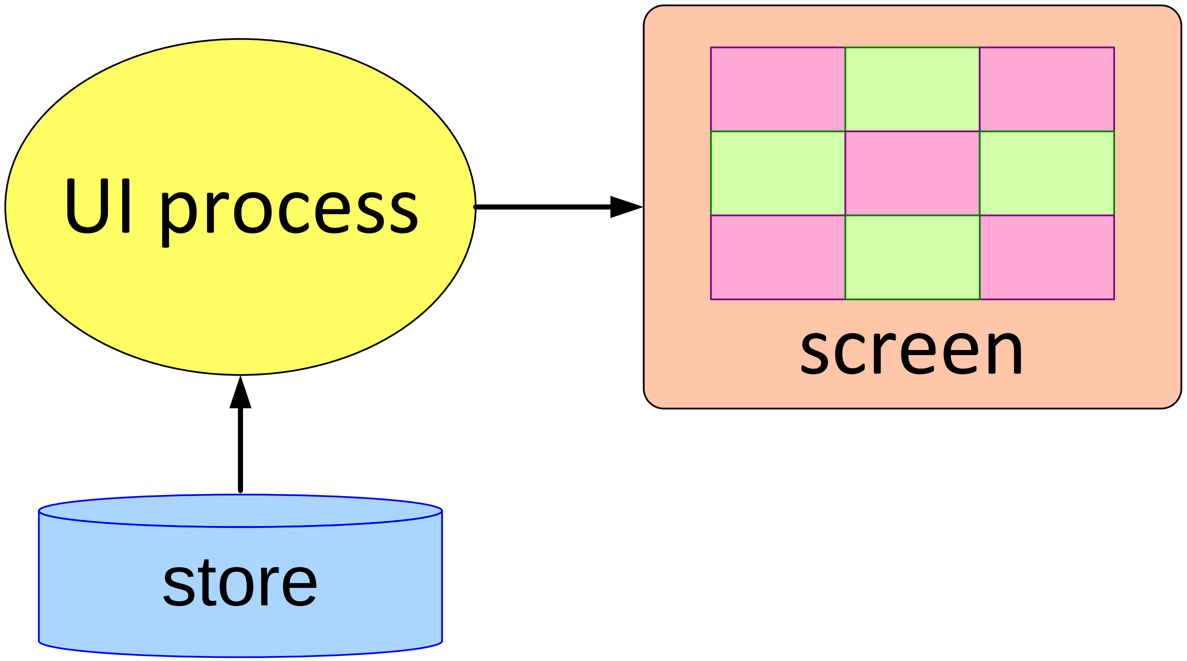
```
<screen 3*3>
    <row 1>
    <col 1><image 1,1></col 1>
    <col 2><image 1,2></col 2>
    <col 3><image 1,3></col 3>
  </row 1>
  <row 2>
    <col 1><image 1,1></col 1>
    <col 2><image 1,2></col 2>
    <col 3><image 1,3></col 3>
  </row 1>
  <row 2>
    <col 1><image 1,1></col 1>
    <col 2><image 1,2></col 2>
    <col 3><image 1,3></col 3>
  </row 3>
</screen 3*3>
```

Straight Forward Read and Display

What If....

Sample application code:

```
for x = 1 to 3 {
  for y = 1 to 3 {
        retrieve_image(x,y)
  }
}
```

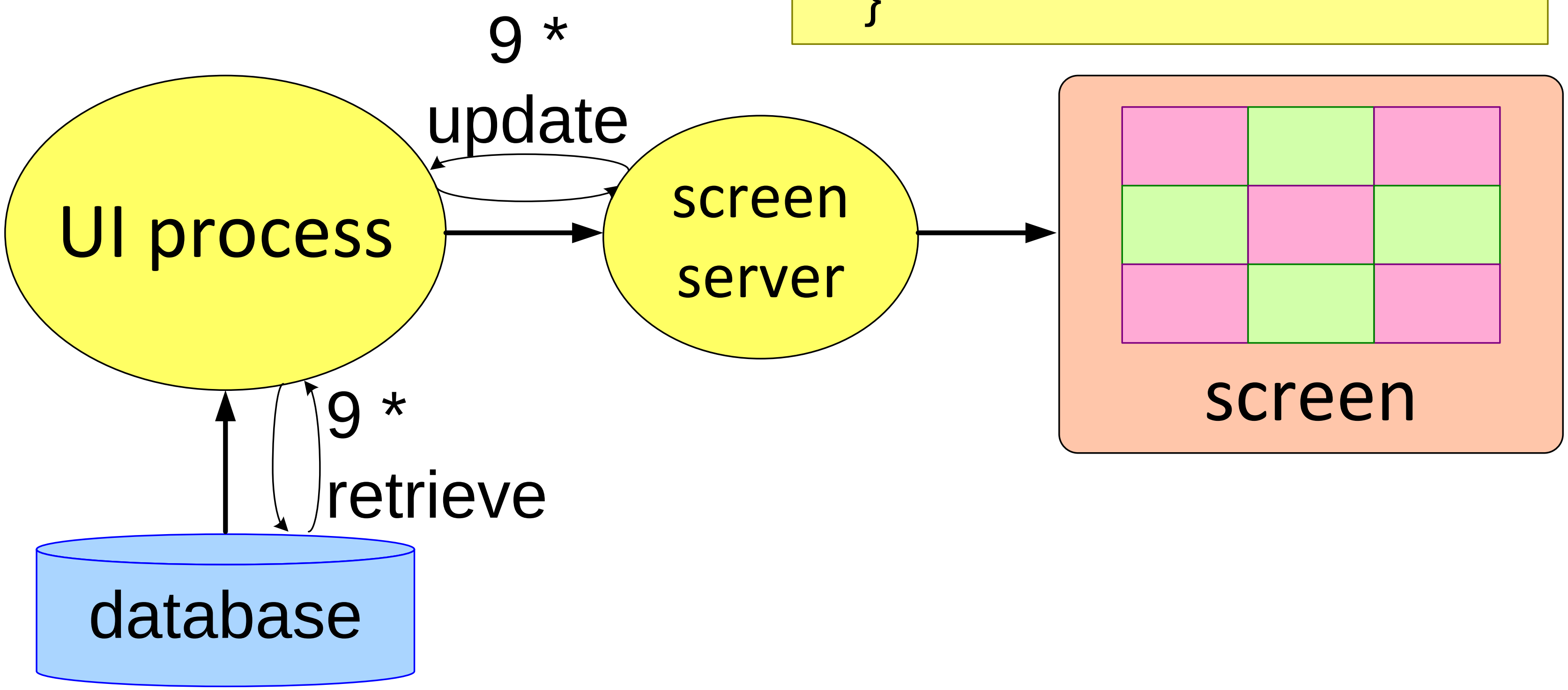


More Process Communication

What If....

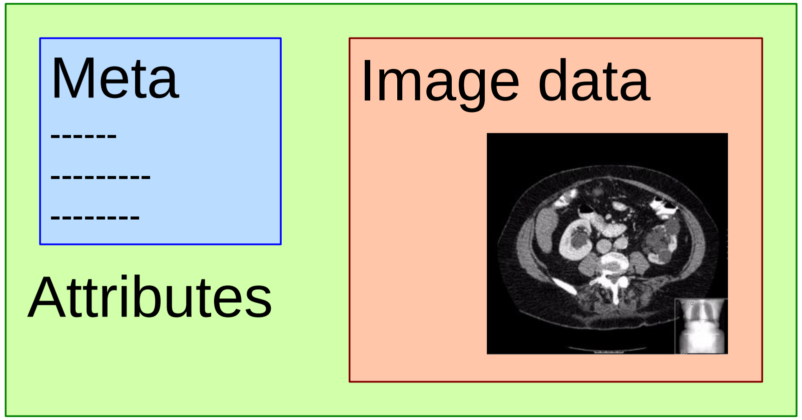
Sample application code:

```
for x = 1 to 3 {
  for y = 1 to 3 {
        retrieve_image(x,y)
  }
}
```



Meta Information Realization Overhead

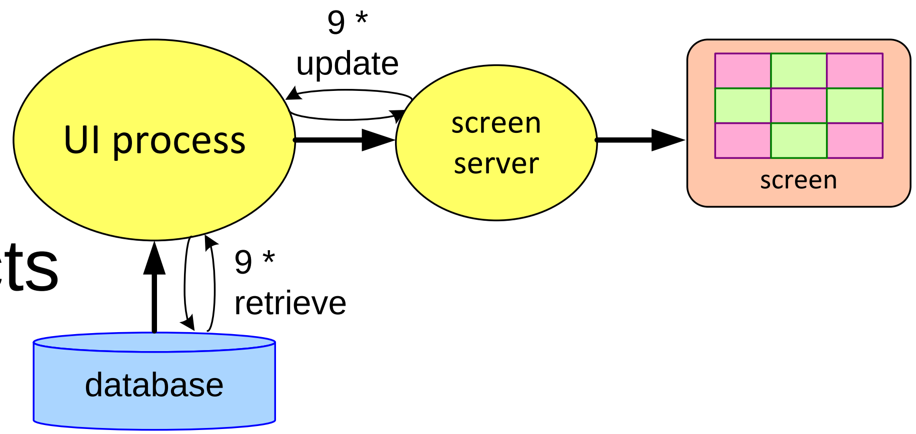
What If....



Sample application code:

```
for x = 1 to 3 {
  for y = 1 to 3 {
        retrieve_image(x,y)
  }
}
```

Attribute = 1 COM object
100 attributes / image
9 images = 900 COM objects
1 COM object = 80µs
9 images = 72 ms



What If....

Sample application code:

```
for x = 1 to 3 {
  for y = 1 to 3 {
        retrieve_image(x,y)
  }
}
```

- I/O on line basis (512^2 image)

$$9 * 512 * t_{I/O}$$

$$t_{I/O} \sim= 1ms$$

- . . .

Non Functional Requirements Require System View

Sample application code:

```
for x = 1 to 3 {
  for y = 1 to 3 {
        retrieve_image(x,y)
  }
}
```

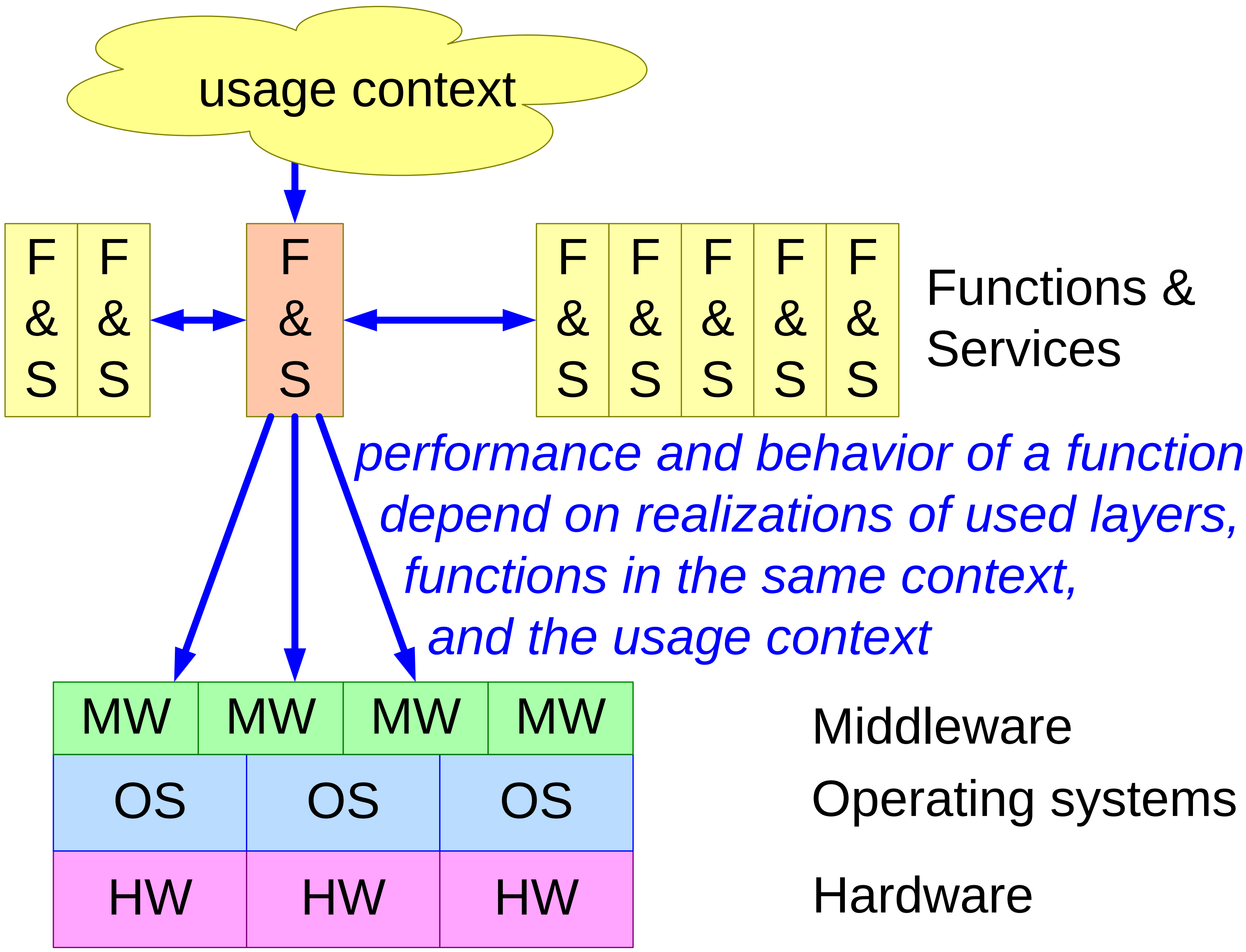
can be:

- fast, but very local
- slow, but very generic
- slow, but very robust
- fast and robust
- ...

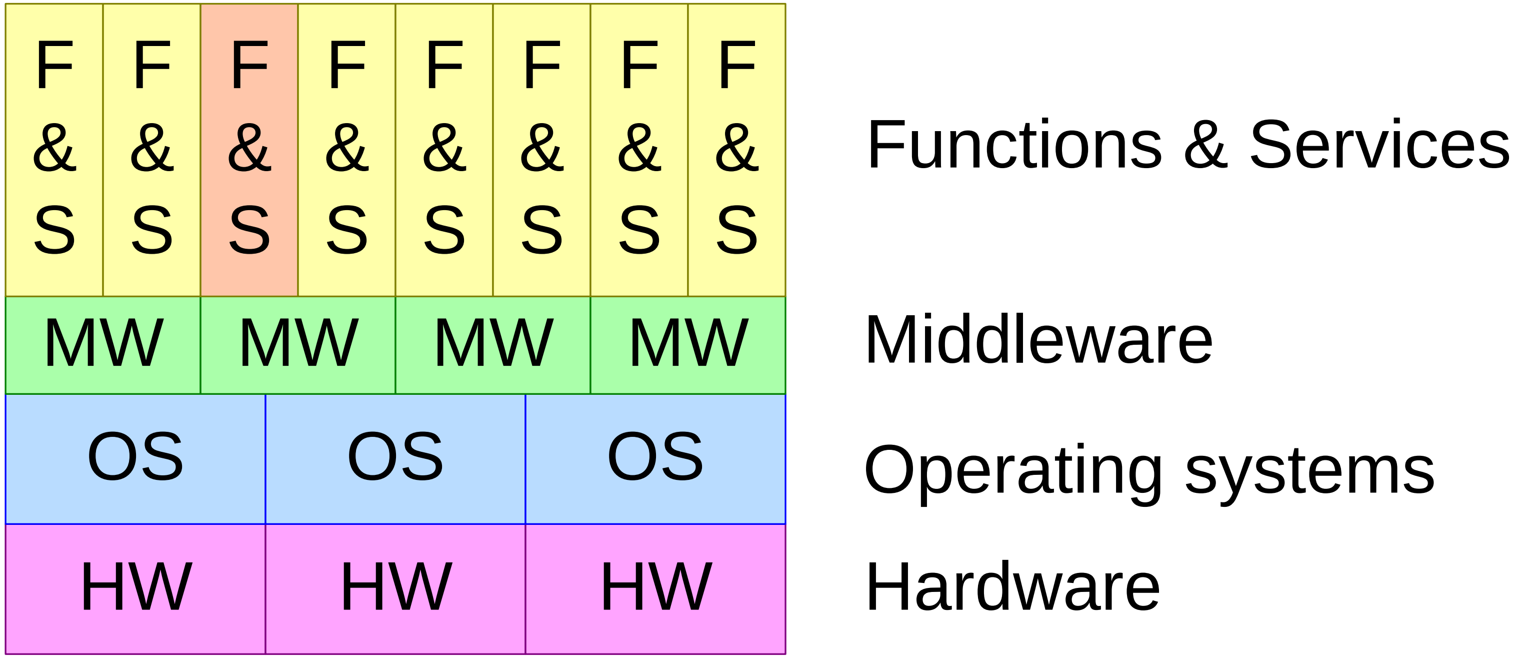
The emerging properties (behavior, performance) cannot be seen from the code itself!

Underlying platform and neighbouring functions determine emerging properties mostly.

Function in System Context



Challenge



Performance = Function (F&S, other F&S, MW, OS, HW)
MW, OS, HW >> 100 Manyear : very complex

Challenge: How to understand MW, OS, HW
with only a few parameters

Summary of Problem Introduction

Summary of Introduction to Problem

Resulting System Characteristics cannot be deduced from local code.

Underlying platform, neighboring applications and user context:

have a big impact on system characteristics

are big and complex

Models require decomposition, relations and representations to analyse.

Intermezzo: So Why to Model and Analyse?

Why do we model?

- what are indicators that modeling and analysis beyond "business as usual" architecture is needed.

- What questions trigger Modeling and Analysis.

The answer to the question from business side is *not evident*

The answer is business *critical* (e.g. poor performance -> unusable service). We did not discuss business value for this case.

Past experience shows that design choices have big impact on the outcome, in other words this part of the design is *critical*

Modeling and Analysis Fundamentals of Technology

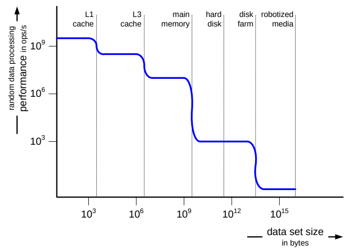
by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This presentation shows fundamental elements for models that are ICT-technology related. Basic hardware functions are discussed: storage, communication and computing with fundamental characteristics, such as throughput, latency, and capacity. A system is build by layers of software on top of hardware. The problem statement is how to reason about system properties, when the system consists of many layers of hardware and software.

Distribution

July 3, 2023
status: preliminary draft
version: 0.5



Presentation Content Fundamentals of Technology

content of this presentation

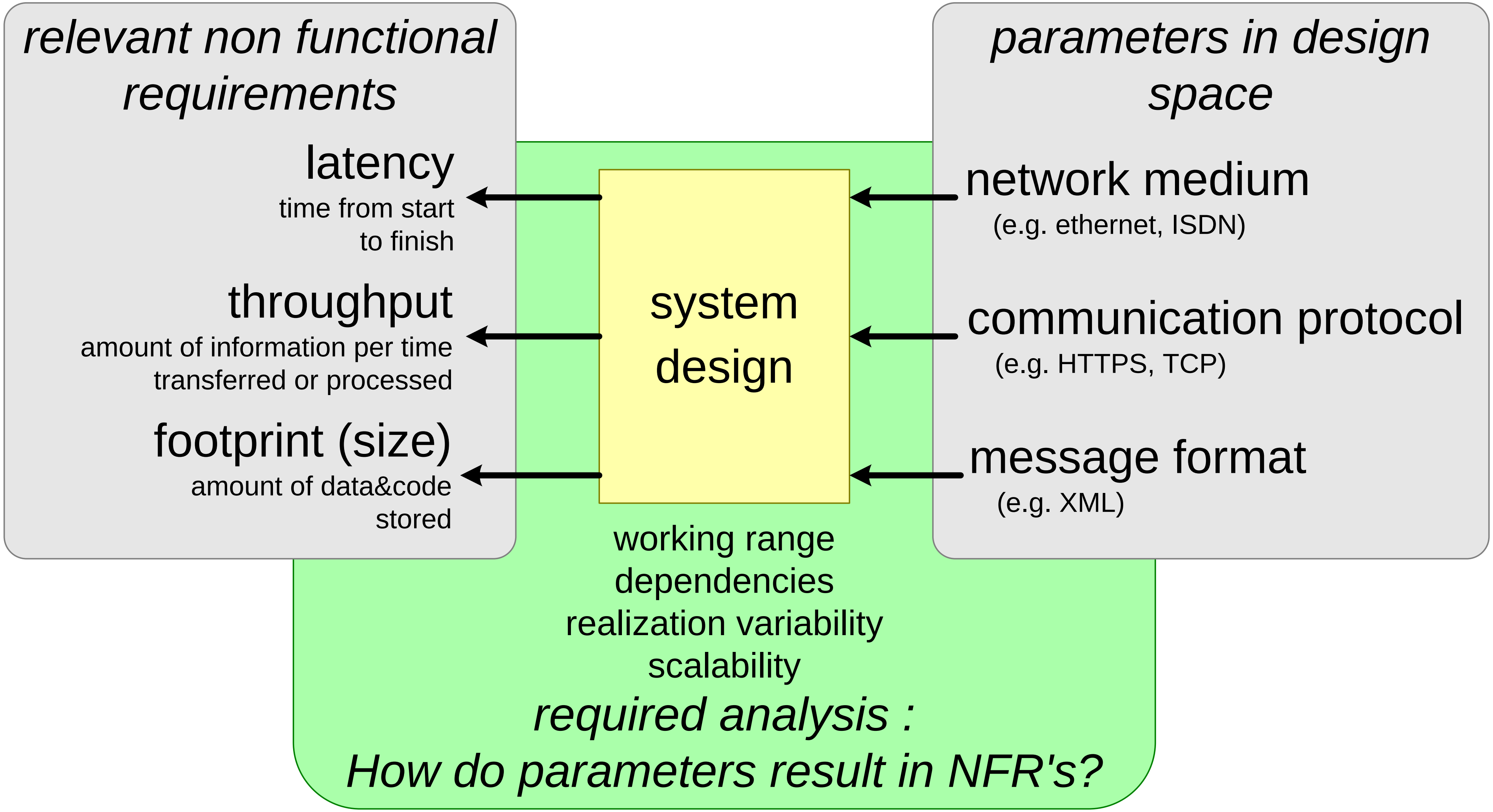
generic layering and block diagrams

typical characteristics and concerns

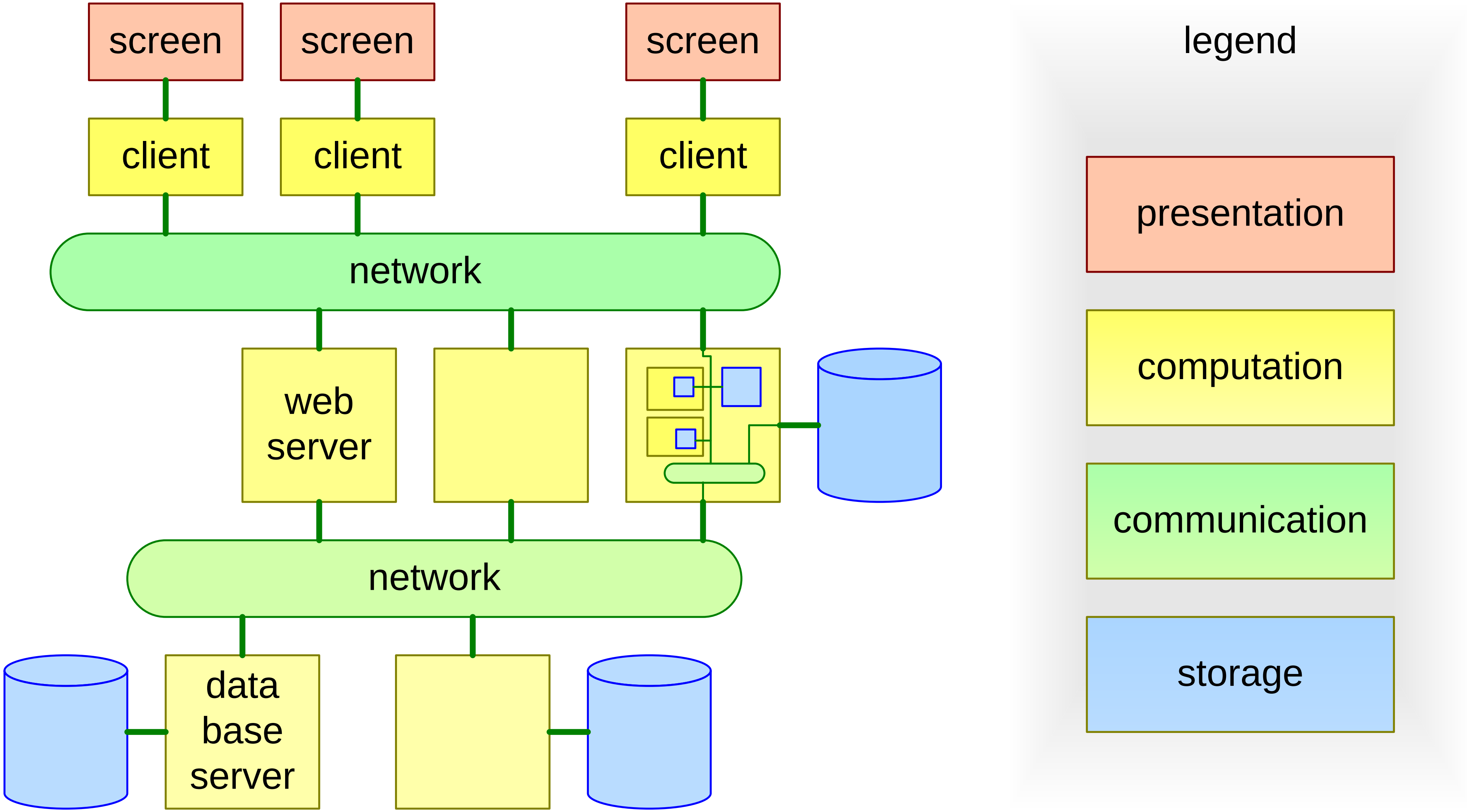
figures of merit

example of picture caching in web shop application

What do We Need to Analyze?



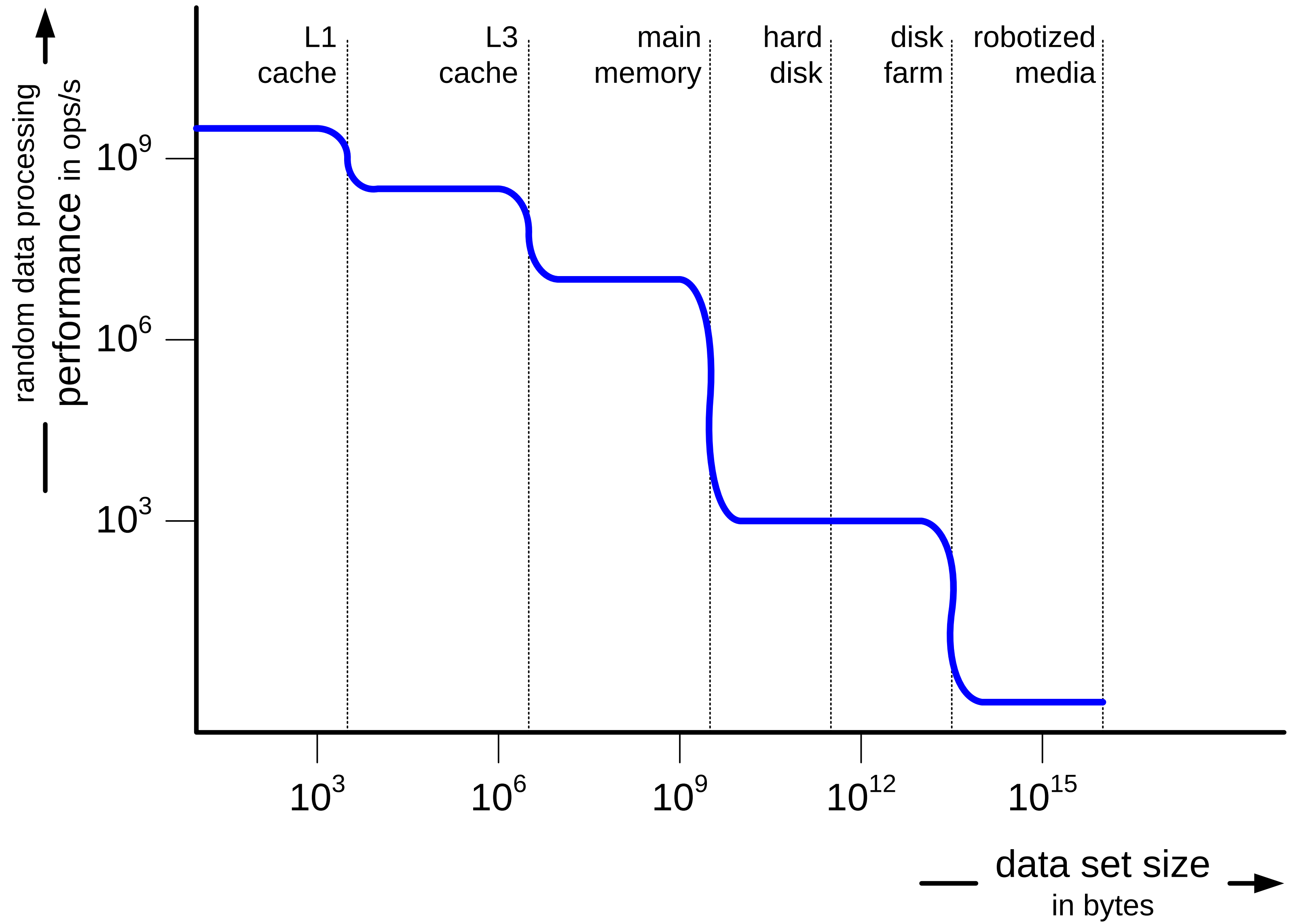
Typical Block Diagram and Typical Resources



Hierarchy of Storage Technology Figures of Merit

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

Performance as Function of Data Set Size



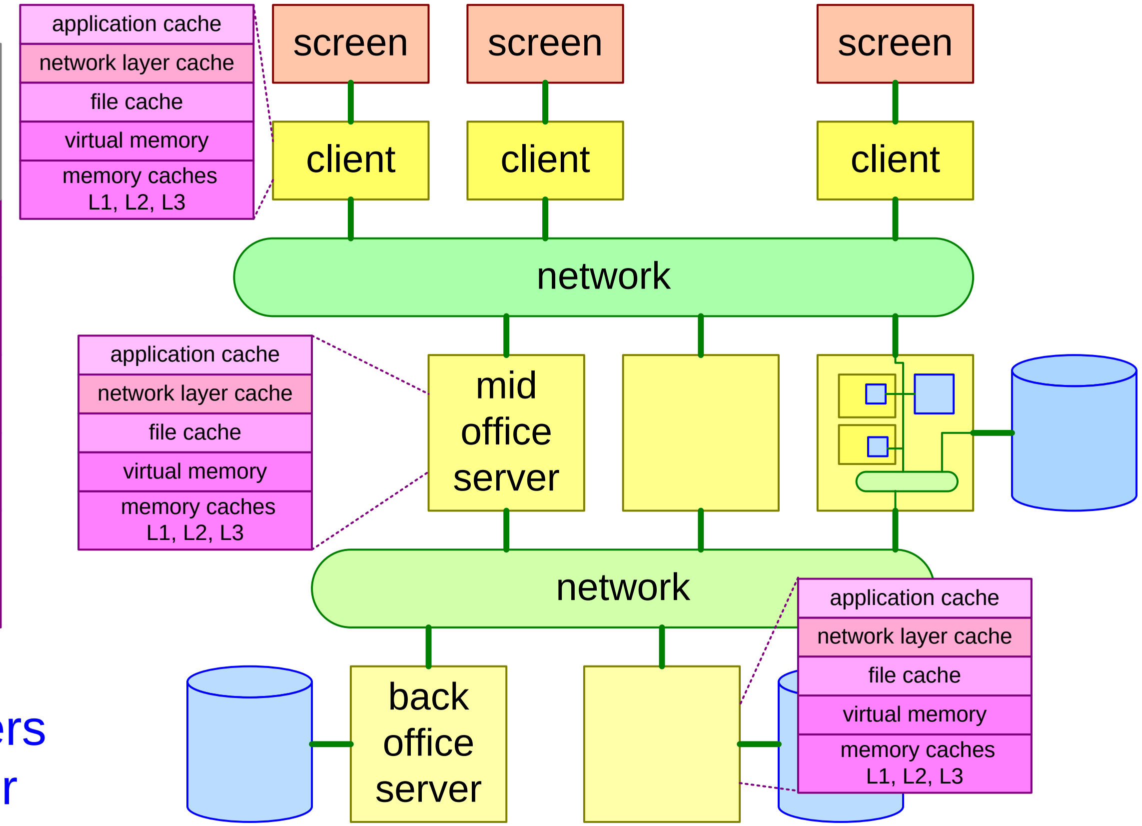
Communication Technology Figures of Merit

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

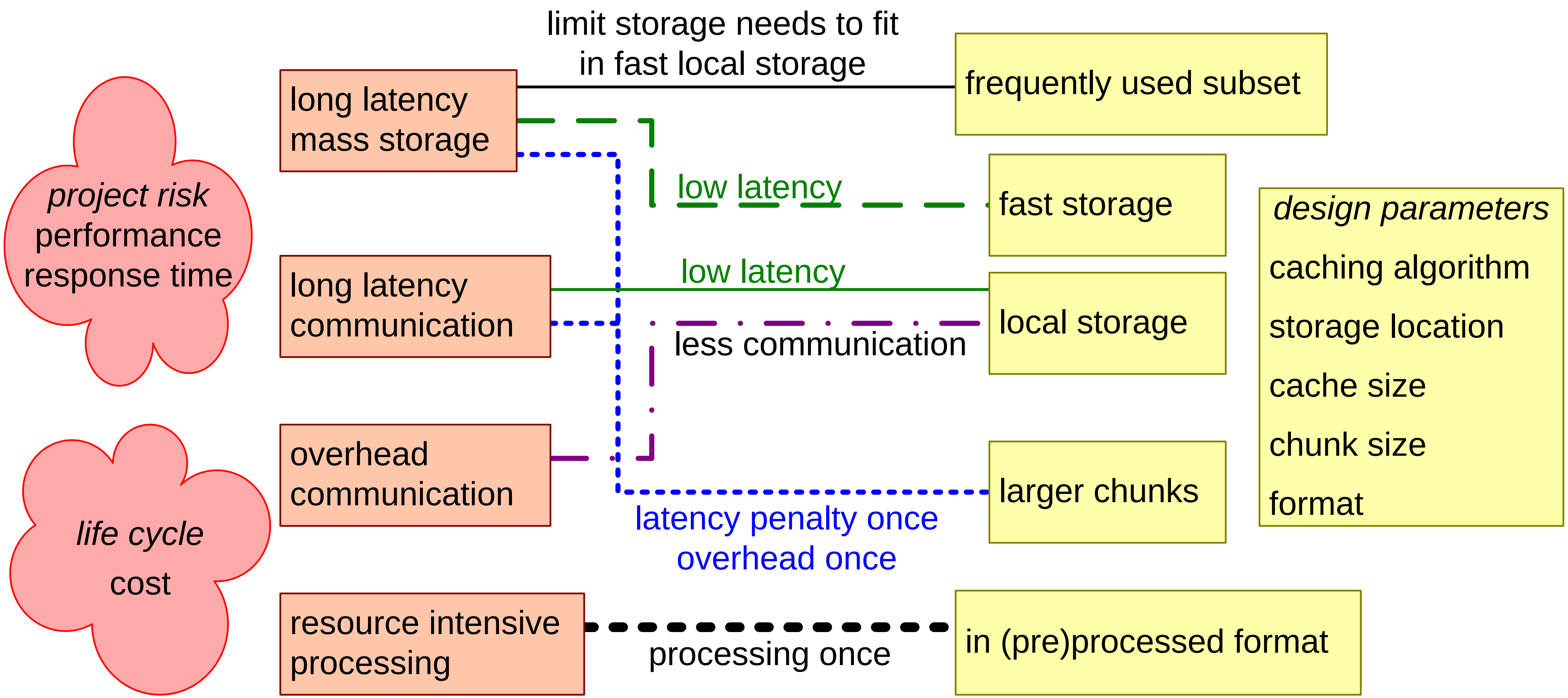
Multiple Layers of Caching

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

typical cache 2 orders
of magnitude faster

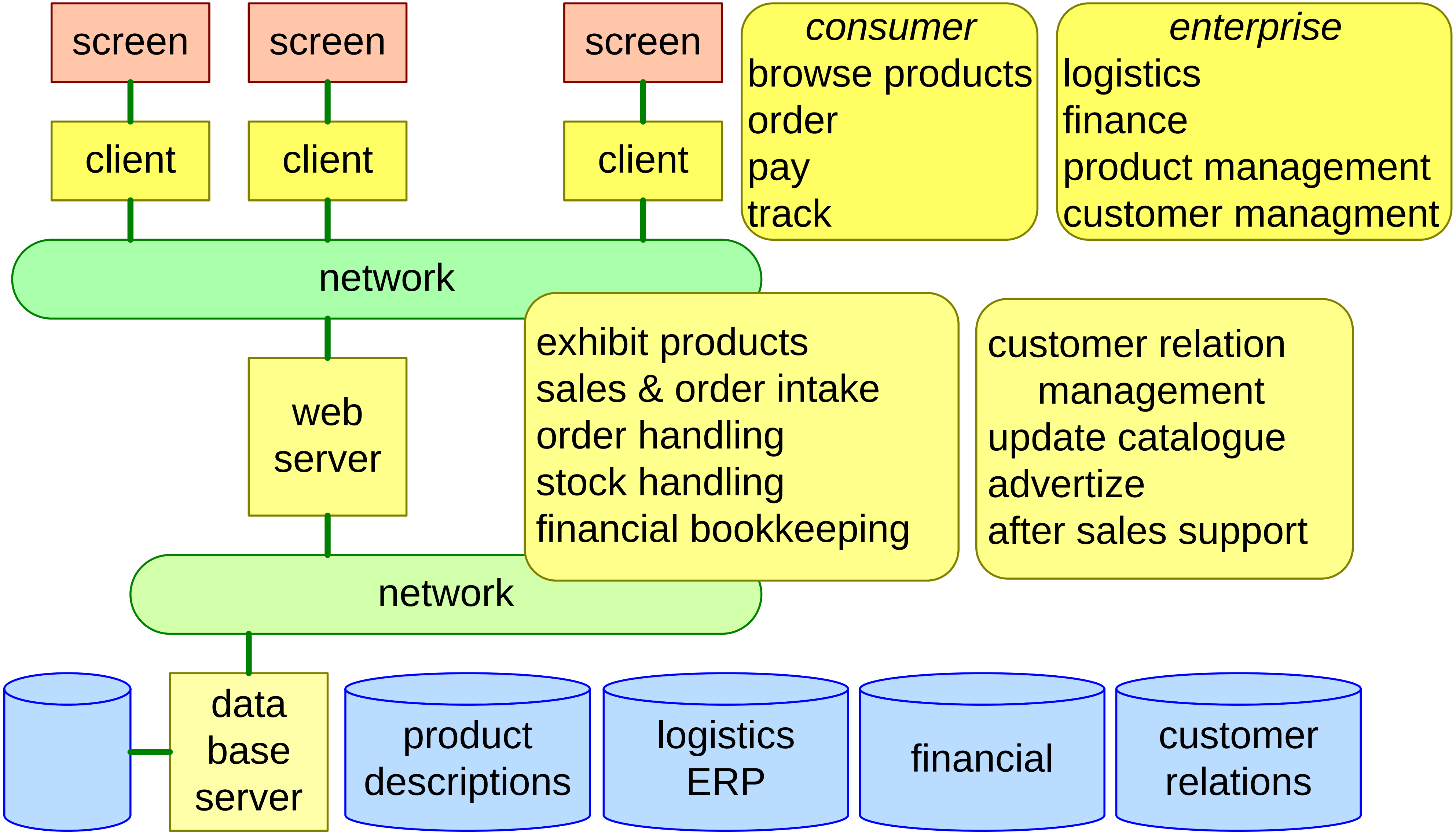


Why Caching?

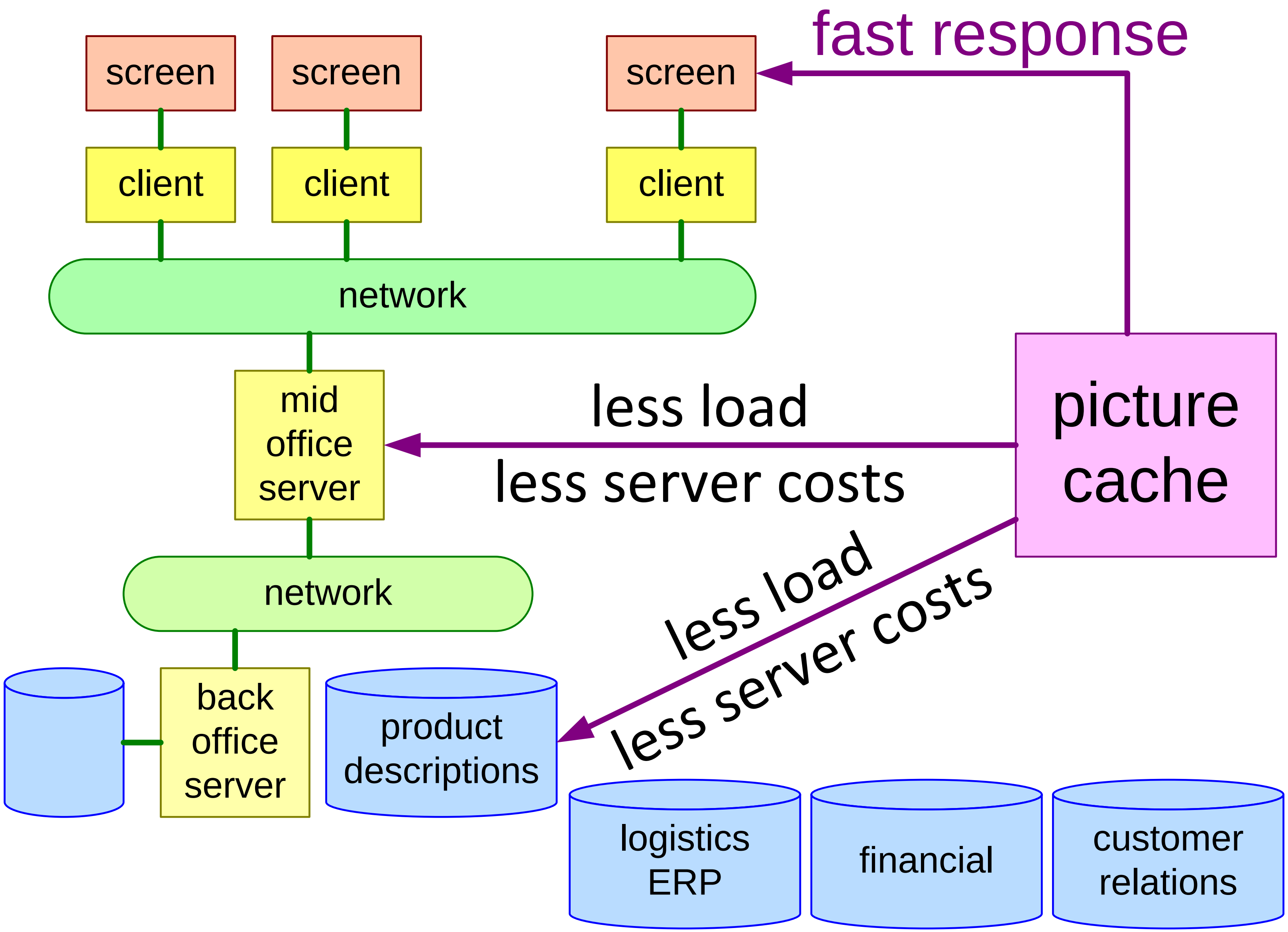


project risk
performance
response time
life cycle
cost
long latency
mass storage
long latency
communication
overhead
communication
resource intensive
processing
limit storage needs to fit
in fast local storage
low latency
low latency
less communication
latency penalty once
overhead once
processing once
frequently used subset
fast storage
local storage
larger chunks
in (pre)processed format
design parameters
caching algorithm
storage location
cache size
chunk size
format

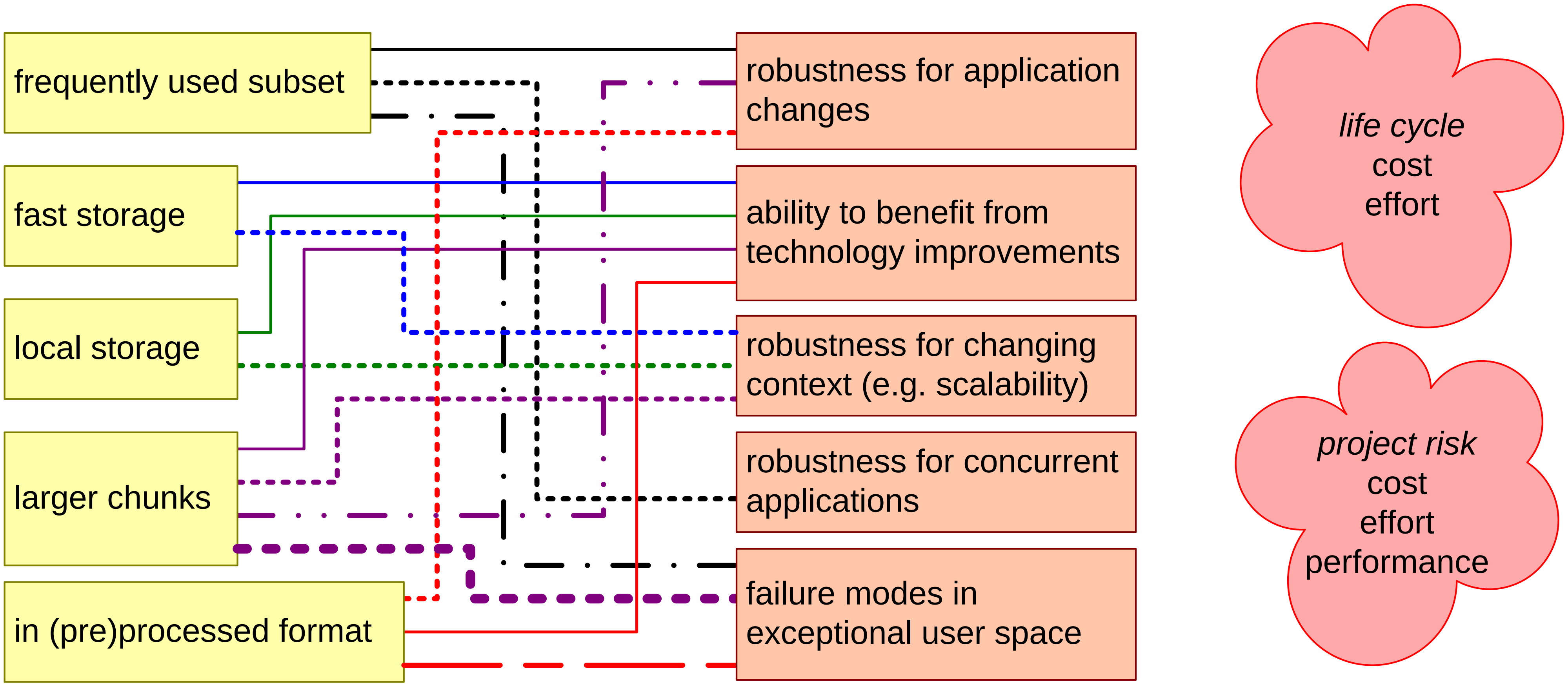
Example Web Shop



Impact of Picture Cache



Risks of Caching



Summary

Conclusions

Technology characteristics can be discontinuous

Caches are an example to work around discontinuities

Caches introduce complexity and decrease transparancy

Techniques, Models, Heuristics of this module

Generic block diagram: Presentation, Computation, Communication and Storage

Figures of merit

Local reasoning (e.g. cache example)

Modeling and Analysis: Measuring

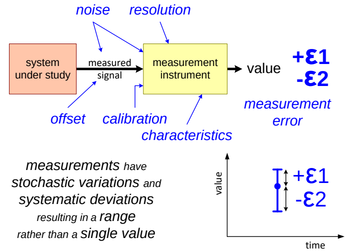
by *Gerrit Muller* University of South-Eastern Norway-SE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This presentation addresses the fundamentals of measuring: What and how to measure, impact of context and experiment on measurement, measurement errors, validation of the result against expectations, and analysis of variation and credibility.

Distribution

July 3, 2023
status: preliminary draft
version: 1.2



Presentation Content

content

What and How to measure

Impact of experiment and context on measurement

Validation of results, a.o. by comparing with expectation

Consolidation of measurement data

Analysis of variation and analysis of credibility

Measuring Approach: What and How

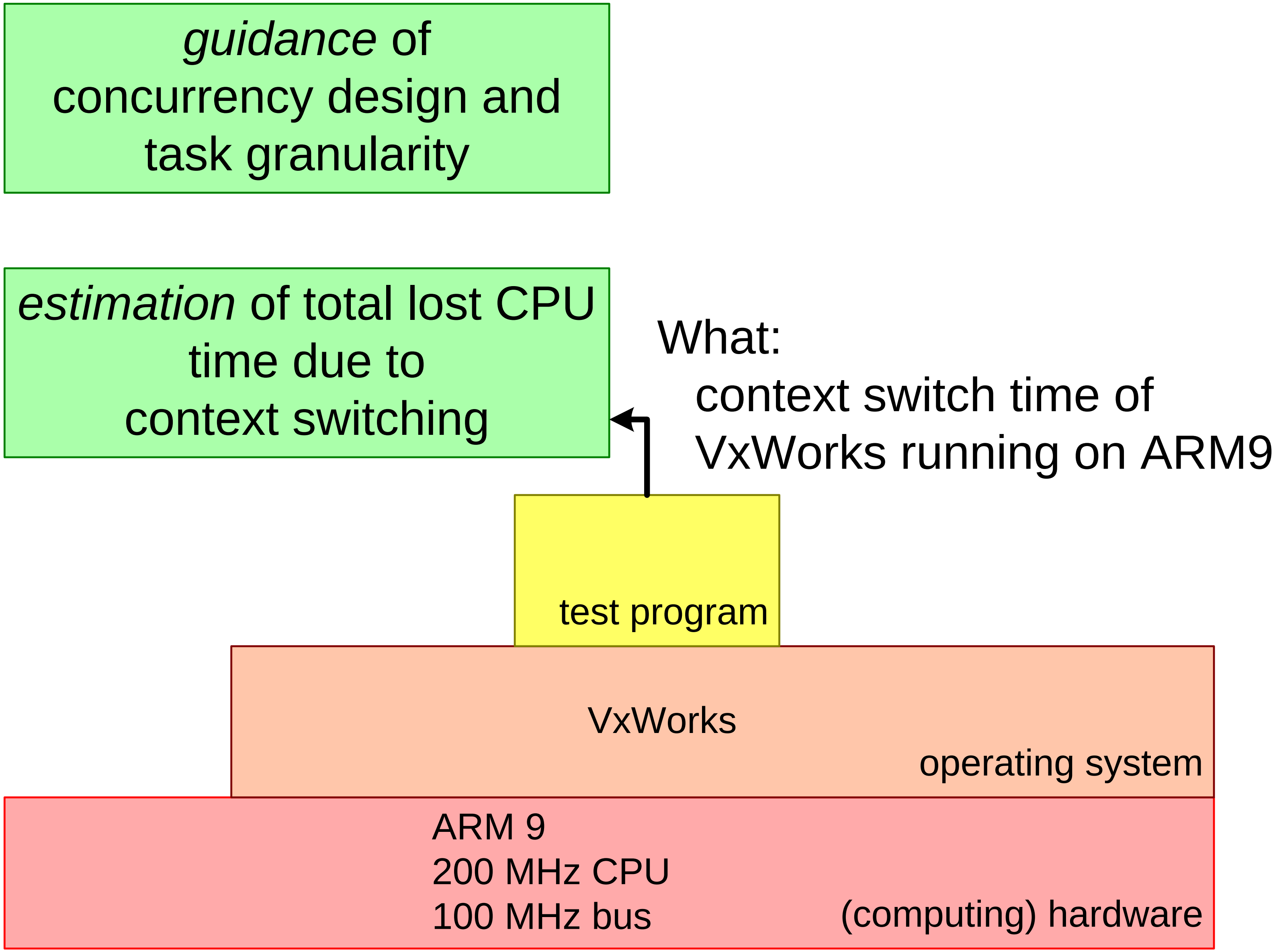
what

1. What do we need to know?	
2. Define quantity to be measured.	initial model
3. Define required accuracy	purpose
4A. Define the measurement circumstances	fe.g. by use cases
4B. Determine expectation	historic data or estimation
4C. Define measurement set-up	
5. Determine actual accuracy	uncertainties, measurement error
6. Start measuring	
7. Perform sanity check	expectation versus actual outcome

iterate

how

1. What do We Need? Example Context Switching

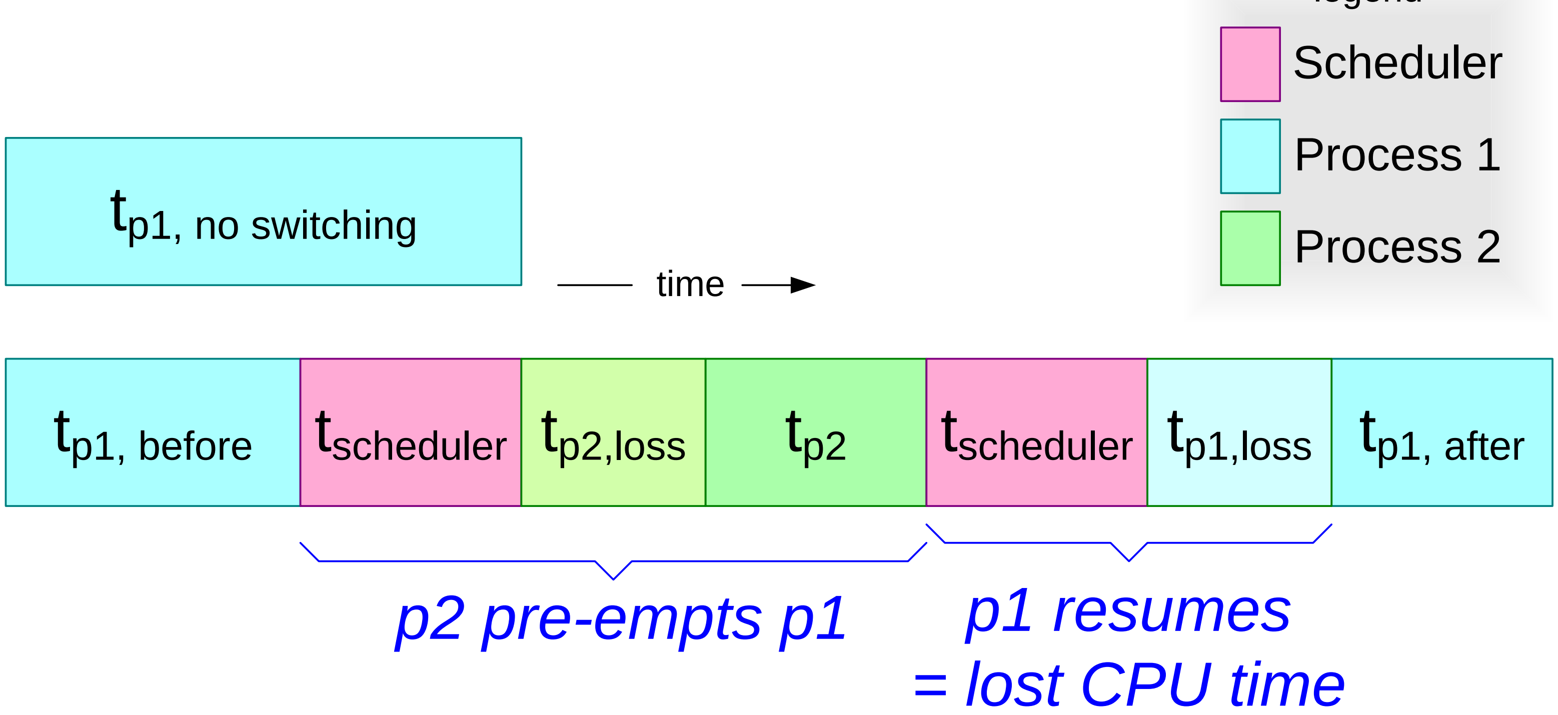


2. Define Quantity by Initial Model

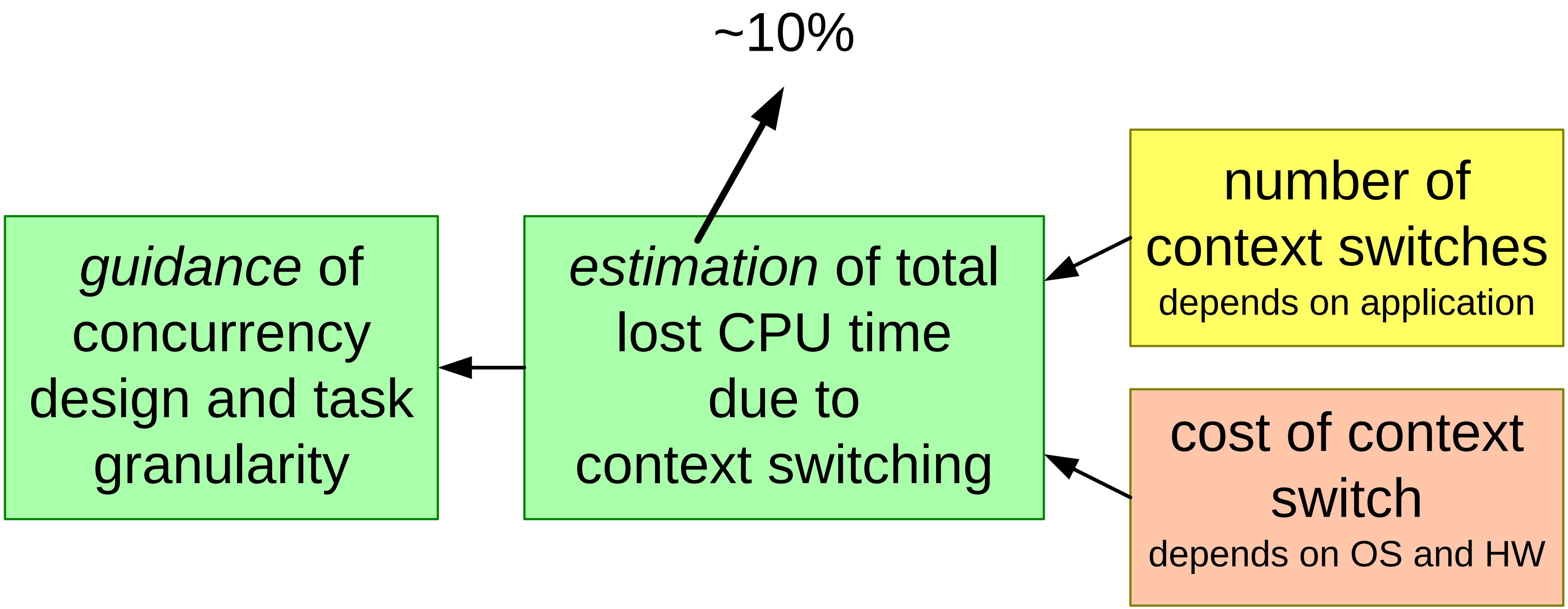
What (original):
context switch time of
VxWorks running on ARM9

What (more explicit):
The amount of lost CPU time,
due to context switching on
VxWorks running on ARM9
on a heavy loaded CPU

$$t_{\text{context switch}} = t_{\text{scheduler}} + t_{\text{p1, loss}}$$

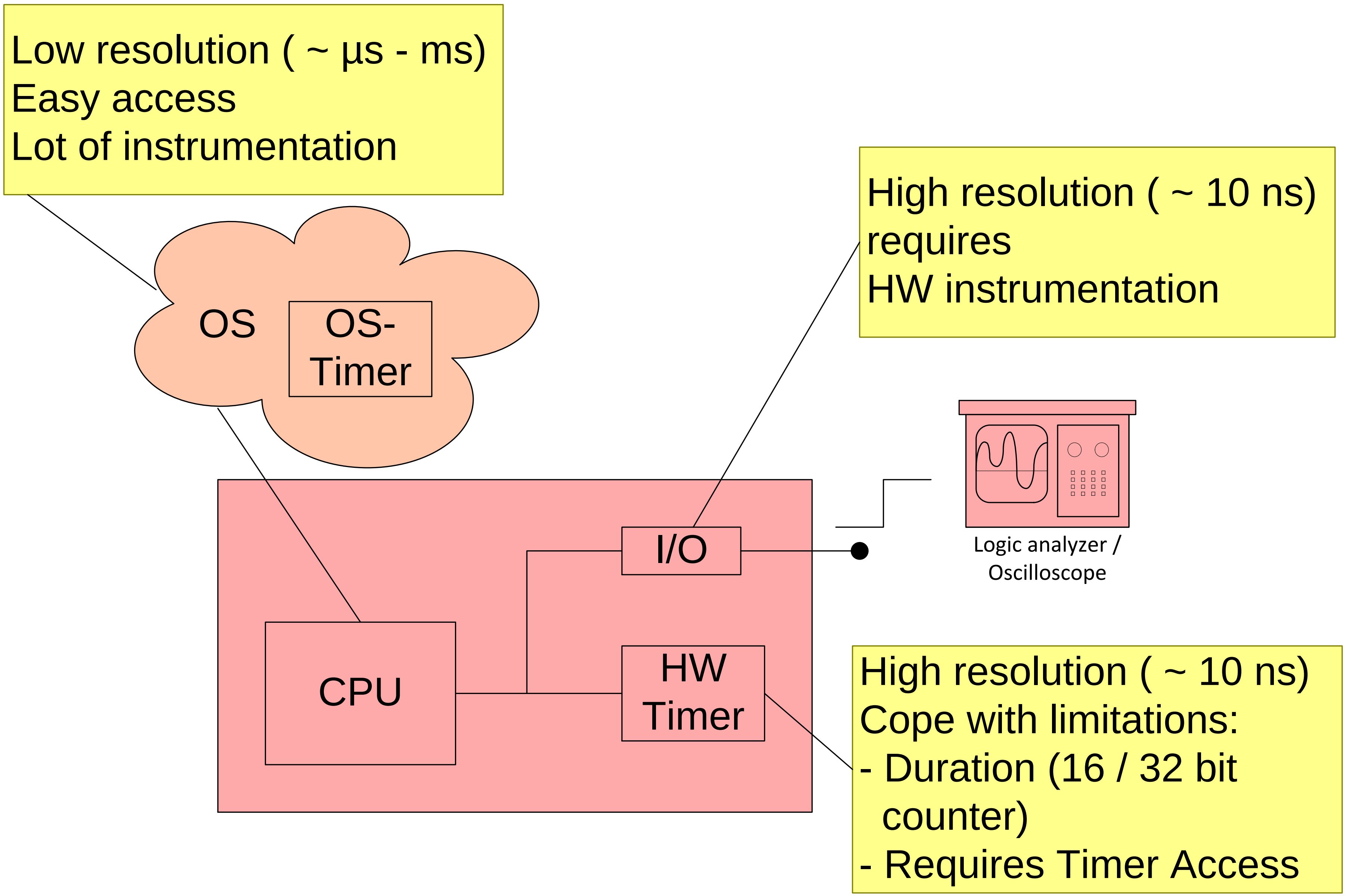


3. Define Required Accuracy



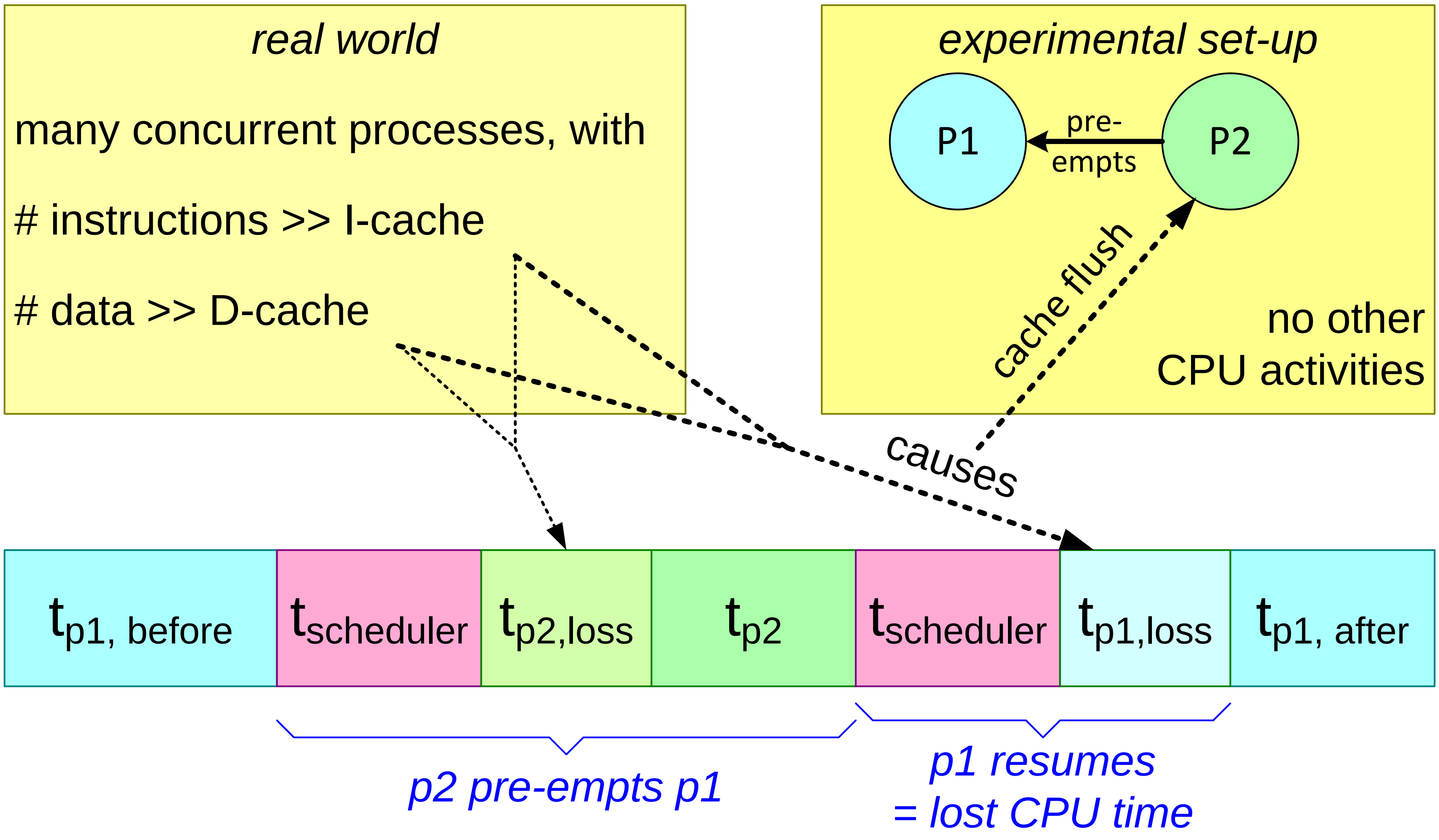
purpose drives required accuracy

Intermezzo: How to Measure CPU Time?

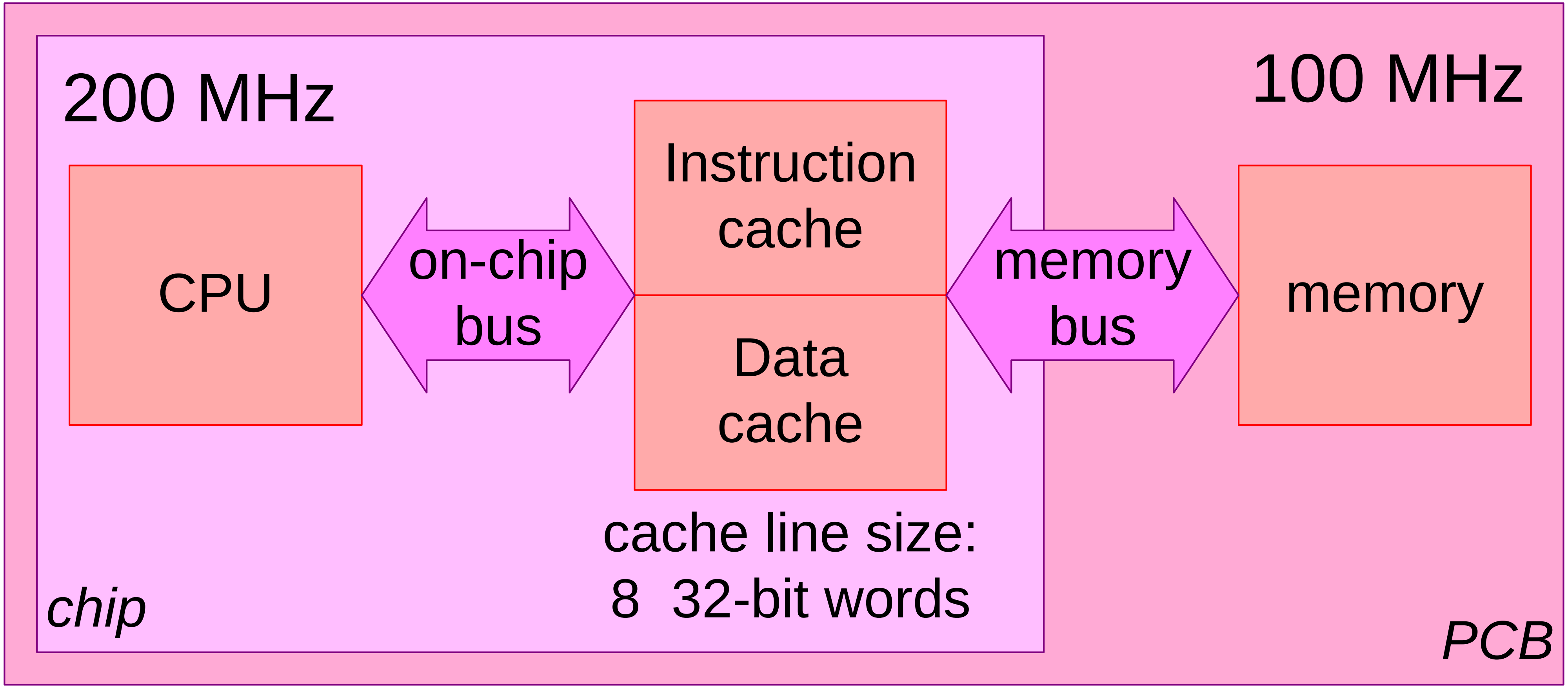


4A. Define the Measurement Set-up

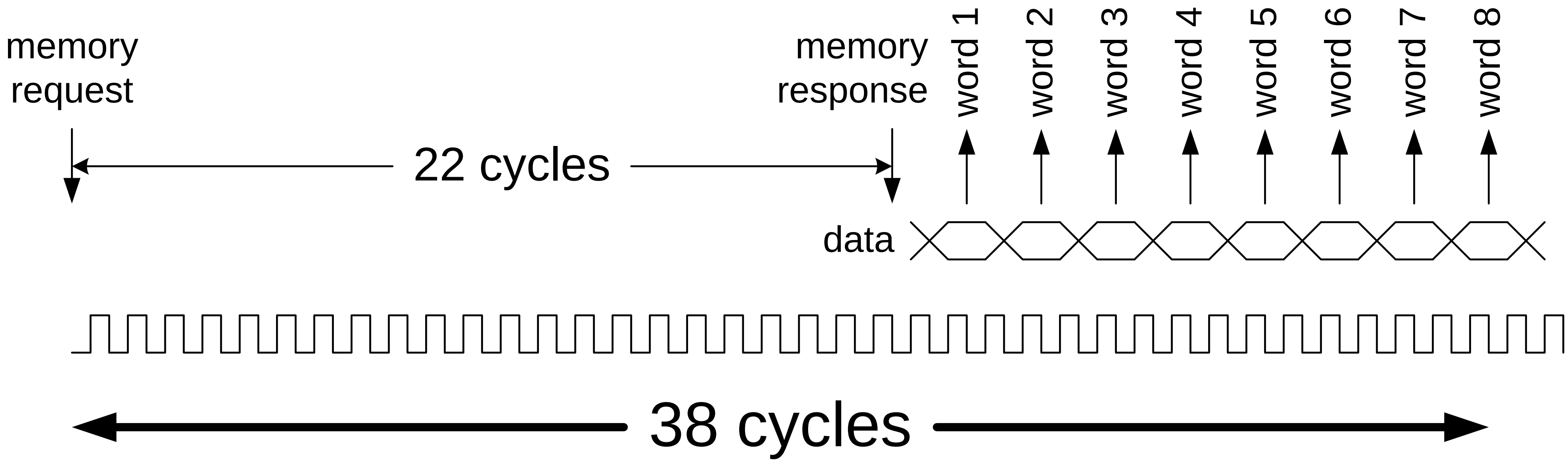
Mimick relevant real world characteristics



4B. Case: ARM9 Hardware Block Diagram

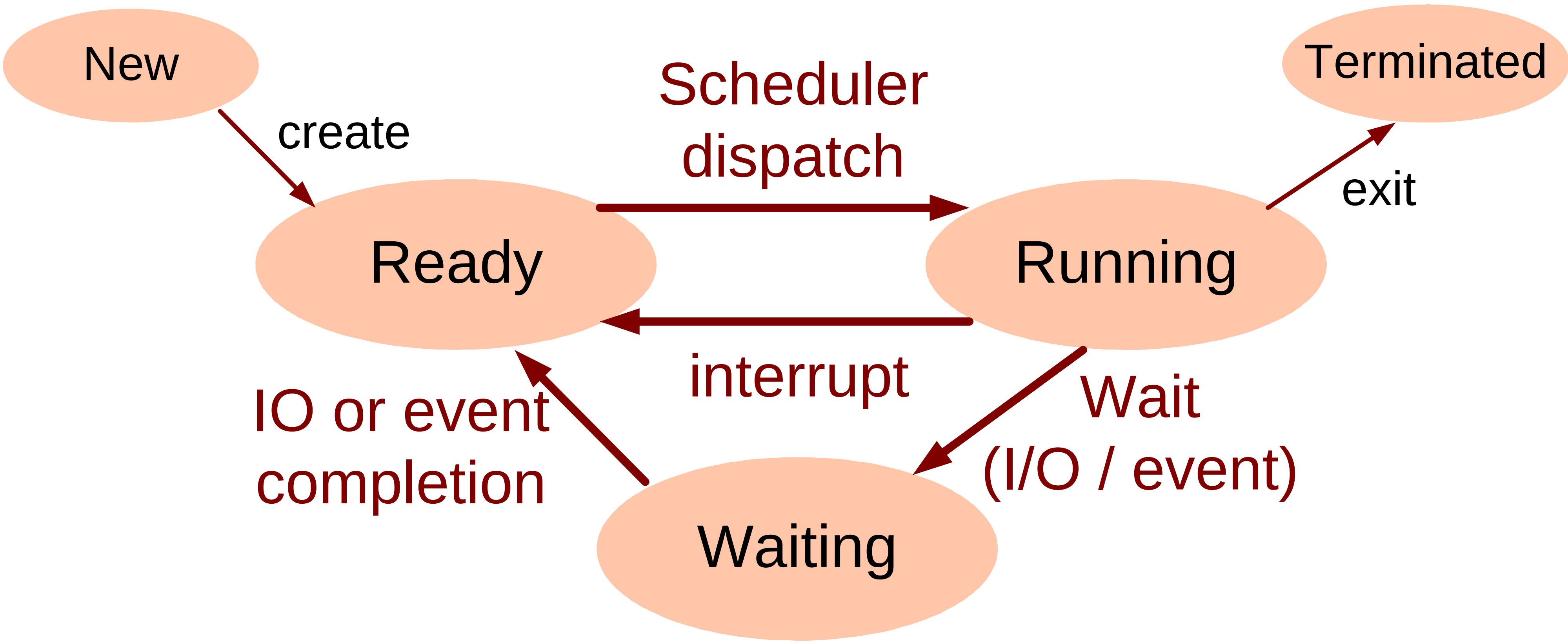


Key Hardware Performance Aspect



memory access time in case of a cache miss
200 Mhz, 5 ns cycle: 190 ns

OS Process Scheduling Concepts



Determine Expectation

simple SW model of context switch:

- save state P1
- determine next runnable task
- update scheduler administration
- load state P2
- run P2

Estimate how many instructions and memory accesses are needed per context switch

input data HW:

$t_{\text{ARM instruction}} = 5$ ns

$t_{\text{memory access}} = 190$ ns

Calculate the estimated time needed per context switch

Determine Expectation Quantified

instructions	memory accesses	simple SW model of context switch:
10	1	save state P1
50	2	determine next runnable task
20	1	update scheduler administration
10	1	load state P2
10	1 +	run P2
100	6	

Estimate how many instructions and memory accesses are needed per context switch

input data HW:

$t_{\text{ARM instruction}} = 5$ ns

$t_{\text{memory access}} = 190$ ns

500 ns
1140 ns +
1640 ns

Calculate the estimated time needed per context switch

round up (as margin) gives expected $t_{\text{context switch}} = 2\ \mu s$

4C. Code to Measure Context Switch

Task 1

Time Stamp End
Cache Flush
Time Stamp Begin
Context Switch

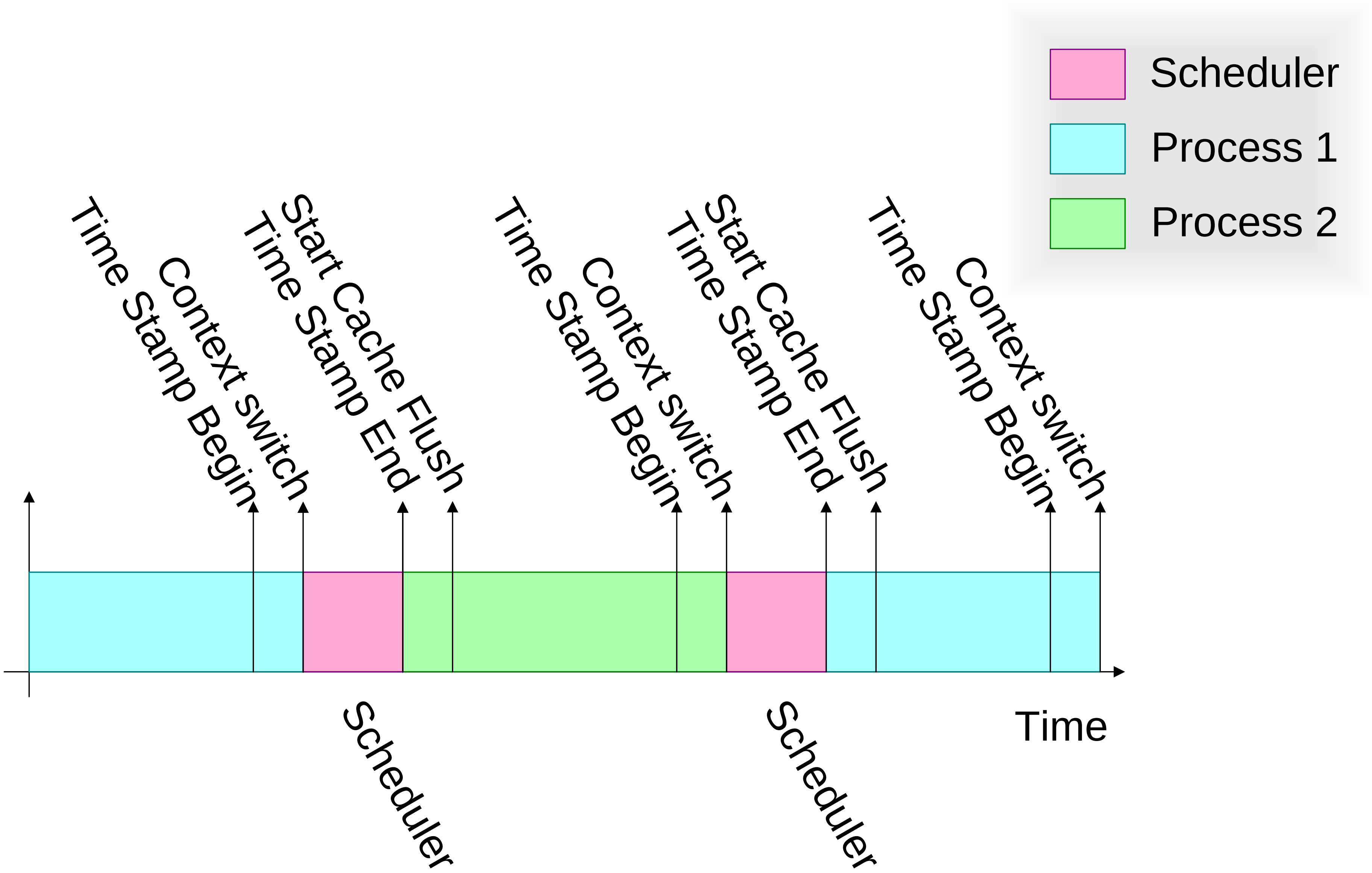
Time Stamp End
Cache Flush
Time Stamp Begin
Context Switch

Task 2

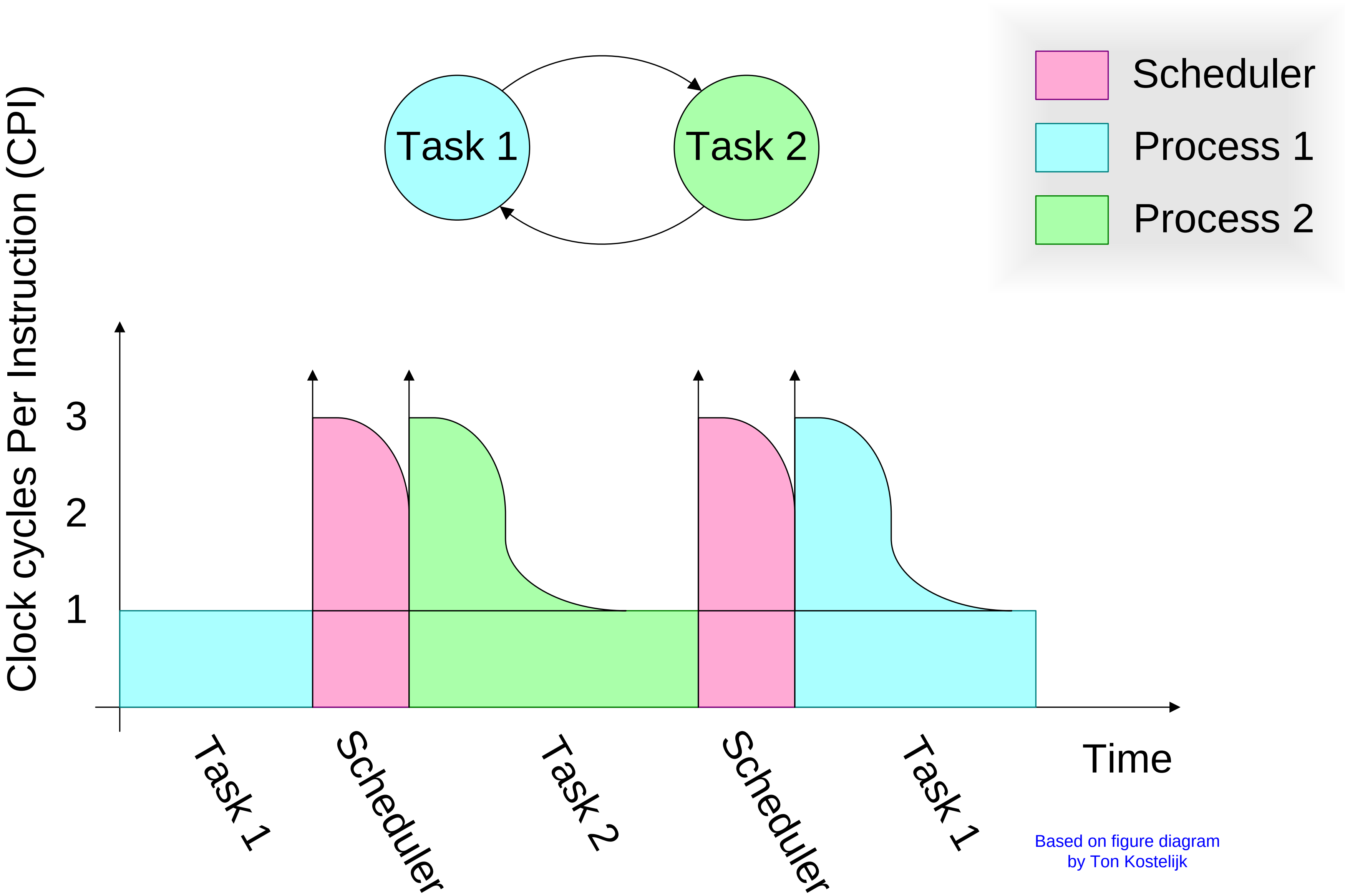
Time Stamp End
Cache Flush
Time Stamp Begin
Context Switch

Time Stamp End
Cache Flush
Time Stamp Begin
Context Switch

Measuring Task Switch Time

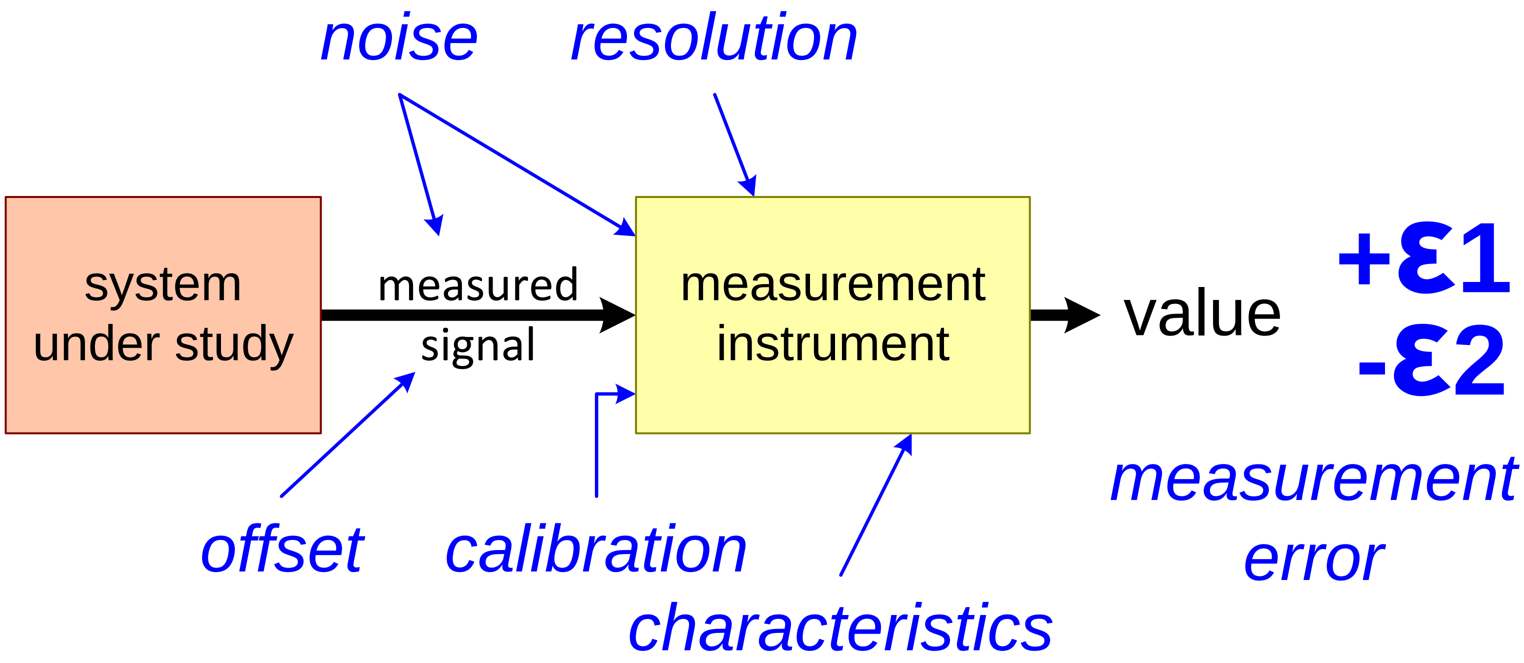


Understanding: Impact of Context Switch

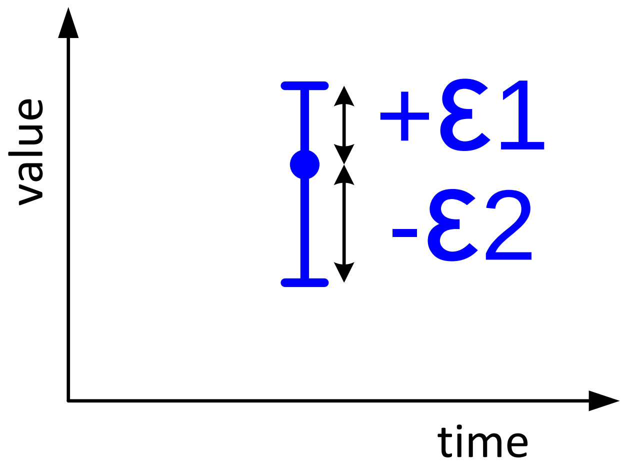


Based on figure diagram
by Ton Kostelijk

5. Accuracy: Measurement Error



measurements *have*
stochastic variations *and*
systematic deviations
resulting in a *range*
rather than a *single value*



Accuracy 2: Be Aware of Error Propagation

$$t_{duration} = t_{end} - t_{start}$$

$$t_{start} = 10 \text{ +/- } 2\ \mu s$$

$$t_{end} = 14 \text{ +/- } 2\ \mu s$$

$$t_{duration} = 4 \text{ +/- } ?\ \mu s$$

systematic errors: add linear

stochastic errors: add quadratic

Intermezzo Modeling Accuracy

Measurements have

stochastic variations and systematic deviations

resulting in a range rather than a single value.

The inputs of modeling,

"facts", assumptions, and measurement results,

also have stochastic variations and systematic deviations.

Stochastic variations and systematic deviations

propagate (add, amplify or cancel) through the model

resulting in an output range.

ARM9 200 MHz $t_{context\ switch}$ as function of cache use

cache setting	$t_{context\ switch}$
From cache	2 µs
After cache flush	10 µs
Cache disabled	50 µs

7. Expectation versus Measurement

	expected	$t_{context\ switch} = 2\ \mu s$
	measured	$t_{context\ switch} = 10\ \mu s$

How to explain?

simple SW model of context switch:

instructions	memory accesses	
10	1	save state P1
50	2	determine next runnable task
20	1	update scheduler administration
10	1	load state P2
10	1 +	run P2
100	6	

input data HW:

$t_{ARM\ instruction} = 5$ ns

$t_{memory\ access} = 190$ ns

500 ns

1140 ns +

1640 ns

potentially missing in expectation:

- memory accesses due to instructions
  - ~10 instruction memory accesses ~= 2 µs
- memory management (MMU context)
- complex process model (parents, permissions)
- bookkeeping, e.g performance data
- layering (function calls, stack handling)
- the combination of above issues

However, measurement seems to make sense

Conclusion Context Switch Overhead

$$t_{overhead} = n_{context\ switch} * t_{context\ switch}$$

$n_{context\ switch}$ (s^{-1})	$t_{context\ switch} = 10\mu s$		$t_{context\ switch} = 2\mu s$	
	$t_{overhead}$	CPU load overhead	$t_{overhead}$	CPU load overhead
500	5ms	0.5%	1ms	0.1%
5000	50ms	5%	10ms	1%
50000	500ms	50%	100ms	10%

Summary Context Switching on ARM9

goal of measurement

Guidance of concurrency design and task granularity

Estimation of context switching overhead

Cost of context switch on given platform

examples of measurement

Needed: context switch overhead ~10% accurate

Measurement instrumentation: HW pin and small SW test program

Simple models of HW and SW layers

Measurement results for context switching on ARM9

Summary Measuring Approach

Conclusions

Measurements are an important source of factual data.

A measurement requires a well-designed experiment.

Measurement error, validation of the result determine the credibility.

Lots of consolidated data must be reduced to essential understanding.

Techniques, Models, Heuristics of this module

experimentation

error analysis

estimating expectations

Colophon

This work is derived from the EXARCH course at CTT developed by *Ton Kostelijk* (Philips) and *Gerrit Muller*.

The Boderc project contributed to the measurement approach. Especially the work of

Peter van den Bosch (Océ),

Oana Florescu (TU/e),

and *Marcel Verhoef* (Chess)

has been valuable.

ASP Python Exercise

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

Abstract

A simple measurement exercise is described. Purpose of this exercise is to build up experience in measuring and its many pitfalls. The programming language Python is used as platform, because of its availability and low threshold for use.

July 3, 2023
status: preliminary draft
version: 0

logo TBD

Case for Exercise

Select a programming environment,

where loop overhead and file open

can be measured in 30 minutes.

If this environment is not available,

then use Python.

Python download and information

Active State Python (Freeware distribution, runs directly)

http://www.activestate.com/Products/ActivePython/

Python Language Website

http://www.python.org/

Python Reference Card

http://admin.oreillynet.com/python/excerpt/PythonPocketRef/examples/python.pdf

Python example

```python
import time

for n in (1,10,100,1000,10000,100000,1000000):
    a = 0
    tstart = time.time()
    for i in xrange(n):
        a = a+1
    tend=time.time()

    print n, tend-tstart, (tend-tstart)/n

def example_filehandling():
    f = open("c:\\temp\\test.txt")
    for line in f.readlines():
        print line
    f.close()

tstart = time.time()
example_filehandling()
tend=time.time()
print "file open, read & print, close: ",tend-tstart,"s"
```

```
>>>
1 0.0 0.0
10 0.0 0.0
100 0.0 0.0
1000 0.0 0.0
10000 0.00999999046326 9.99999046326e-007
100000 0.039999961853 3.9999961853e-007
1000000 0.44000005722 4.4000005722e-007
test line 1

line 2

line 3

file open, read, close:  0.039999961853 s
```


Exercise

- Perform the following measurements
  1. loop overhead
  2. file open
- Determine for every measurement:

  What is the expected result?

  What is the measurement error?

  What is the result?

  What is the credibility of the result?

  Explain the result.

  (optional) What is the variation? Explain the variation.

Reflection on Exercise

+ measuring is easy
+ measuring provides data and understanding

~ result and expectation often don't match

- sensible measuring is more difficult

Module Modeling and Analysis: System model

by *Gerrit Muller* HSN-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module addresses Modeling and Analysis Performance. What are the customer performance needs, what are the operational performance considerations? What are the performance related design choices? How to analyze feasibility, explore design options, and how to validate performance?

July 3, 2023
status: preliminary draft
version: 0.4

Module Content

goal of this module

provide a stepwise approach to system modeling

provide concrete examples of system models

content of this module

web shop system model

Non Functional requirements (NFR), System Properties and Critical Technologies

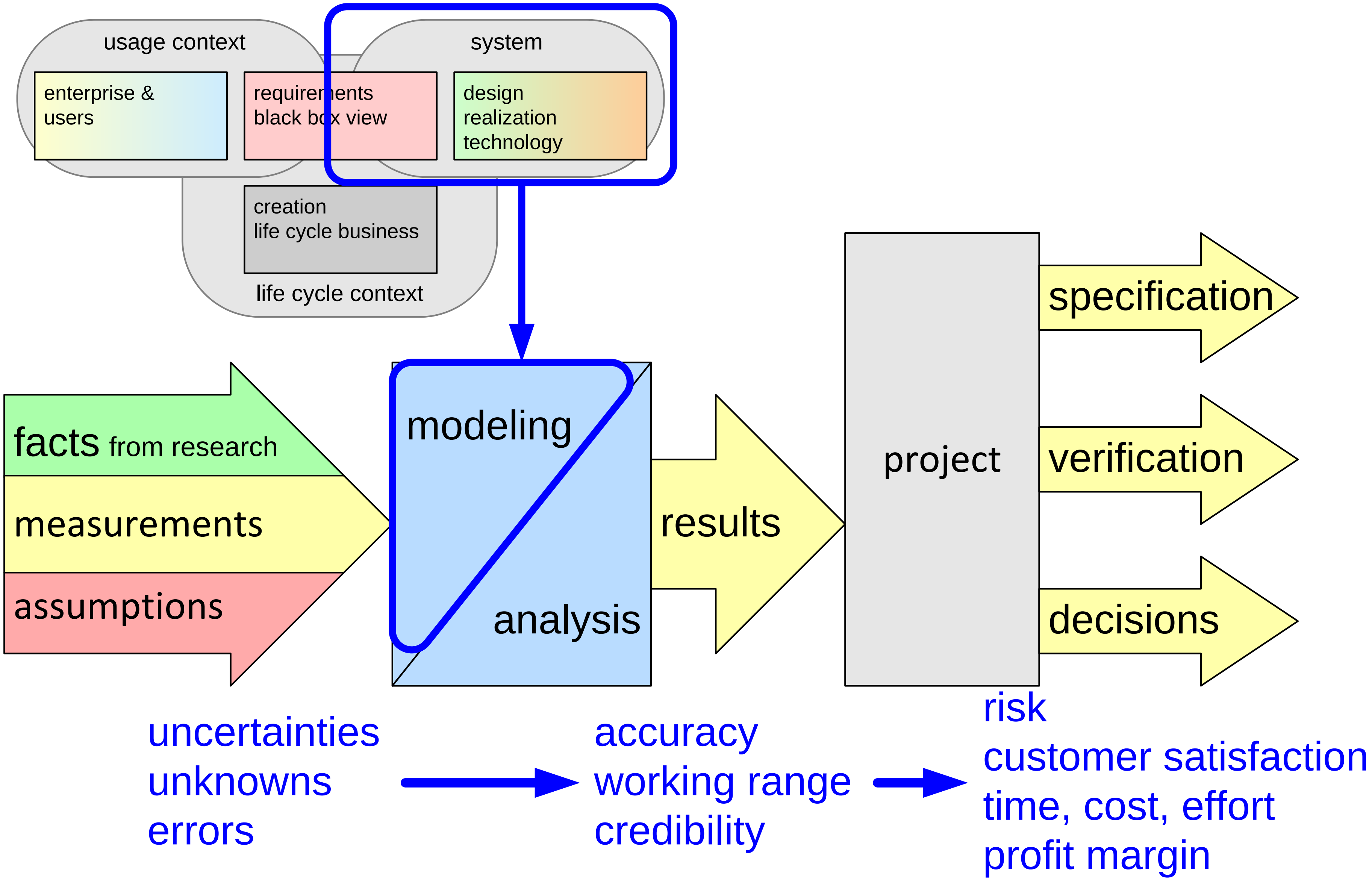
zero order and first order load models

budgeting

exercise

model one NFR in relation to a critical technology choice

Where are we in the Course?



Modeling and Analysis: System Model

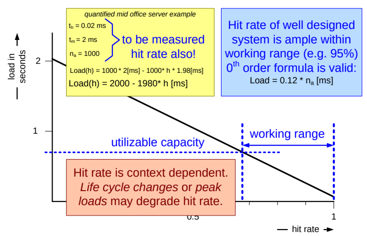
by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

This presentation uses a web shop service as example system to construct a system model. The caching of pictures of the products in the shop is modeled to analyze performance, robustness, scalability and reliability of the system.

Distribution

July 3, 2023
status: preliminary draft
version: 0.4



Content of Modeling and Analysis System Model

content

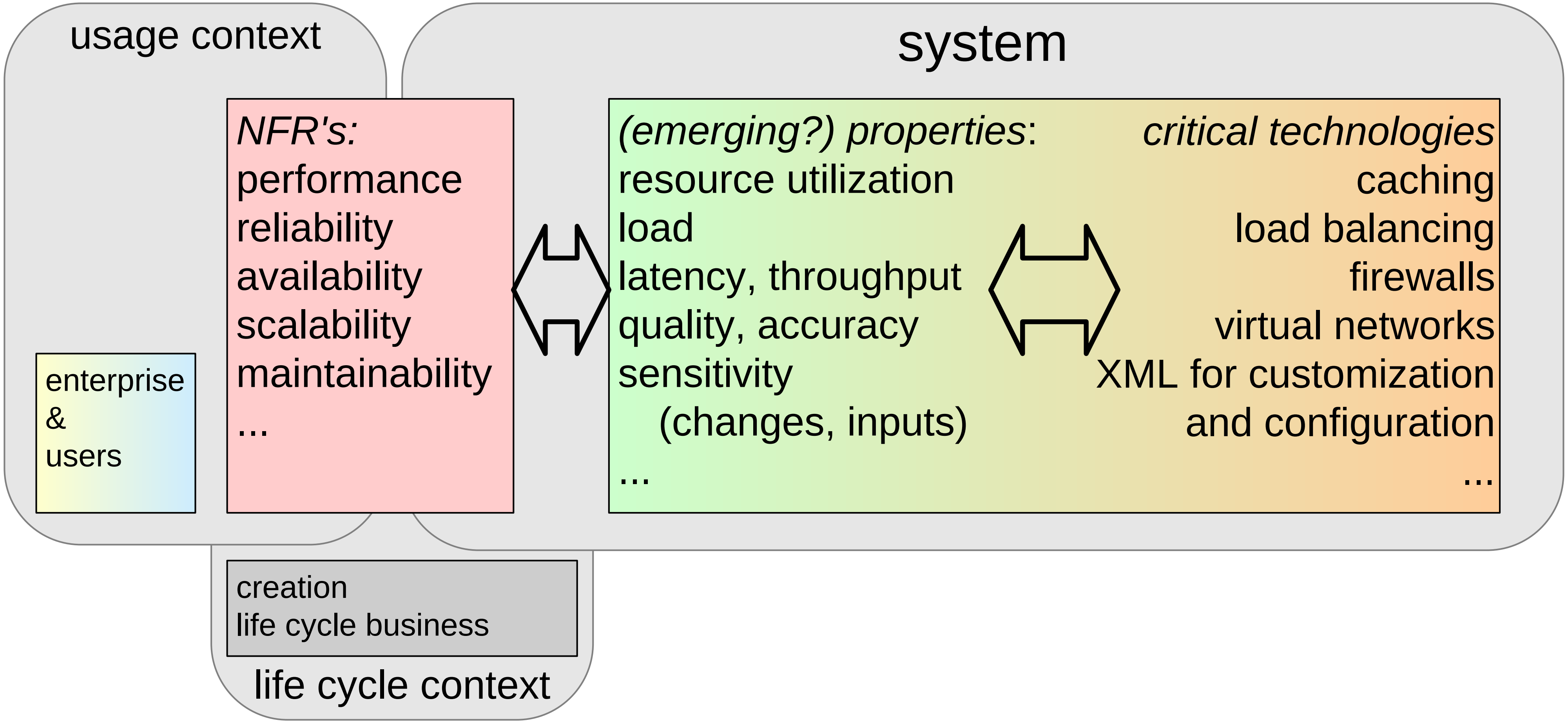
What to model of the system

Stepwise approach to system modeling

Non Functional requirements (NFR), System Properties and Critical Technologies

Examples of web shop case

What to Model in System Context?



Approach to System Modeling

1. determine relevant Non Functional Requirements (NFR's)
2. determine relevant system design properties
3. determine critical technologies
4. relate NFR's to properties to critical technologies
5. rank the relations in relevancy and criticality
6. model relations with a high score

Web Shop: NFR's, Properties and Critical Technologies

system

1

NFR's:

- performance browsing
- initial cost
- running costs
- reliability/availability
- scalability order rate
- maintainability
  - effort product changes
  - effort staff changes
- security

2

(emerging?) properties:

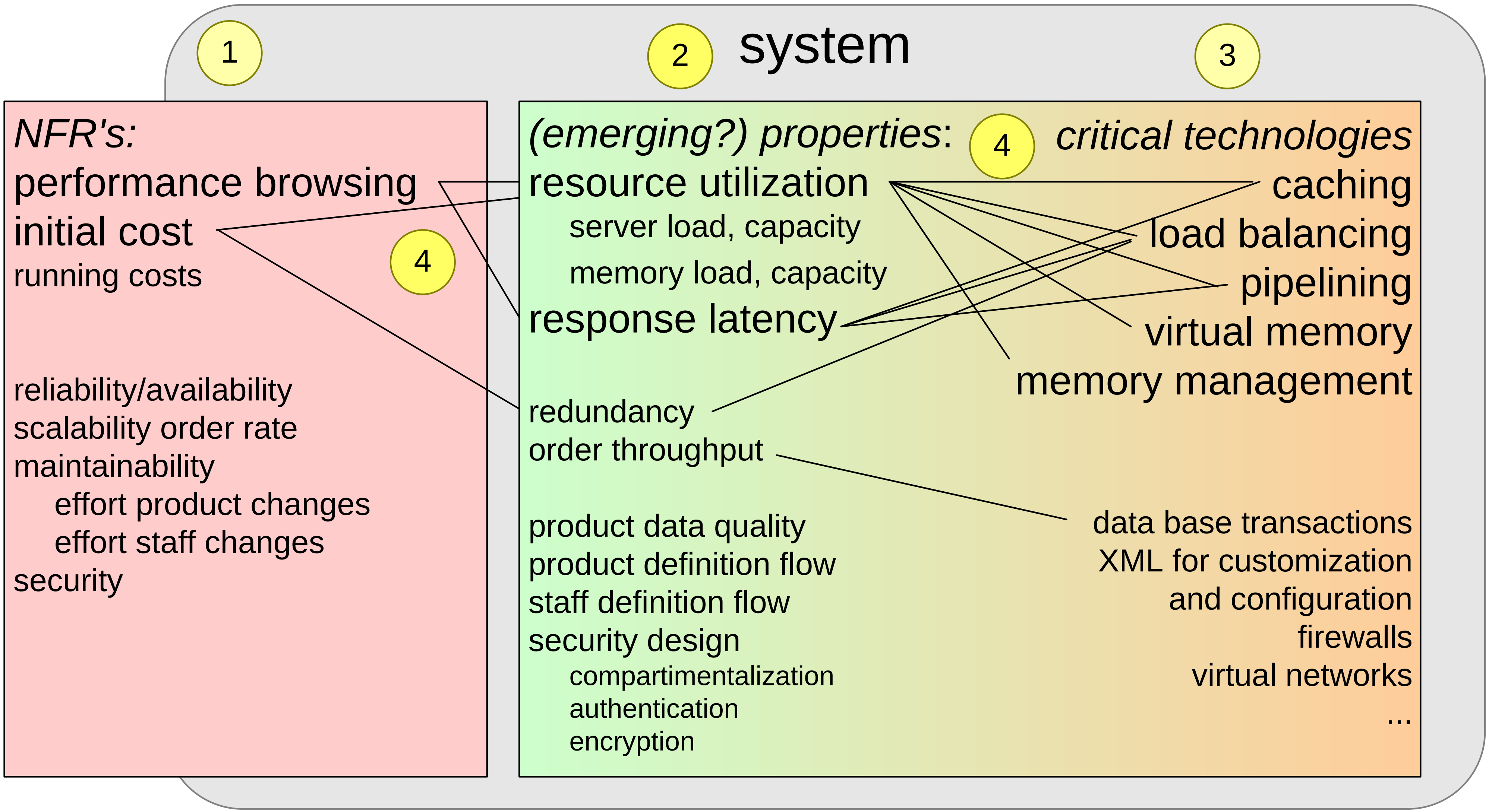
- resource utilization
  - server load, capacity
  - memory load, capacity
- response latency
- redundancy
- order throughput
- product data quality
- product definition flow
- staff definition flow
- security design
  - compartimentalization
  - authentication
  - encryption

3

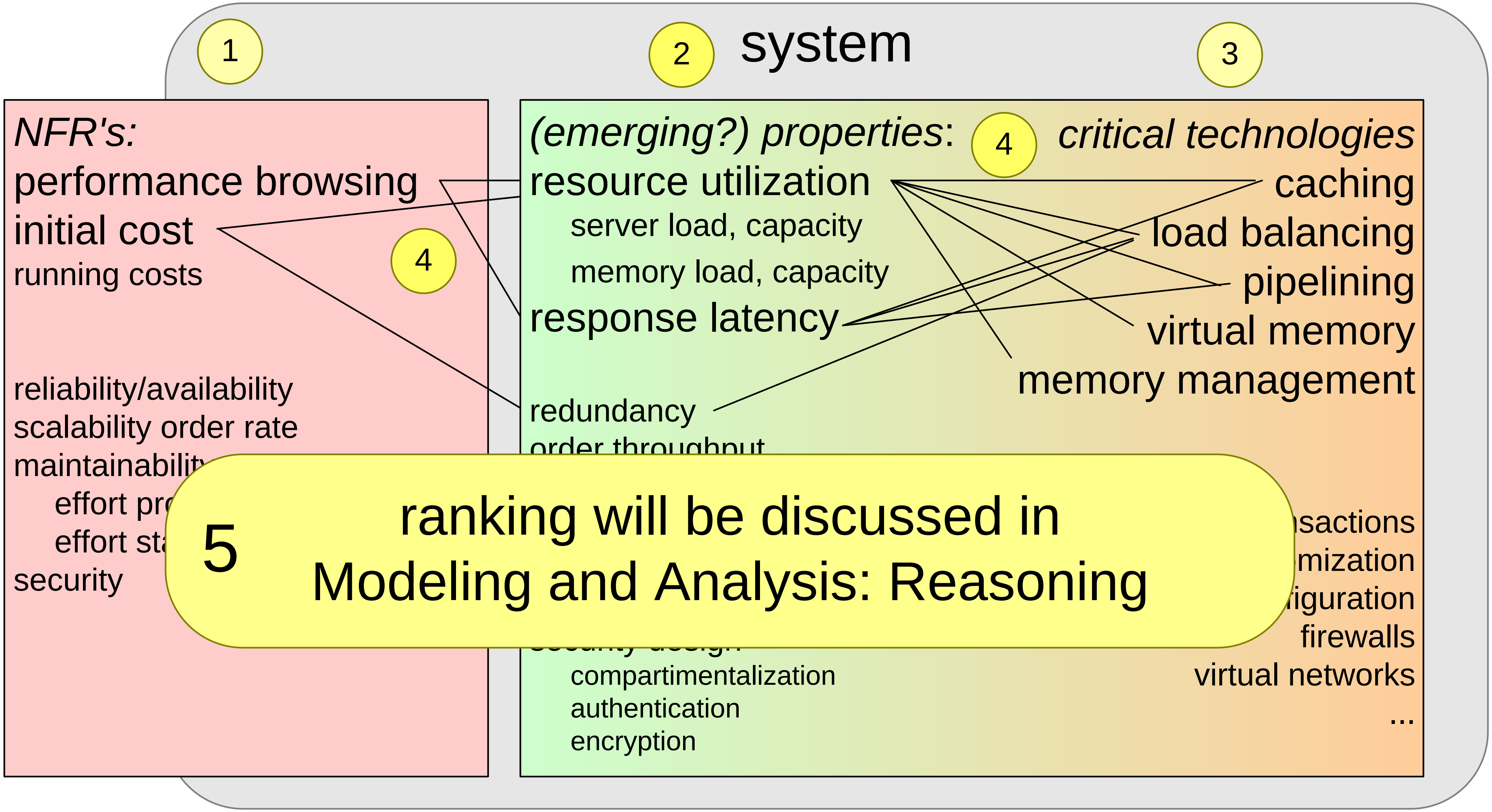
critical technologies

- caching
- load balancing
- pipelining
- virtual memory
- memory management
- data base transactions
- XML for customization and configuration
- firewalls
- virtual networks
- ...

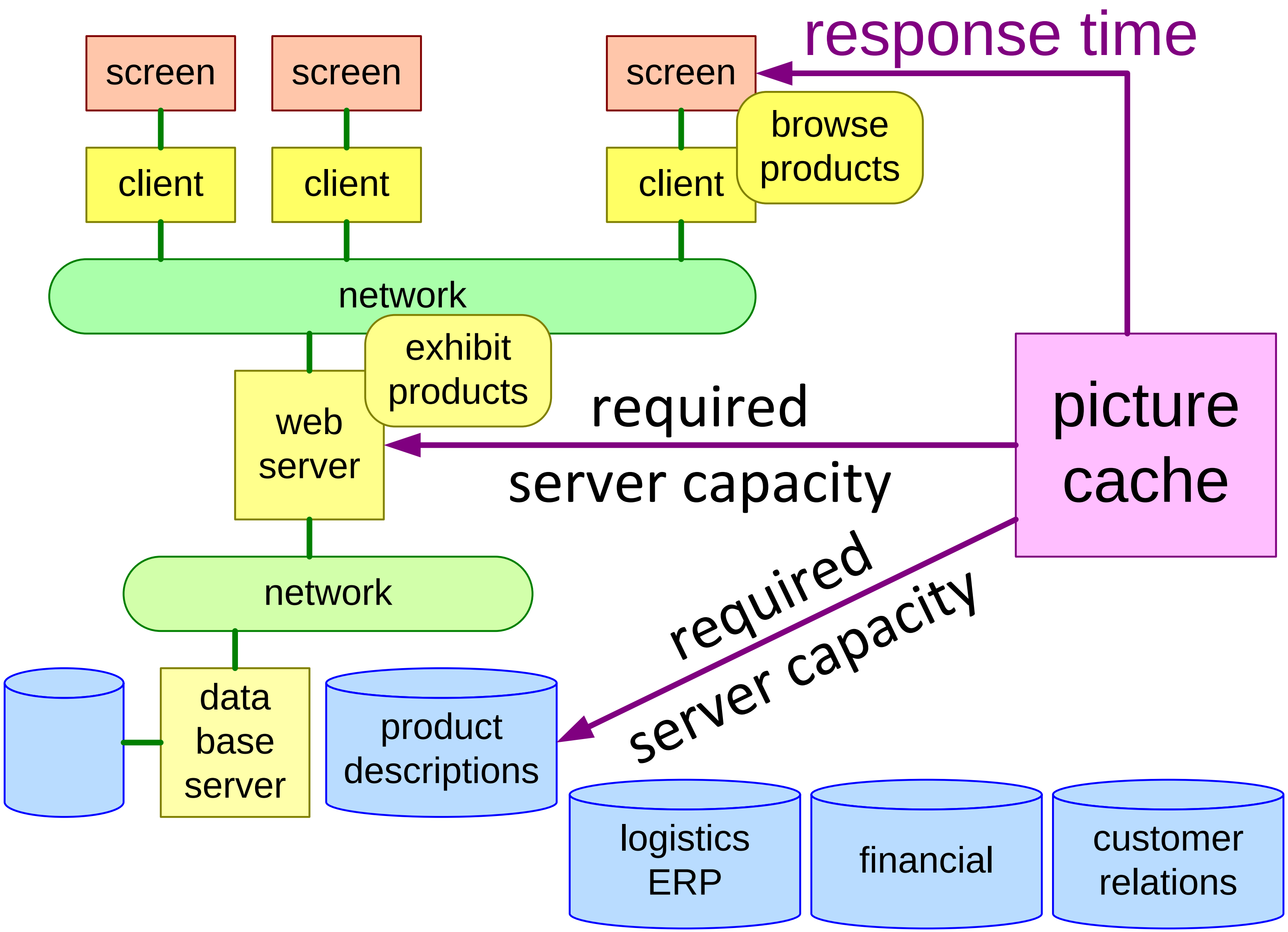
4. Determine Relations



5. Rank Relations



Purpose of Picture Cache Model in Web Shop Context



Zero Order Load Model

zero order web server load model

$$Load = n_a * t_a$$

n_a = total requests

t_a = cost per request

First Order Load Model

first order web server load model

$$Load = n_{a,h} * t_h + n_{a,m} * t_m$$

$n_{a,h}$ = accesses with cache hit

$n_{a,m}$ = accesses with cache miss

t_h = cost of cache hit

t_m = cost of cache miss

$$n_{a,h} = n_a * h$$

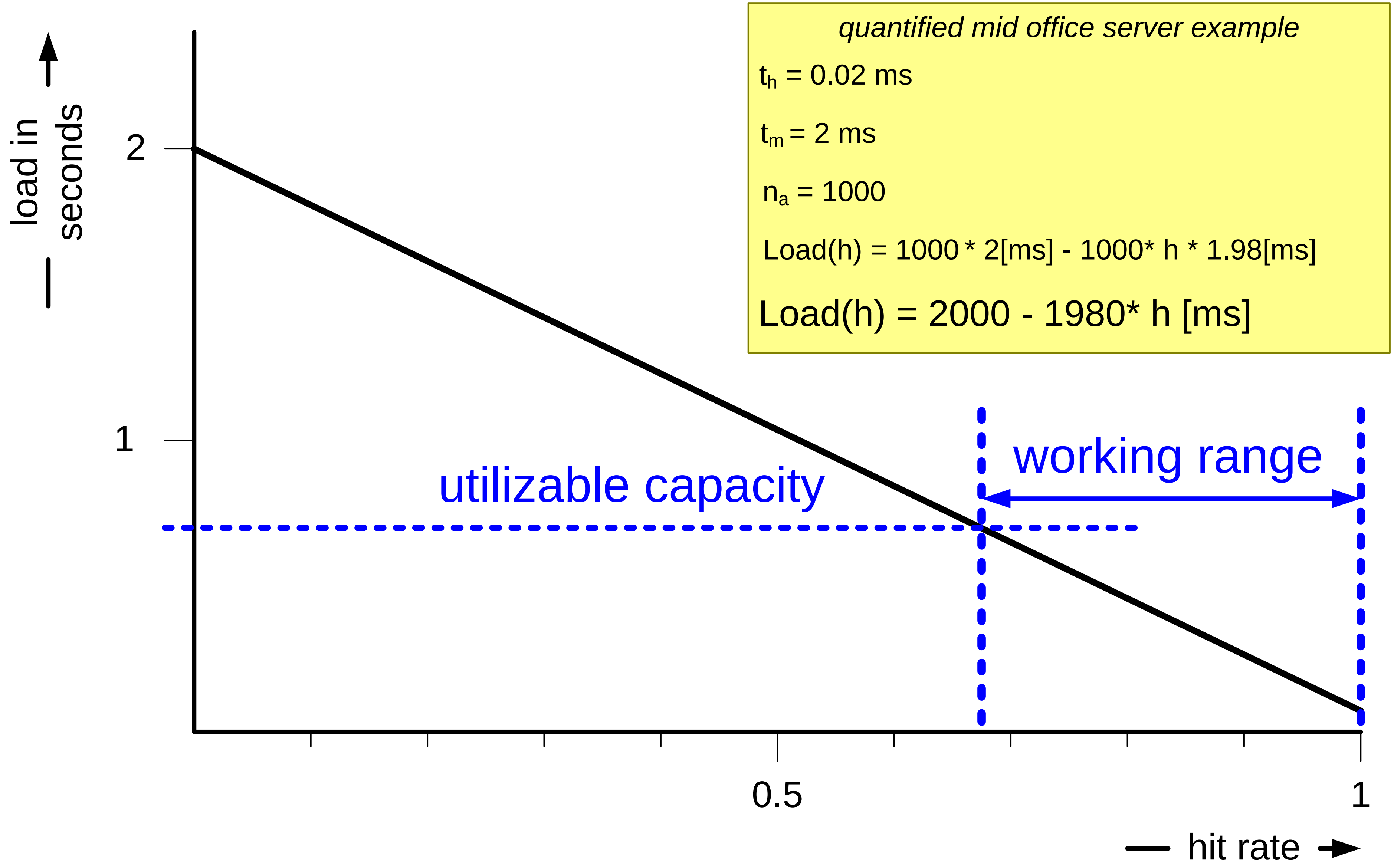
$$n_{a,m} = n_a * (1-h)$$

n_a = total accesses

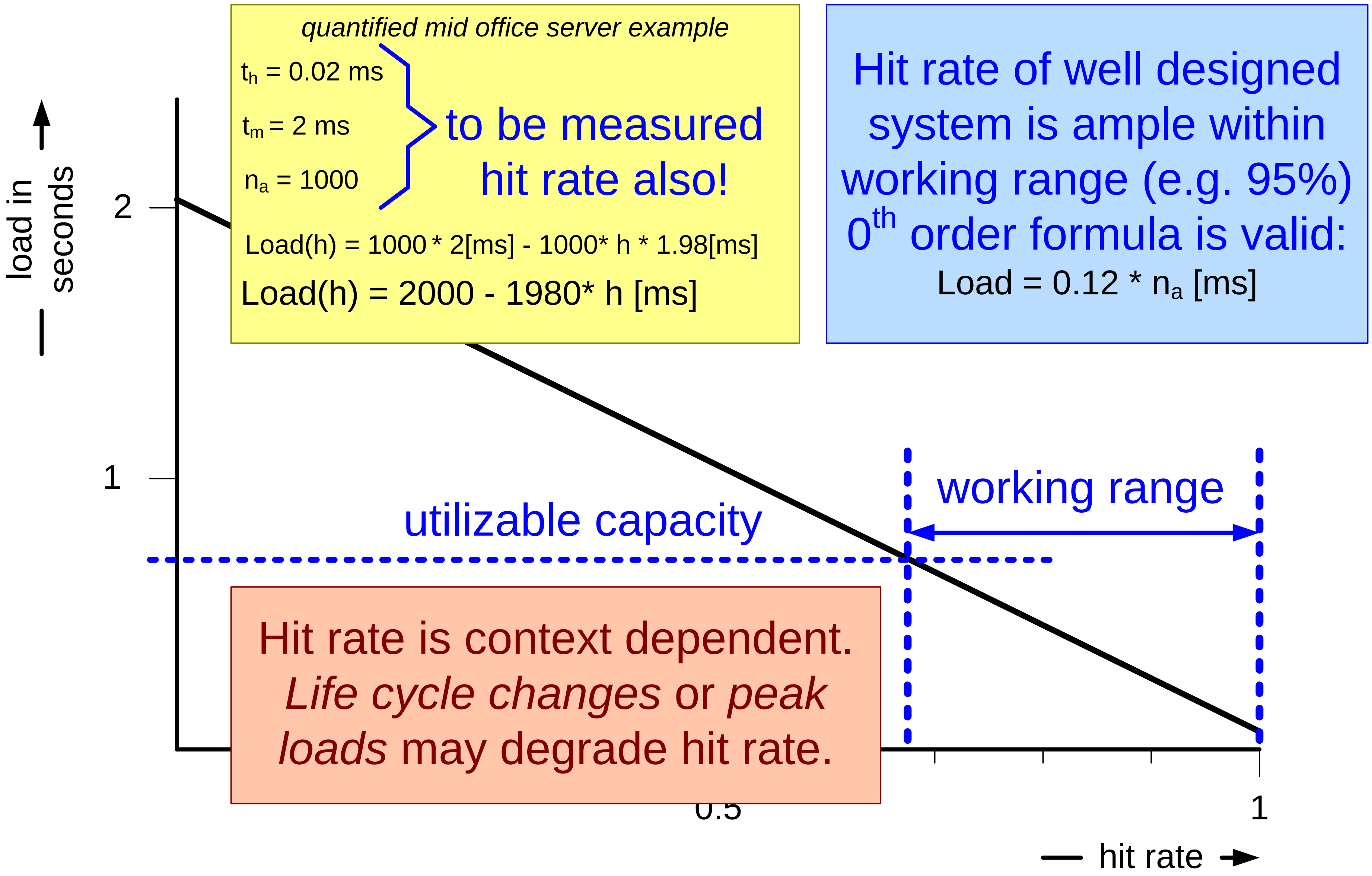
h = hit rate

$$Load(h) = n_a * h * t_h + n_a * (1-h) * t_m \quad = n_a * t_m - n_a * h * (t_m - t_h)$$

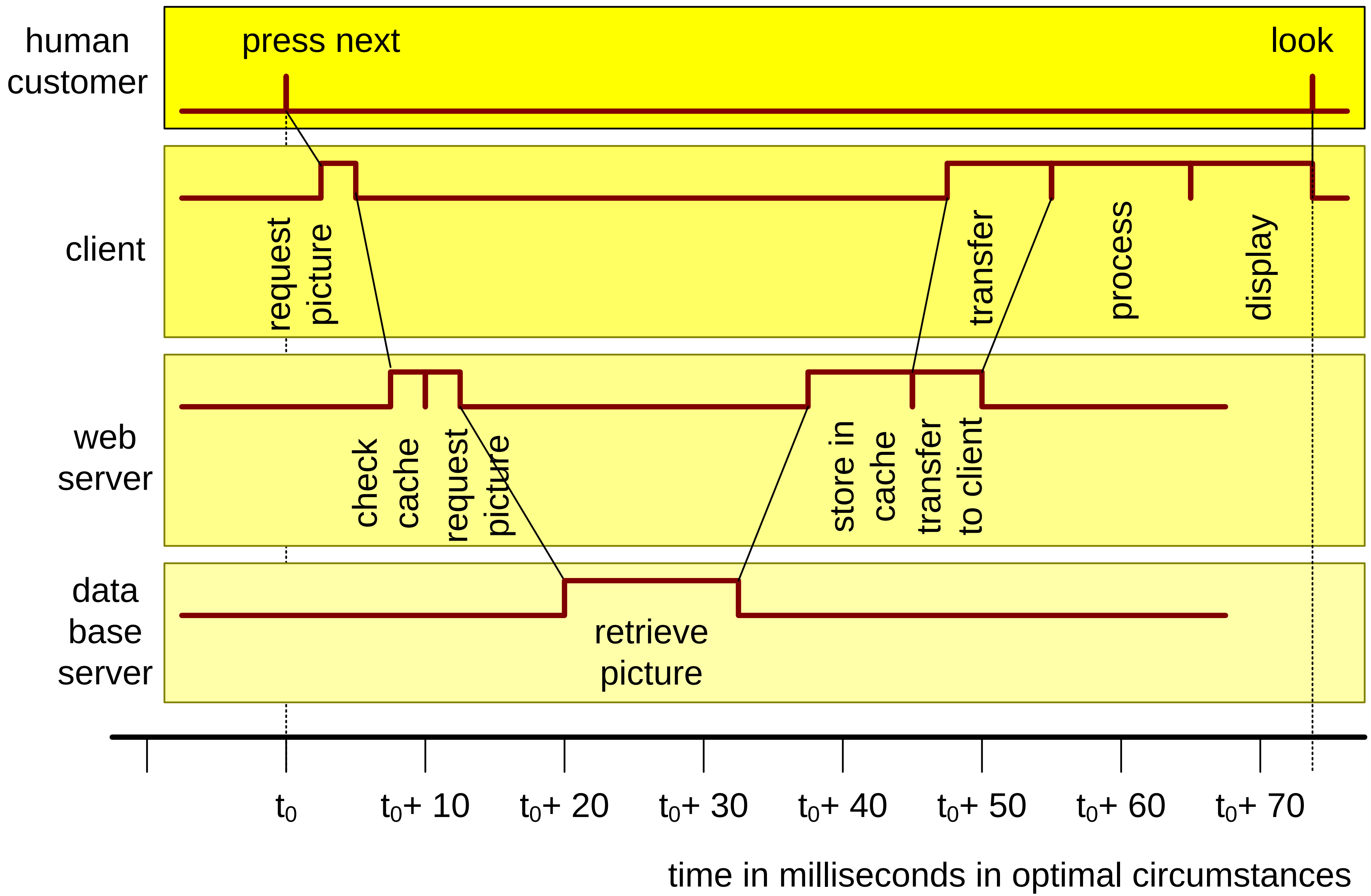
Quantification: From Formulas to Insight



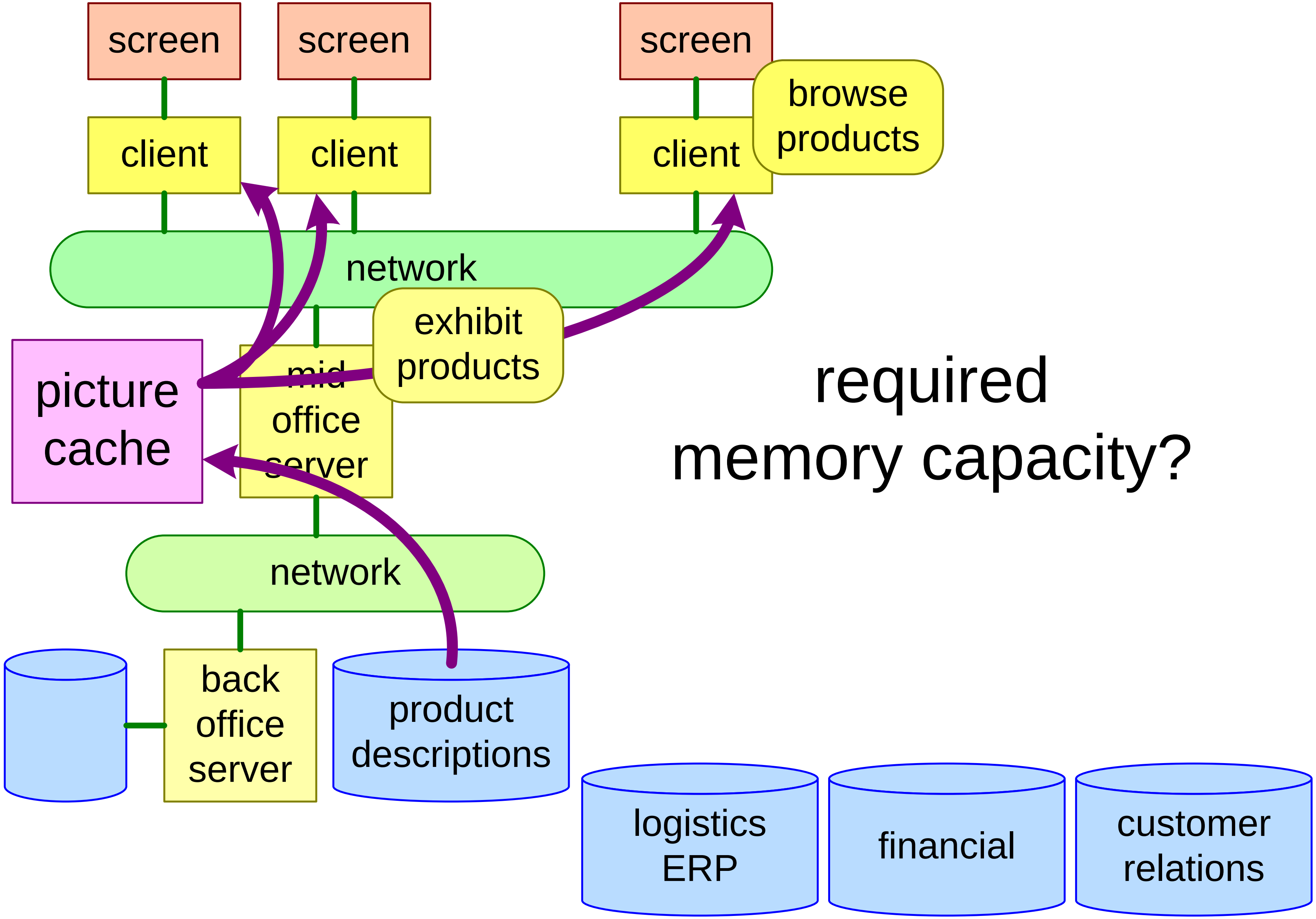
Hit Rate Considerations



Response Time

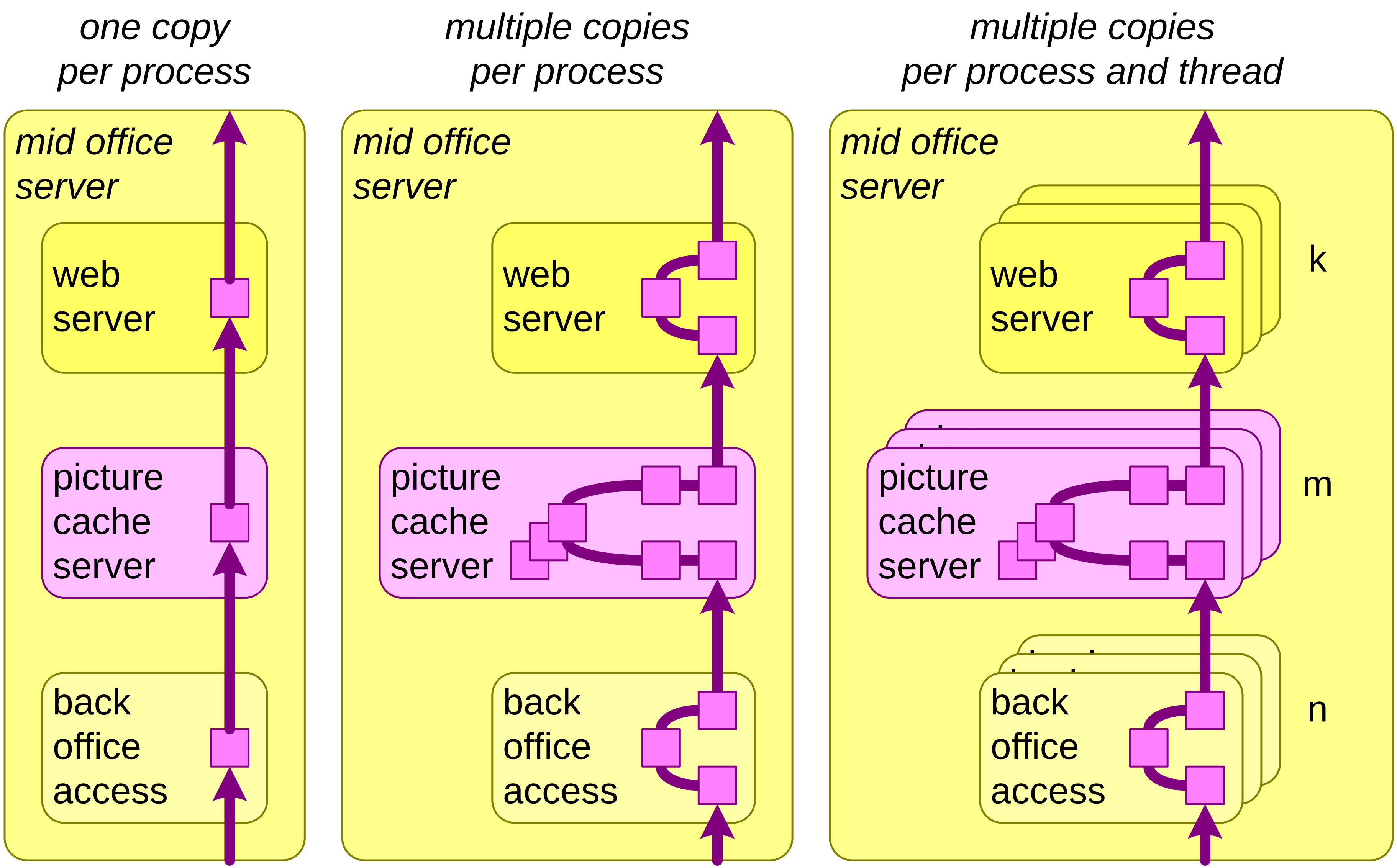


What Memory Capacity is Required for Picture Transfers?



required memory capacity?

Process View of Picture Flow in Web Server



Formula memory Use Web Server

picture memory =

3 * n * s +

5 * m * s + c * s +

3 * k * s

where

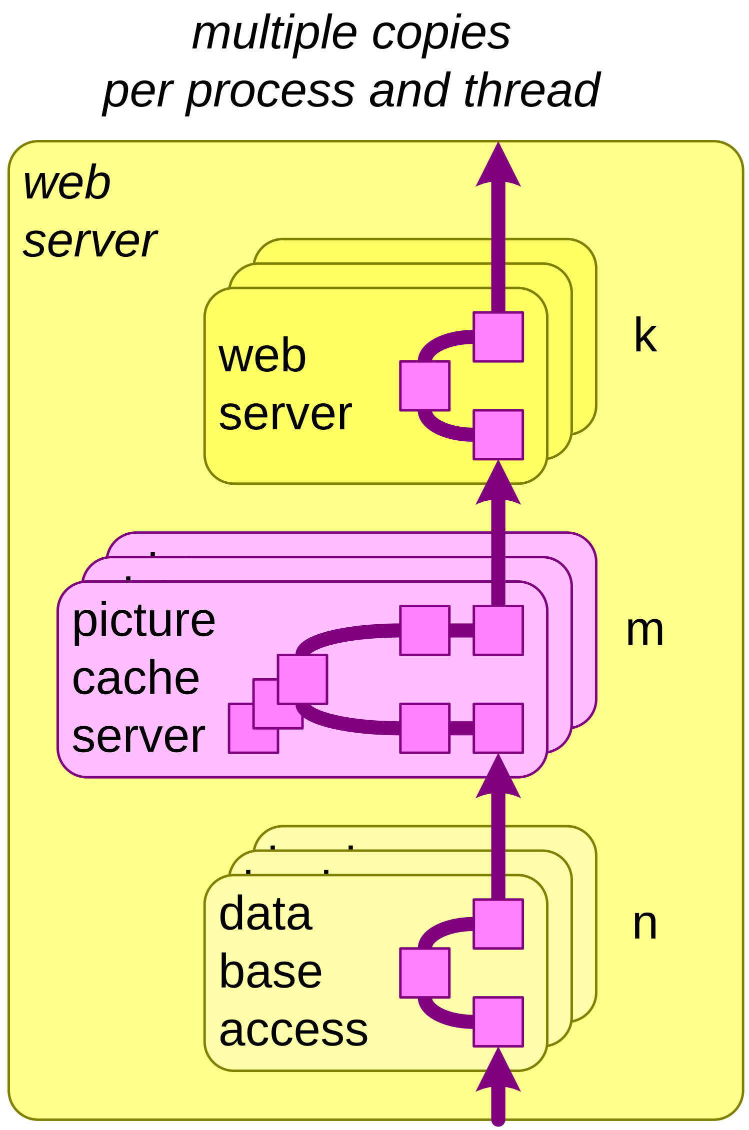
n = # data base access threads

m = # picture cache threads

k = # web server threads

s = picture size in bytes

c = in memory cache capacity in # pictures



Web Server Memory Capacity

use case	n	m	k	s	c	MB	storage type
small shop	1	1	1	100	10	1.5	L3
	2	4	10	100	20	5.3	main
highly concurrent	2	4	1000	100	100	296	main
large pictures	2	4	1000	1000	100	2,962	main+disk
many pictures	2	4	10	100	100,000	9,540	main+disk
all at once	2	4	1000	1000	100,000	98,234	disk

picture memory = 3 n s + 5 m s + c s +3 k s

where

n = # back office access threads

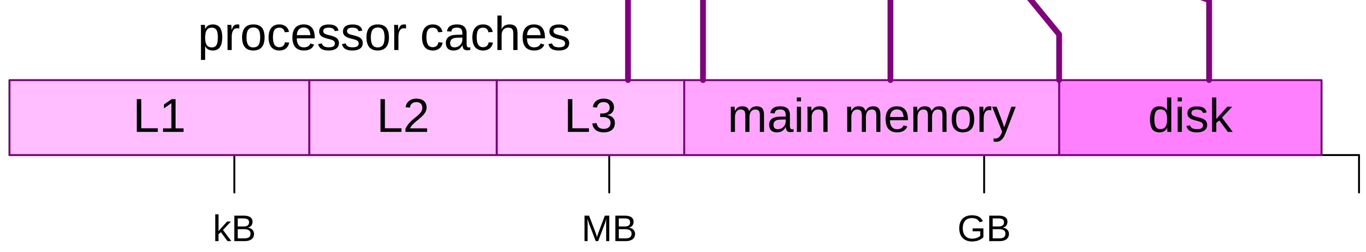
m = # picture cache threads

k = # web server threads

s = picture size in bytes

c = in memory cache capacity in # pictures

memory use
- product browsing only
- pictures only
- single server



What is the performance impact of memory use on other processing?

We Have only Modeled a Small Part of the System...

function	browse/exhibit products	sales, order intake, payments track, order handling stock handling financial bookkeeping customer relation management update catalogue advertize after sales support
data	picture	structured (product attributes, logistics, ...) program code
aspect	server memory use response time server load	network use reliability any resource, any NFR

$$\text{aspect result} = \sum_{d\,=\,\text{all data}} \; \sum_{f\,=\,\text{all functions}} \text{aspect}(d, f)$$

ignoring other dimensions such as applications, users, circumstances

... to Understand Some of the Systems Aspects

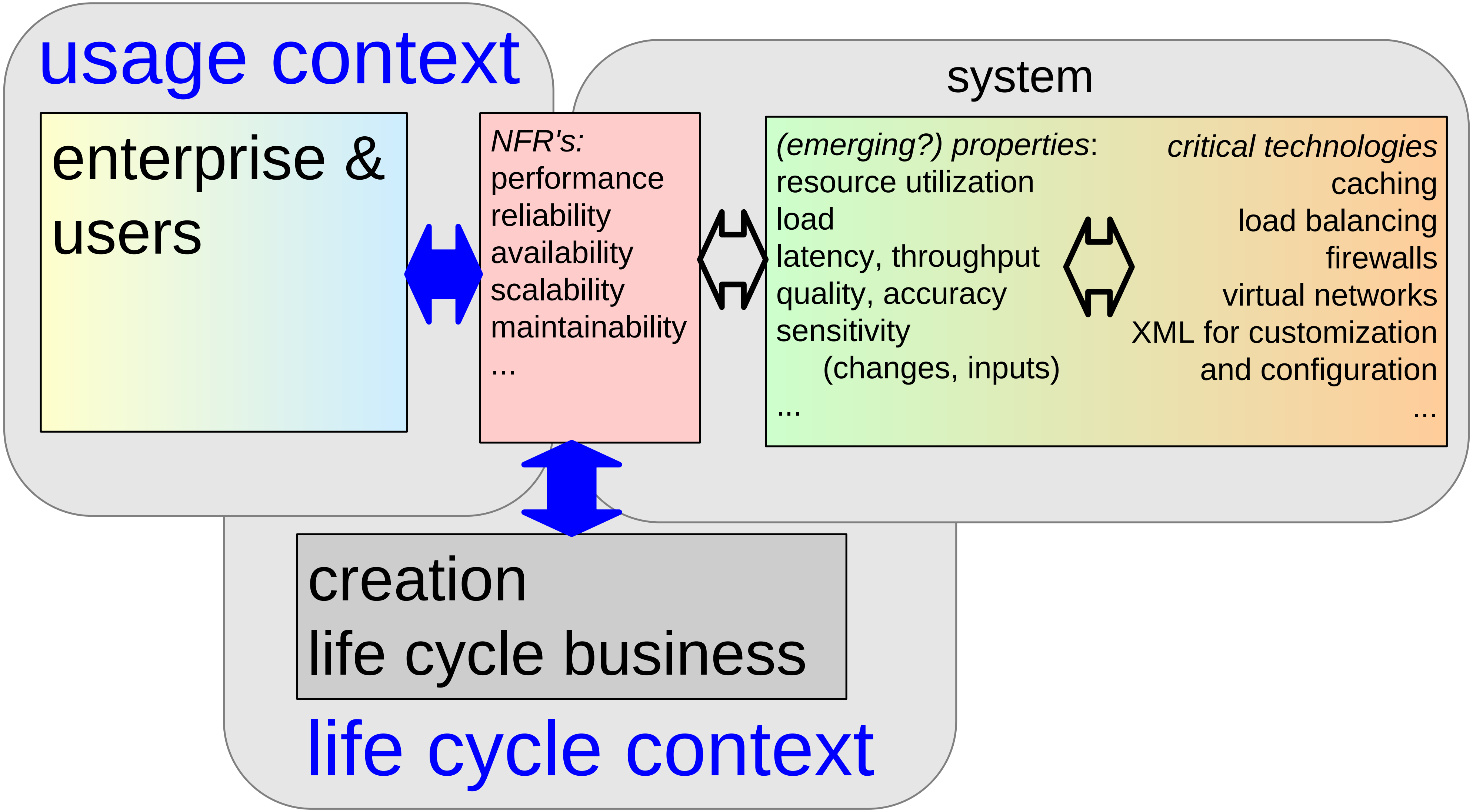
static

mostly assumptions and coarse estimates

some insight in:

- what are key design issues
- what are relevant use case areas

Refinement After Context Modeling



Summary

Conclusions
Non Functional Requirements are the starting point for system modeling
Focus on highest ranking relations between NFR's and critical technologies
Make simple mathematical models
Evaluate quantified instantiations

Techniques, Models, Heuristics of this module
Non functional requirements
System properties
Critical technologies
Graph of relations

Modeling and Analysis: Budgeting

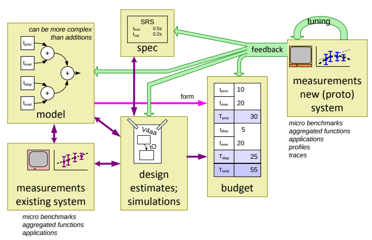
by *Gerrit Muller*   TNO-ESI, HSN-NISE
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

This presentation addresses the fundamentals of budgeting: What is a budget, how to create and use a budget, what types of budgets are there. What is the relation with modeling and measuring.

Distribution

July 3, 2023
status: preliminary draft
version: 1.0



Budgeting

content of this presentation

What and why of a budget

How to create a budget (decomposition, granularity, inputs)

How to use a budget

What is a Budget?

A ***budget*** *is*

a ***quantified instantation*** *of a* ***conceptual model***

A ***budget*** *can*

prescribe *or* ***describe*** *the* ***contributions***

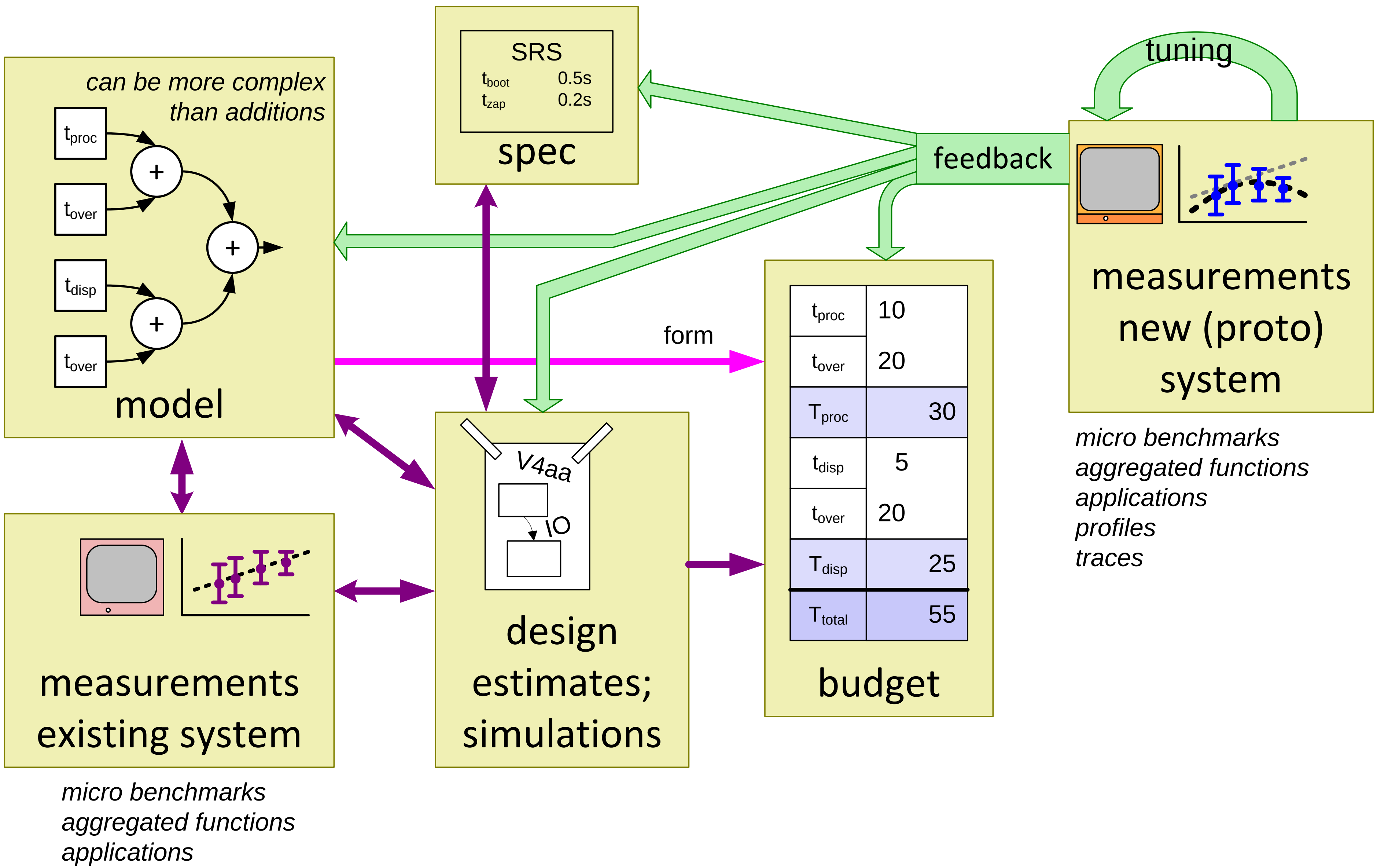
by ***parts*** *of the* ***solution***

to the ***system quality*** *under consideration*

Why Budgets?

- to make the design explicit
- to provide a baseline to take decisions
- to specify the requirements for the detailed designs
- to have guidance during integration
- to provide a baseline for verification
- to manage the design margins explicitly

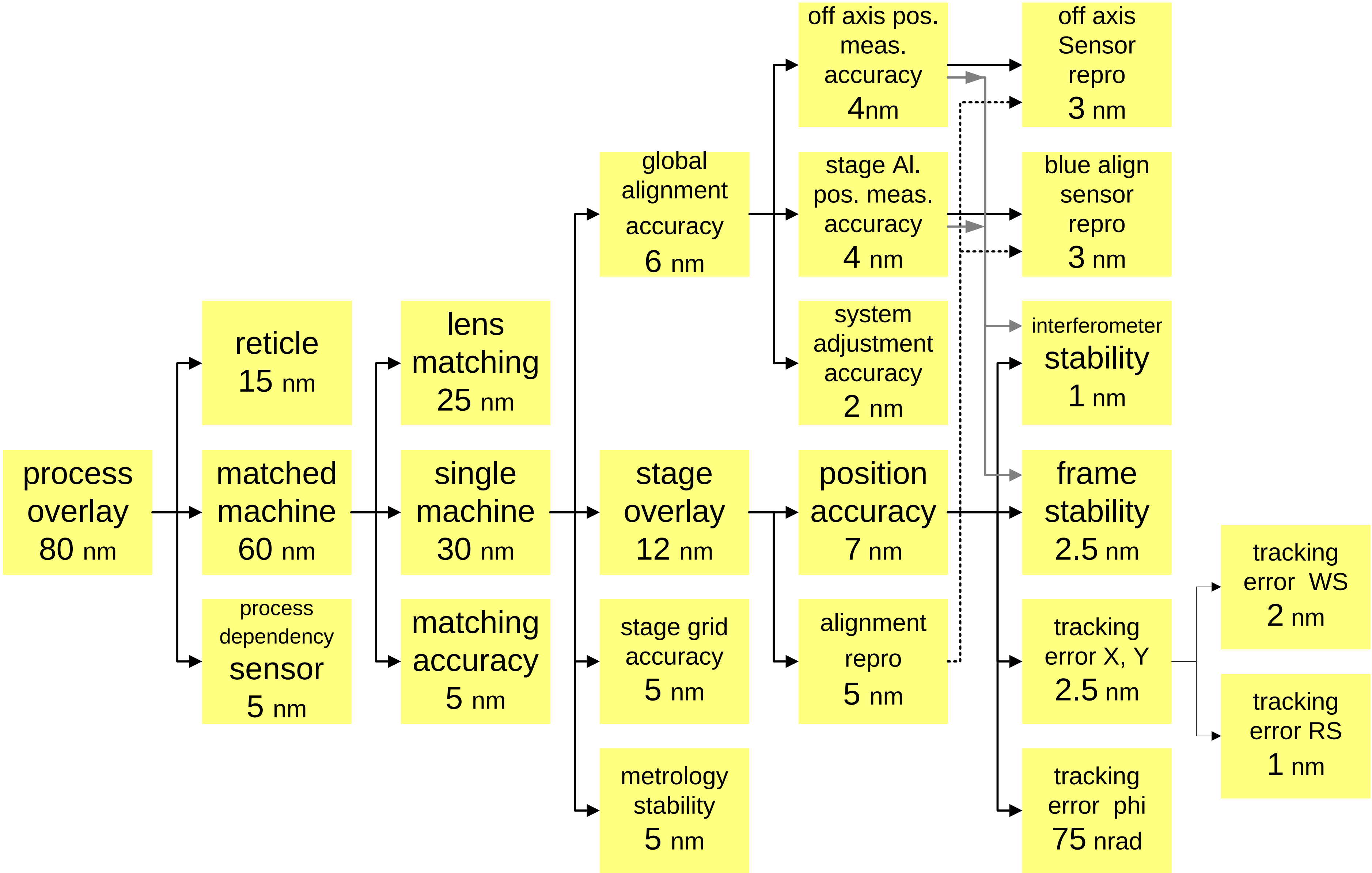
Visualization of Budget Based Design Flow



Stepwise Budget Based Design Flow

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	

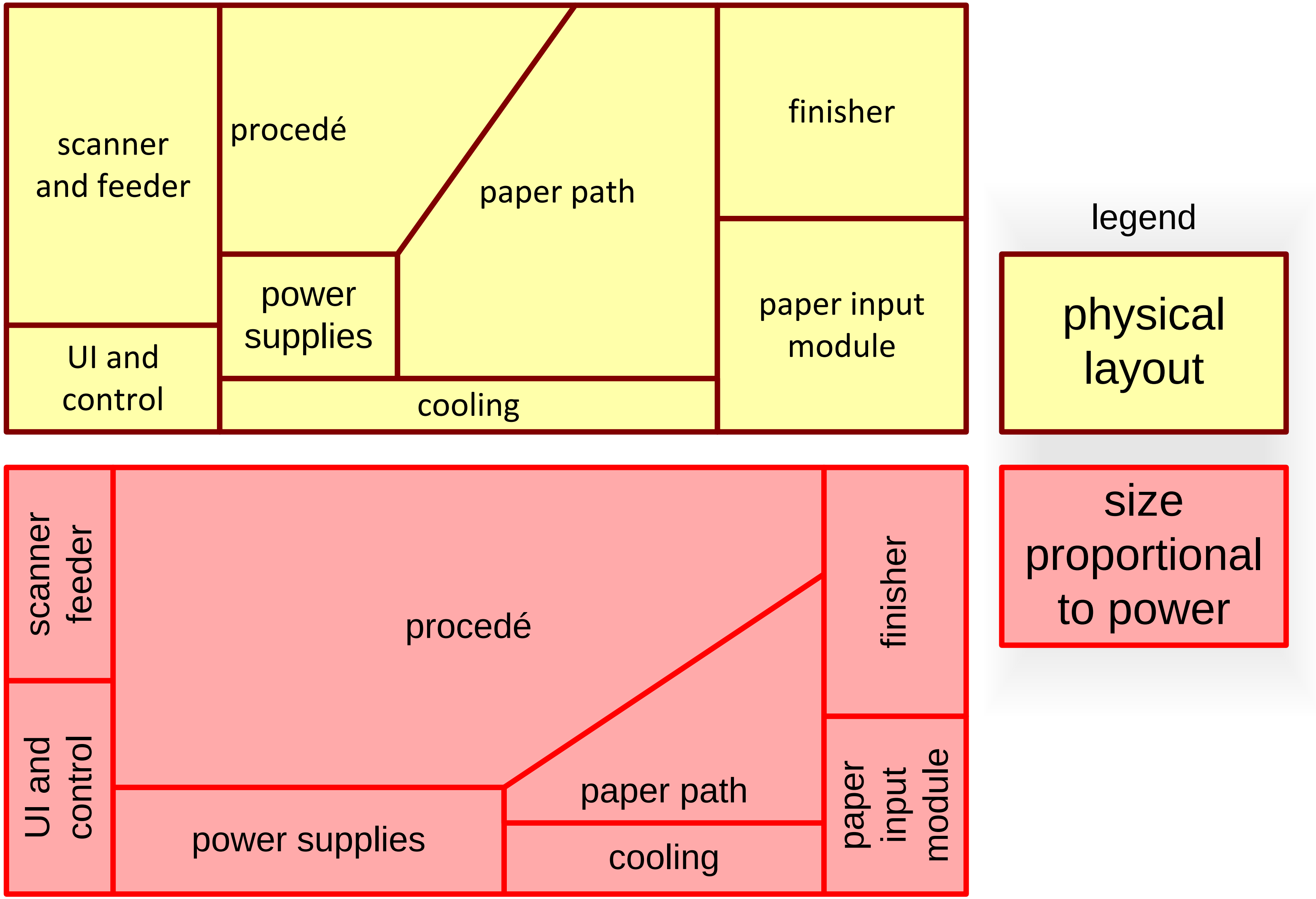
Budgets Applied on Waferstepper Overlay



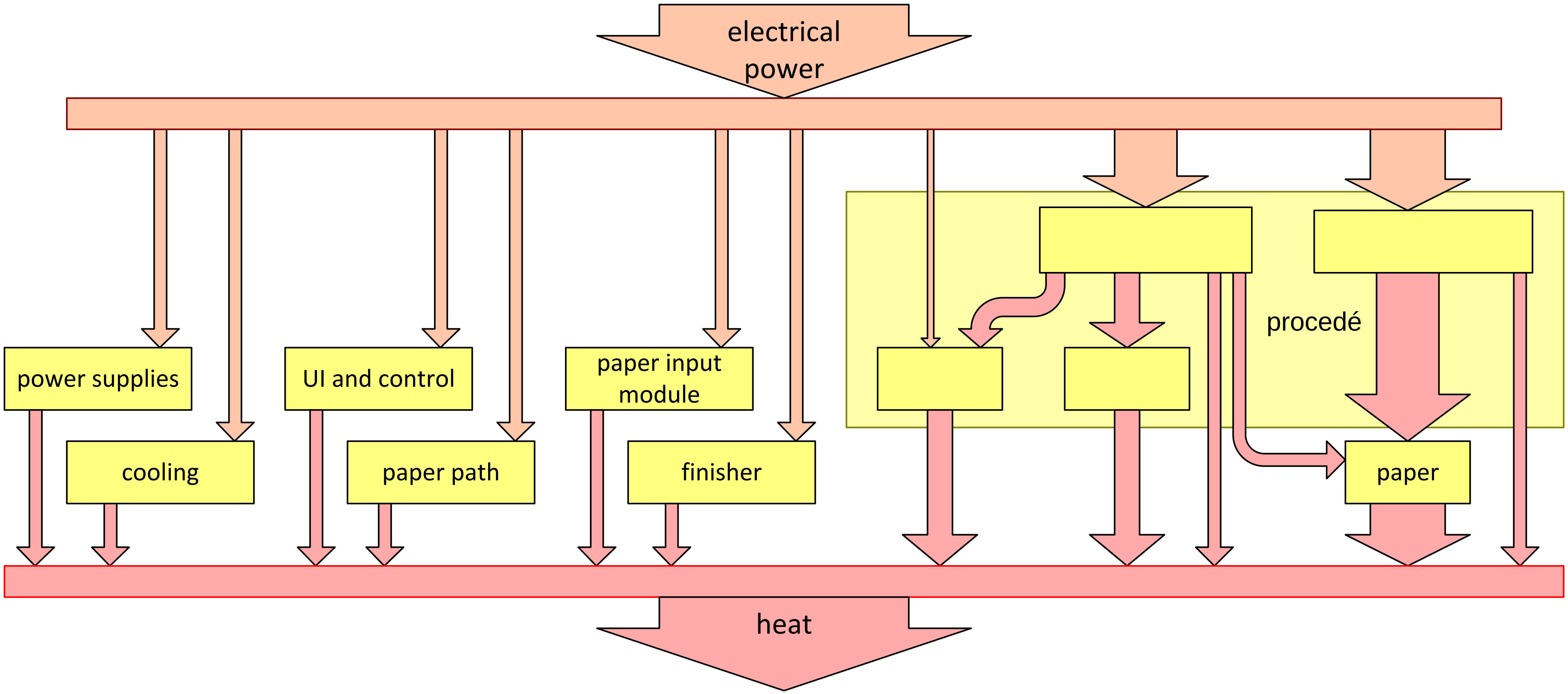
Budgets Applied on Medical Workstation Memory Use

memory budget in Mbytes	code	obj data	bulk data	total
shared code	11.0			11.0
User Interface process	0.3	3.0	12.0	15.3
database server	0.3	3.2	3.0	6.5
print server	0.3	1.2	9.0	10.5
optical storage server	0.3	2.0	1.0	3.3
communication server	0.3	2.0	4.0	6.3
UNIX commands	0.3	0.2	0	0.5
compute server	0.3	0.5	6.0	6.8
system monitor	0.3	0.5	0	0.8
application SW total	13.4	12.6	35.0	61.0
UNIX Solaris 2.x				10.0
file cache				3.0
total				74.0

Power Budget Visualization for Document Handler



Alternative Power Visualization



Evolution of Budget over Time

fact finding through details
aggregate to end-to-end performance
search for appropriate abstraction level(s)

from coarse guesstimate
to reliable prediction

from typical case
to boundaries of requirement space

from static understanding
to dynamic understanding

from steady state
to initialization, state change and shut down

from old system
to prototype
to actual implementation

time

start *later* *only if needed*

Potential Applications of Budget based design

- resource use (CPU, memory, disk, bus, network)
- timing (response, latency, start up, shutdown)
- productivity (throughput, reliability)
- Image Quality parameters (contrast, SNR, deformation, overlay, DOF)
- cost, space, time

What kind of budget is required?

static	dynamic
typical case	worst case
global	detailed
approximate	accurate

is the budget based on
wish, empirical data, extrapolation,
educated guess, or expectation?

Summary of Budgeting

A budget is a quantified instantiation of a model

A budget can prescribe or describe the contributions by parts of the solution to the system quality under consideration

A budget uses a decomposition in tens of elements

The numbers are based on historic data, user needs, first principles and measurements

Budgets are based on models and estimations

Budget visualization is critical for communication

Budgeting requires an incremental process

Many types of budgets can be made; start simple!

Colophon

The Boderc project contributed to Budget Based Design. Especially the work of

Hennie Freriks, Peter van den Bosch (Océ),

Heico Sandee and *Maurice Heemels* (TU/e, ESI)

has been valuable.

Module Modeling and Analysis: Application and Life Cycle Modeling

by *Gerrit Muller* HSN-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module addresses Modeling and Analysis Fundamentals of Application.

July 3, 2023
status: planned
version: 0.3

Module Content

goal of this module

Tangible understanding of the customer enterprise and life cycle aspects

Provide useful views on customer application

Simplify and demystify customer concerns

content of this module

Example financial computations

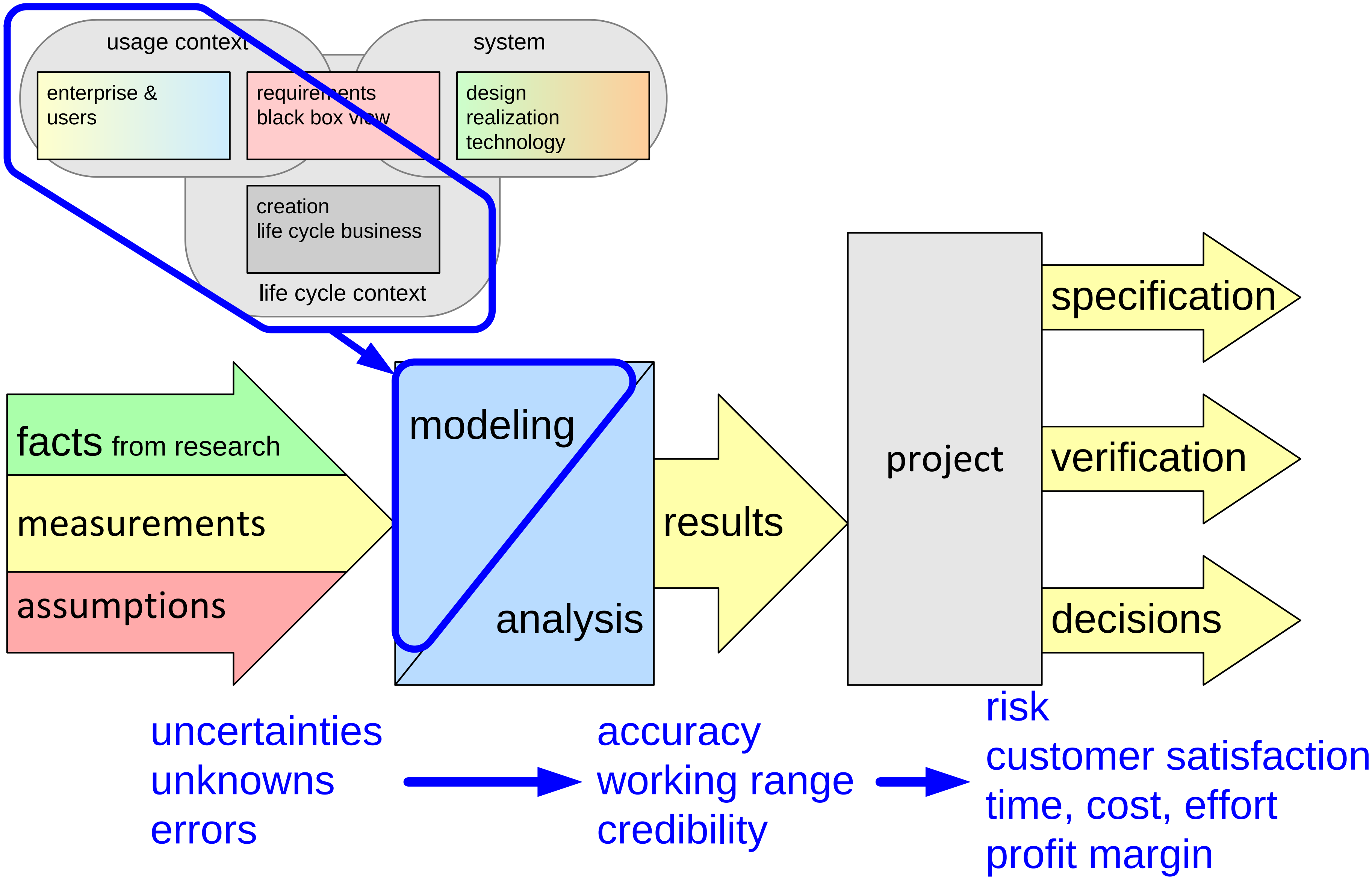
views on customer application:

- stakeholders and concerns
- simple cost models
- simple life cycle models

exercise

Make context and application models

Where are we in the Course?



Modeling and Analysis: Application Models

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

Abstract

The enterprise and its application is a complex system in itself. Specification and design decisions can have a significant impact on this system. We show a number of relevant application models with the purpose to be able to reason about specification and design in relation to the impact on the enterprise.

Distribution

July 3, 2023
status: planned
version: 0.1

logo
TBD

Understanding Usage and Life Cycle Context

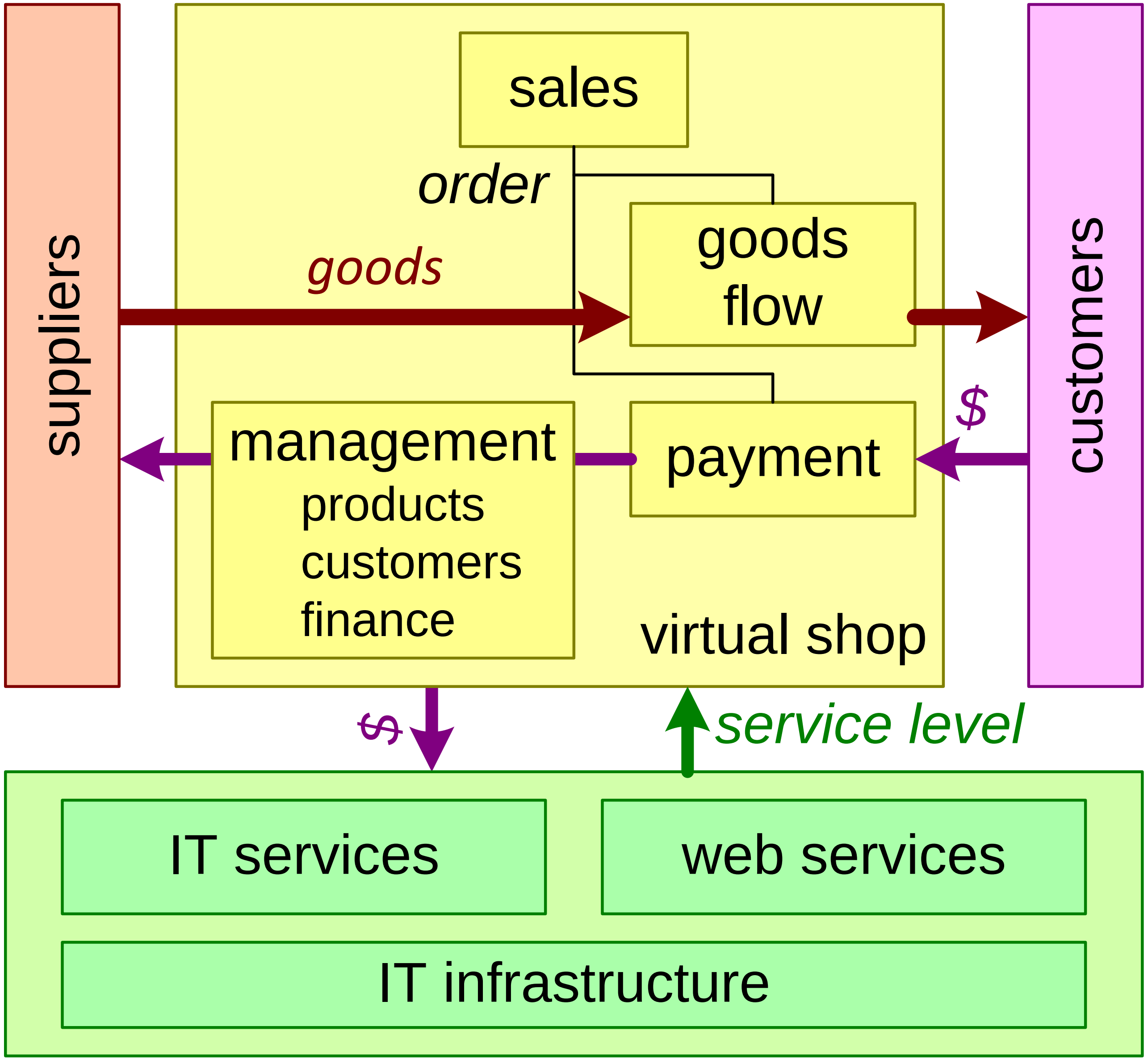
High Level Visual Models

+ value chain
+ map of competitors, partners, suppliers
+ context diagram
+ stakeholder diagram
+ infrastructure diagram
+ aspect diagrams e.g. security, data integrity, ..
+ customer key driver graph
+ life cycle key driver graph

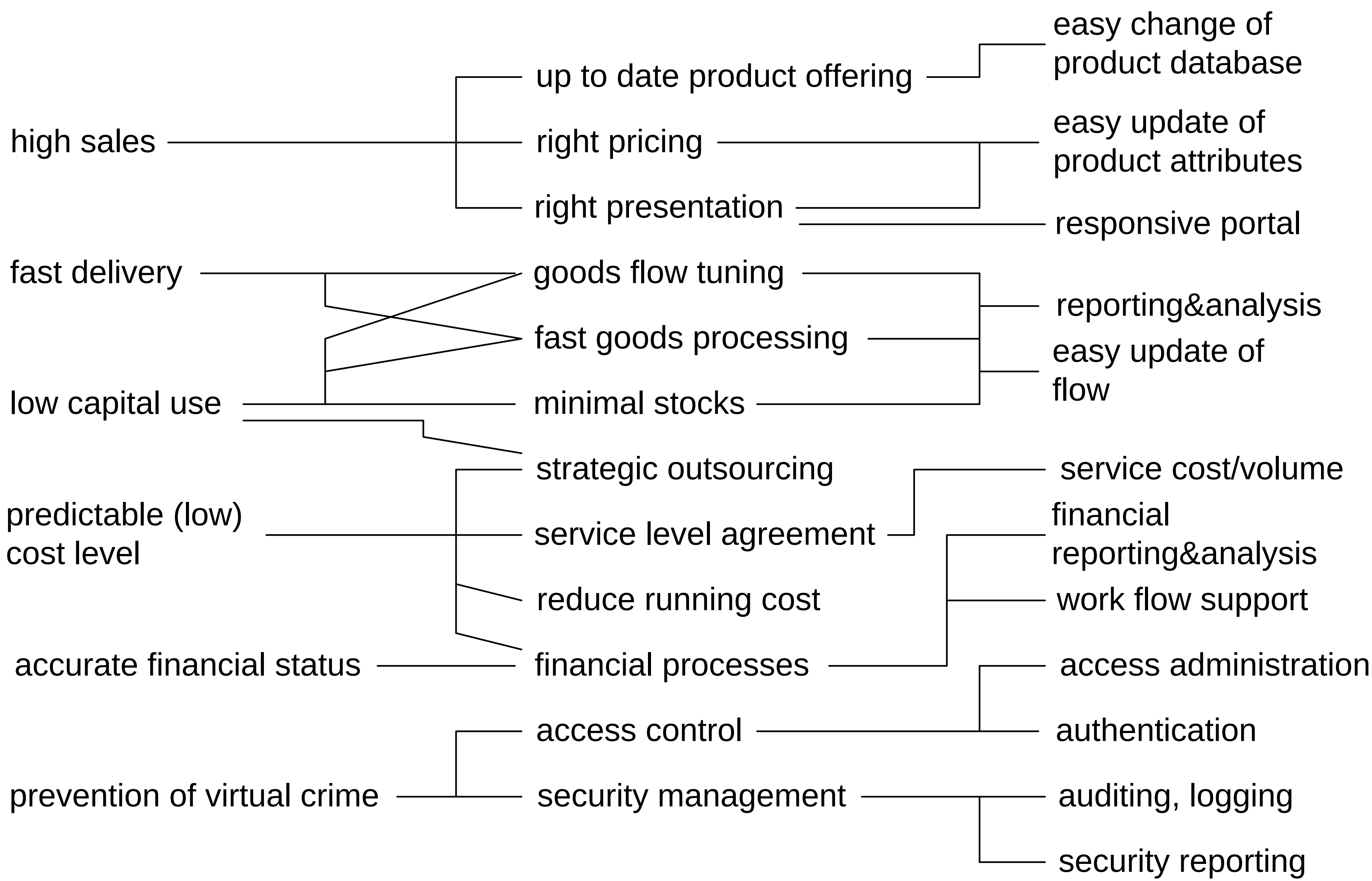
commercial
financial
legal
strategic
tactical
operational
social
technical

relations beyond actual system!

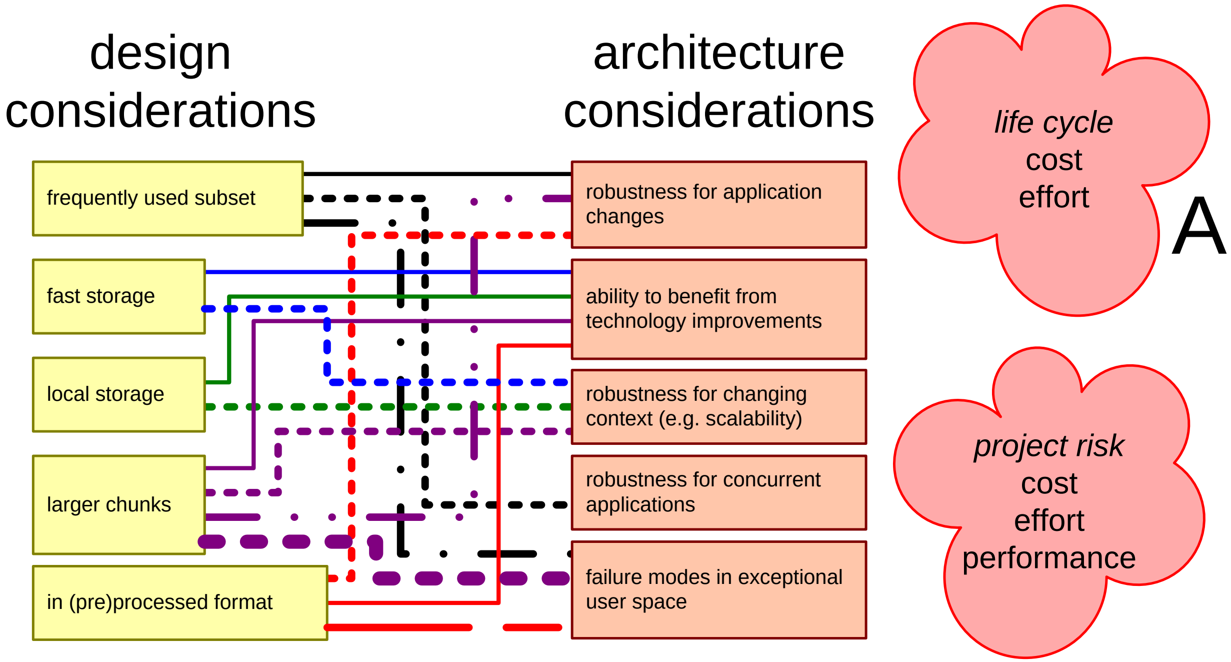
Simplified Web Shop Value Chain



Simplistic Customer Key Driver Graph



Example Assessment of Design Choices

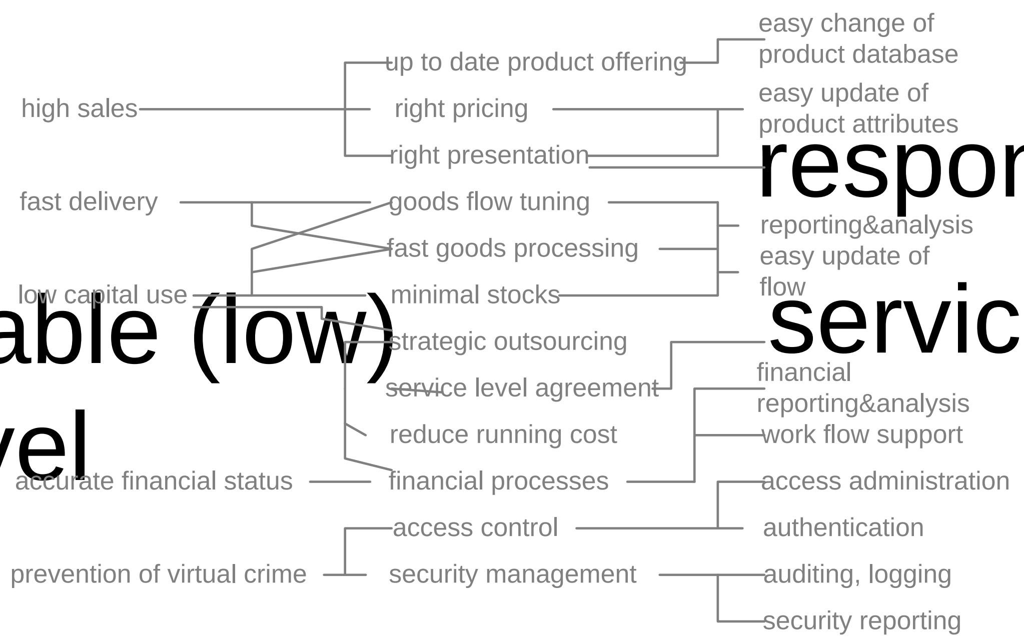


Are these concerns:
relevant?
significant?

What is the impact at enterprise level?

Example Zero Order Problem Statement

How does the
picture cache design
impact



responsive portal

service cost/volume

predictable (low) cost level

Zero Order Cost Model

total cost = f + s(v) + p * v + g * v

where

f = fixed base cost

s = service cost, see below

p = personnel cost including overheads

v = volume

g = goods flow handling

service cost s(v) = b + c * v

where

b = fixed base cost

c = cost / volume

v = volume

all including provider margin

Example Low Volume, Labor Intensive, Shop

low volume, labor intensive, shop

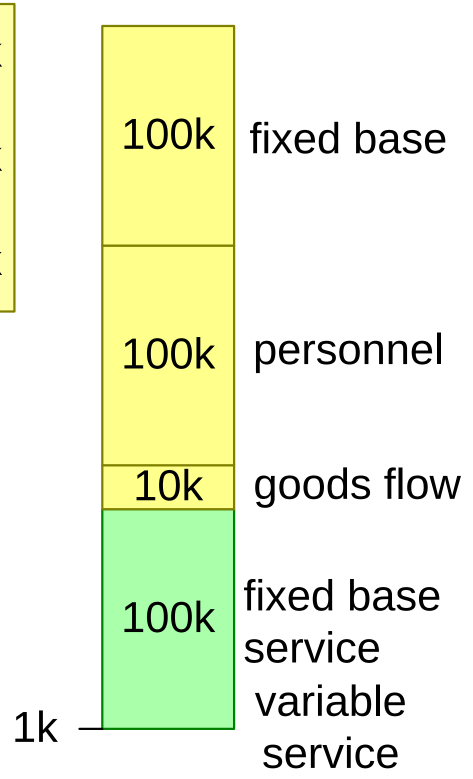
fixed costs and personnel cost dominate:
service cost changes have negligible impact on total cost!

total cost = f + s(v) + p * v + g * v
where
f = fixed base cost
s = service cost, see below
p = personnel cost including overheads
v = volume
g = goods flow handling

f =	100k
p =	1
v =	100k
g =	0.1
s(100k) =	101k

service cost s(v) = b + c * v
where
b = fixed base cost
c = cost / volume
v = volume
all including provider margin

b =	100k
c =	0.1



Example High Volume, Highly Automated, Shop

high volume, highly automated, shop

variable service costs dominate:
service cost changes have big impact on total cost!

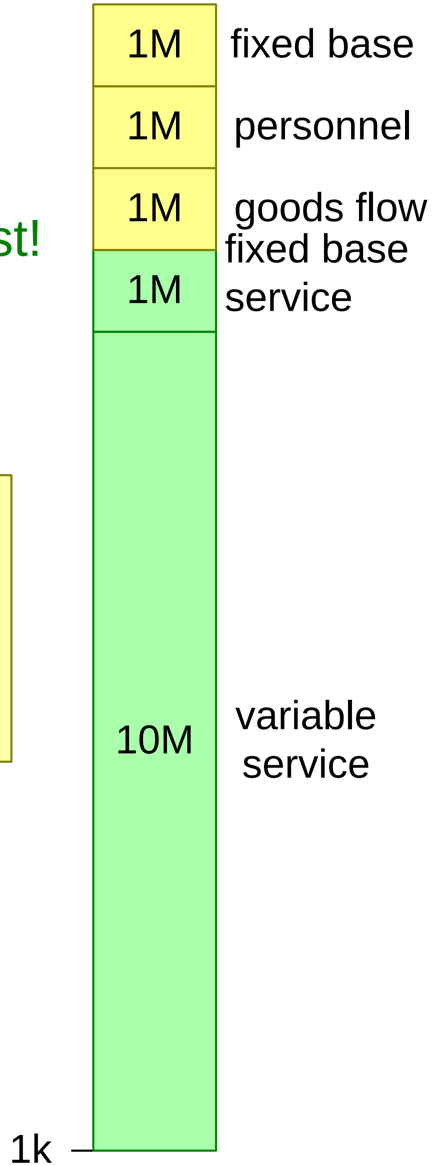
total cost = f + s(v) + p * v + g * v
- where
- f = fixed base cost
- s = service cost, see below
- p = personnel cost including overheads
- v = volume
- g = goods flow handling

f =	1M
p =	0.01
v =	100M
g =	0.01
s(100k) =	101k

service cost s(v) = b + c * v
- where
- b = fixed base cost
- c = cost / volume
- v = volume

all including provider margin

b =	1M
c =	0.1



Conclusion Zero Order Models

Very simple, very coarse, zero order models

provide insight in relevance of

specification and design issues.

These models are used to identify relevant issues

Modeling and Analysis: Life Cycle Models

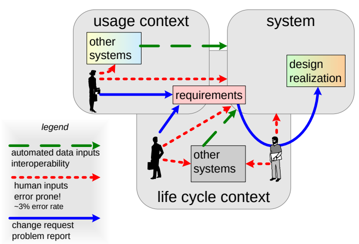
by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

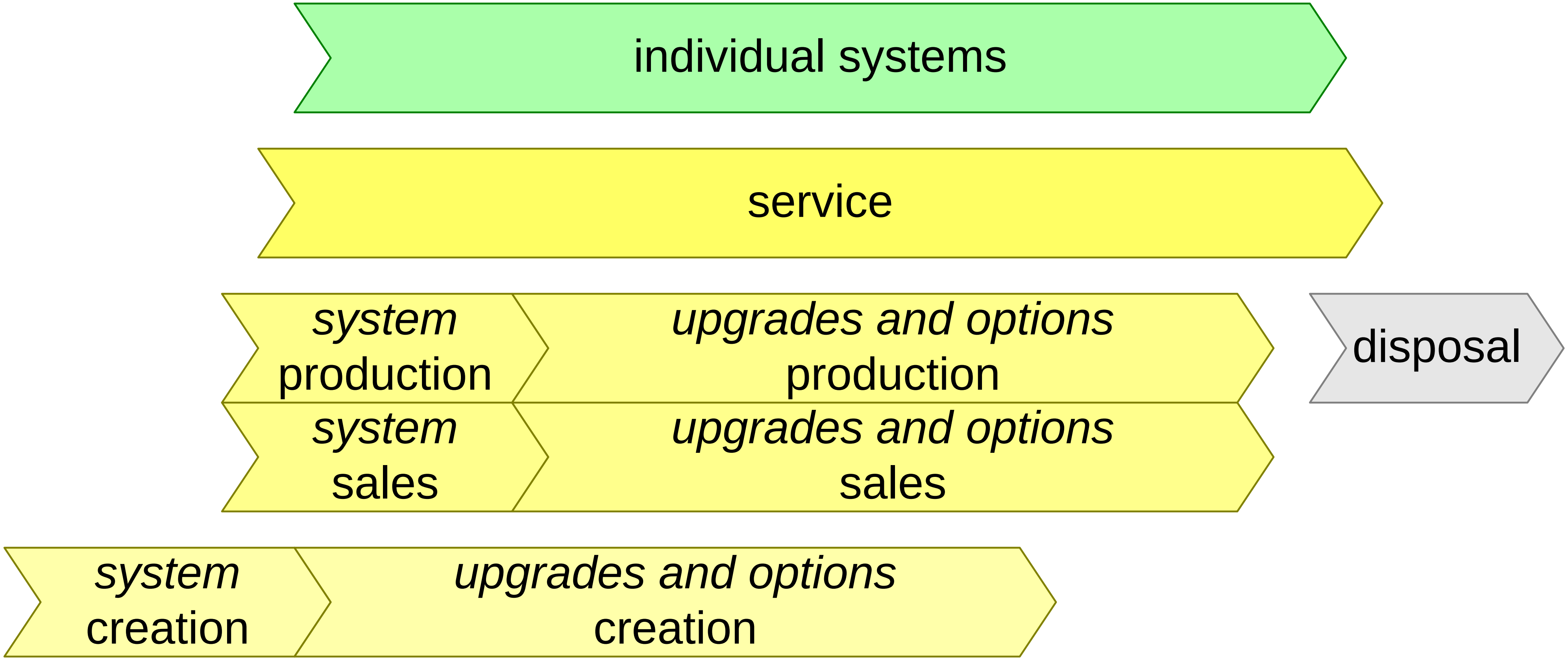
Products and enterprises evolve over time. This presentation explores the impact of these changes on the system and on the business by making (small and simple) models of life cycle aspects.

Distribution

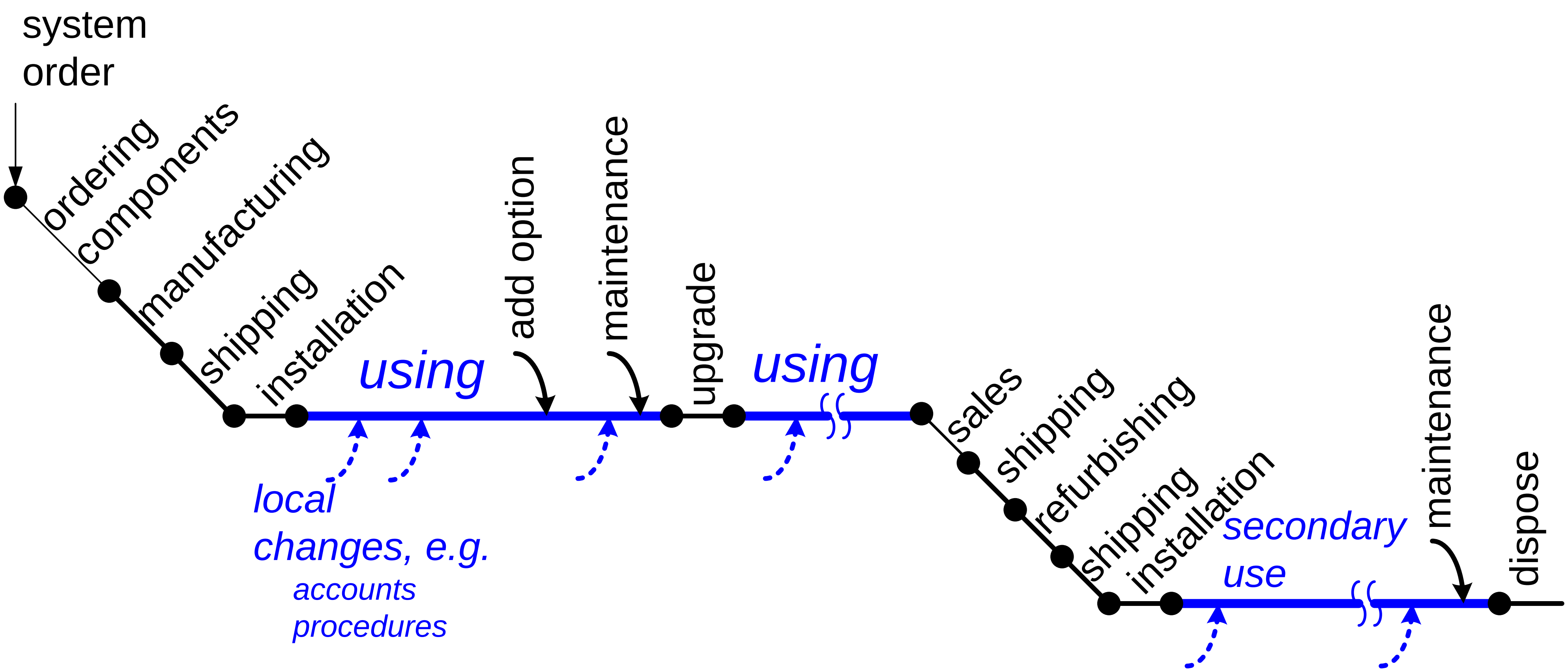
July 3, 2023
status: preliminary draft
version: 0.7



Product Related Life Cycles



System Life Cycle



Approach to Life Cycle Modeling

Identify potential life cycle changes and sources	
Characterize time aspect of changes	how often how fast
Determine required effort	amount type
Determine impact of change on system and context	performance reliability
Analyse risks	business

see reasoning

What May Change During the Life Cycle?

business volume

product mix

product portfolio

product attributes (e.g. price)

customers

personnel

suppliers

application, business processes

et cetera

www.homes4sale.com

www.apple.com/itunes/

www.amazon.com

www.ebay.com

www.shell.com

www.stevens.edu

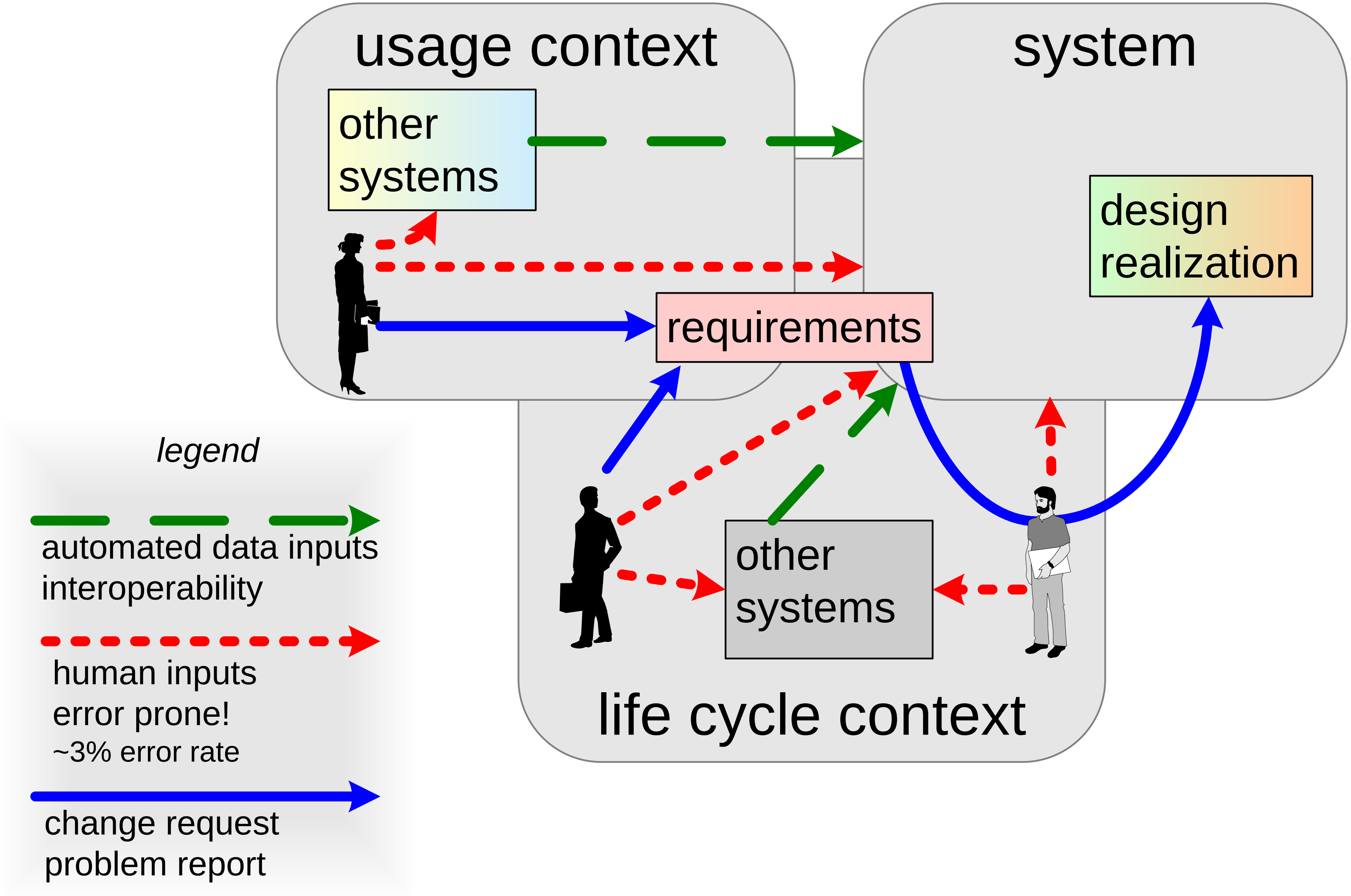
www.nokia.com

stock market

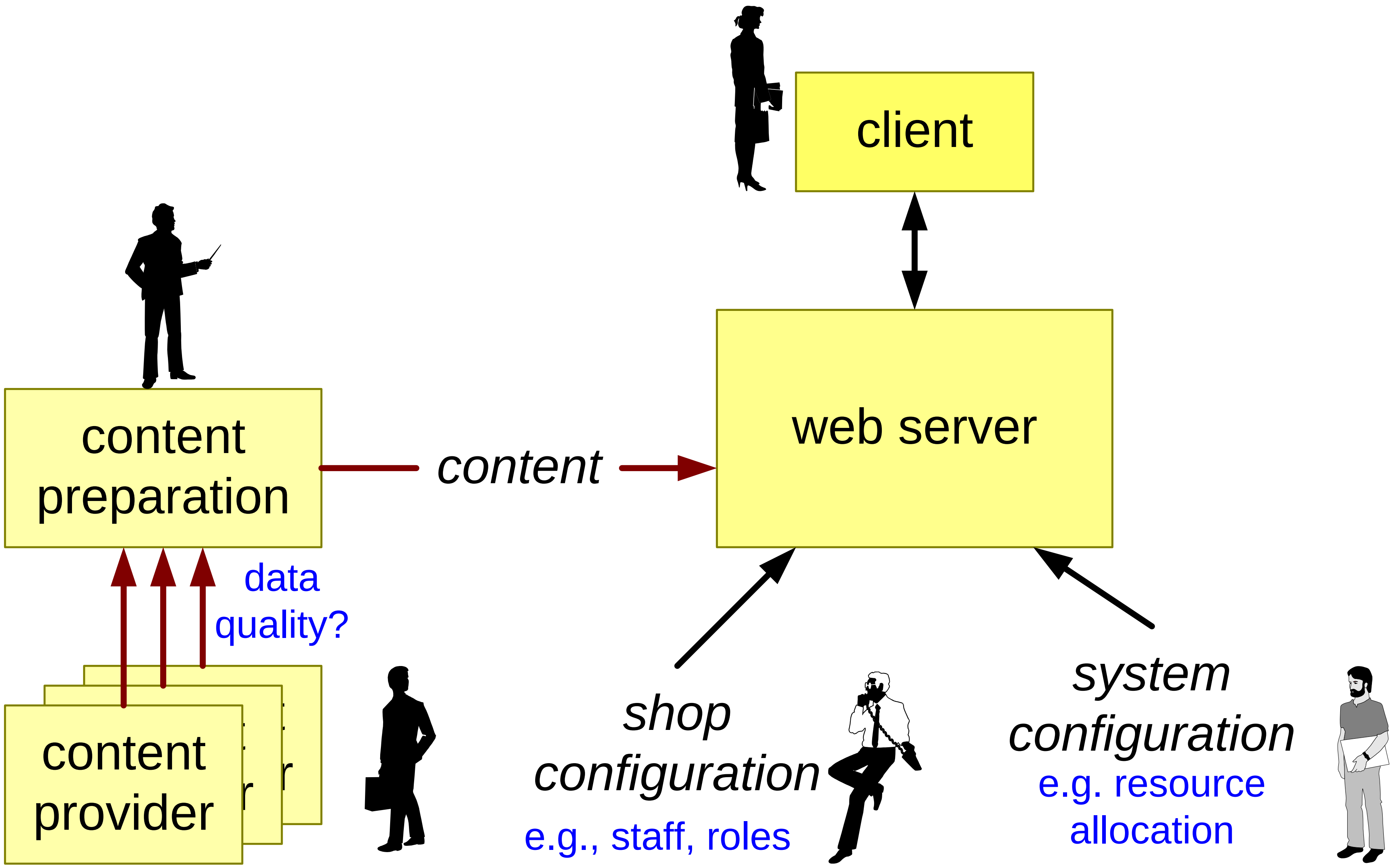
insurance company

local Dutch cheese shop

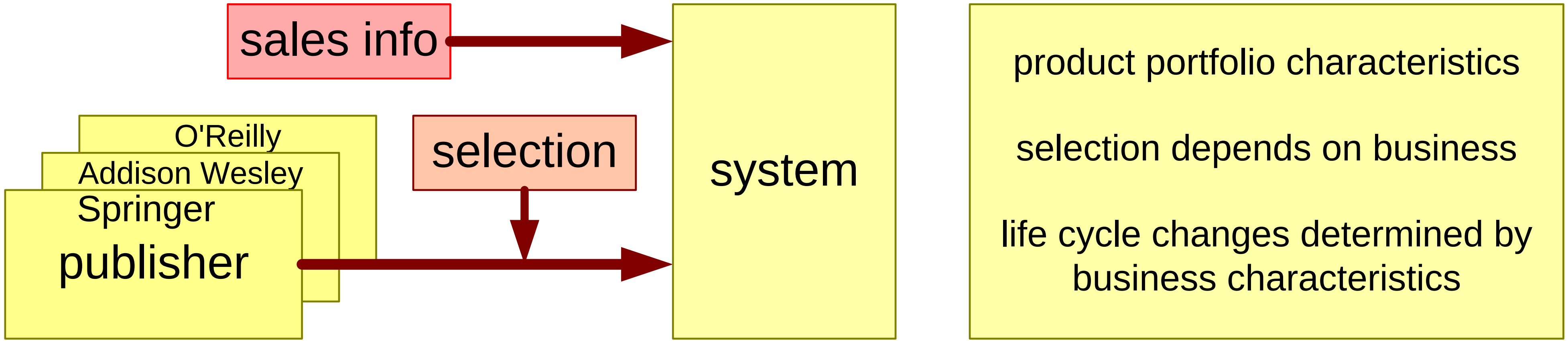
Simple Model of Data Sources of Changes



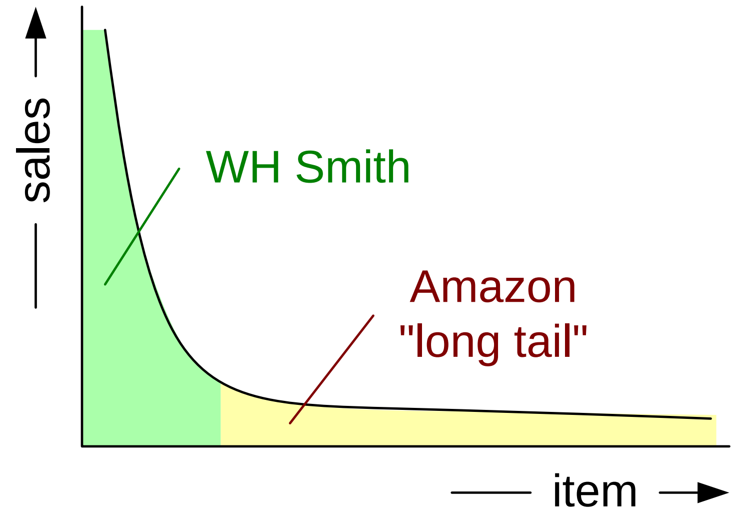
Data Sources of Web Server



Example Product Portfolio Change Books



source: 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

Example Customer Change

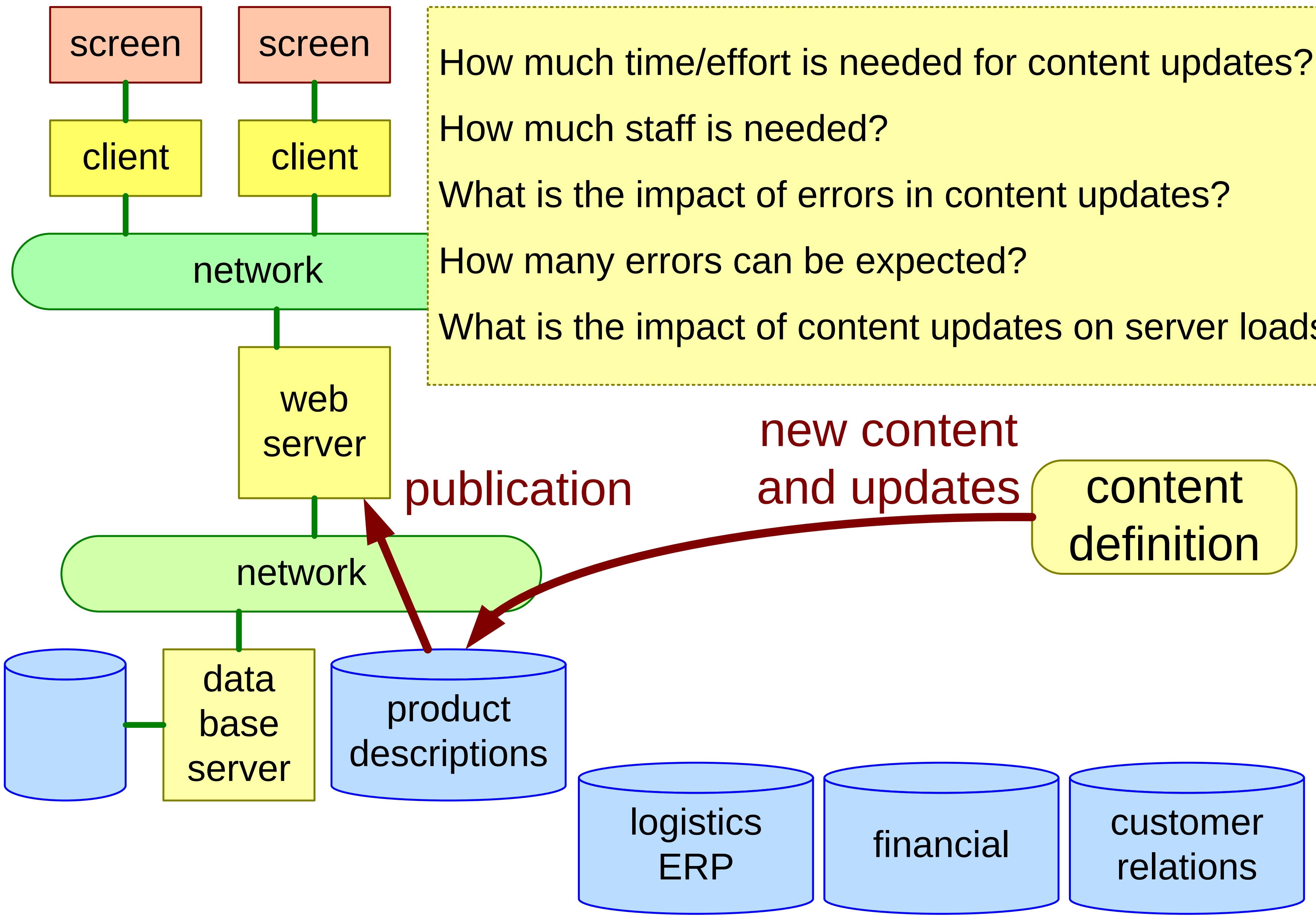
internet: broadband penetration

	Q1 '04	Q2 '04	growth in Q2 '04
Asia Pacific total	48M	54M	12.8%
China	15M	19M	26.1%
India	87k	189k	116.8%

http://www.apira.org/download/world_broadband_statistics_q2_2004.pdf

What is the expected growth of # customers?
What is the impact on system and infrastructure?
What is the impact on CRM (Customer Relation Management)?
What is the impact on customer, sales support staff?

Web Shop Content Update



How much time/effort is needed for content updates?

How much staff is needed?

What is the impact of errors in content updates?

How many errors can be expected?

What is the impact of content updates on server loads?

Web Shop Content Change Effort

prepare change 1 | prepare change 2 | prepare change n

review input
select info
layout&cosmetics
check-in

verify change 1 | verify change n

inspect source
inspect result

commit changes

$$effort_{changes} = n_{changes} * (t_{prepare} + t_{verify}) + t_{commit}$$

$$\#fte = effort_{changes} / \text{hours per day}$$

with $t_{prepare}$ = 4 min

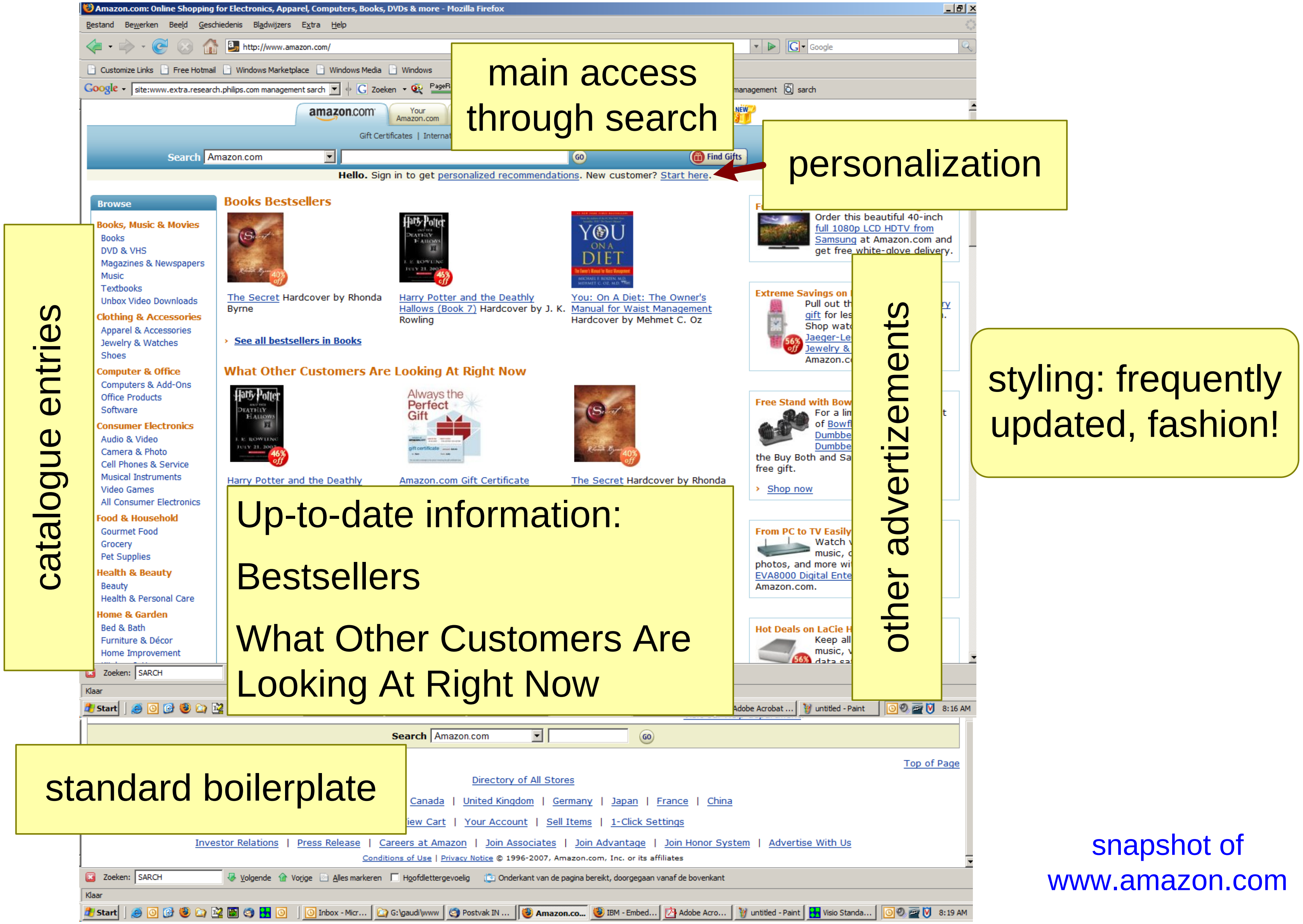
t_{verify} = 2 min

t_{commit} = 1 min

hours per day = 8 hours

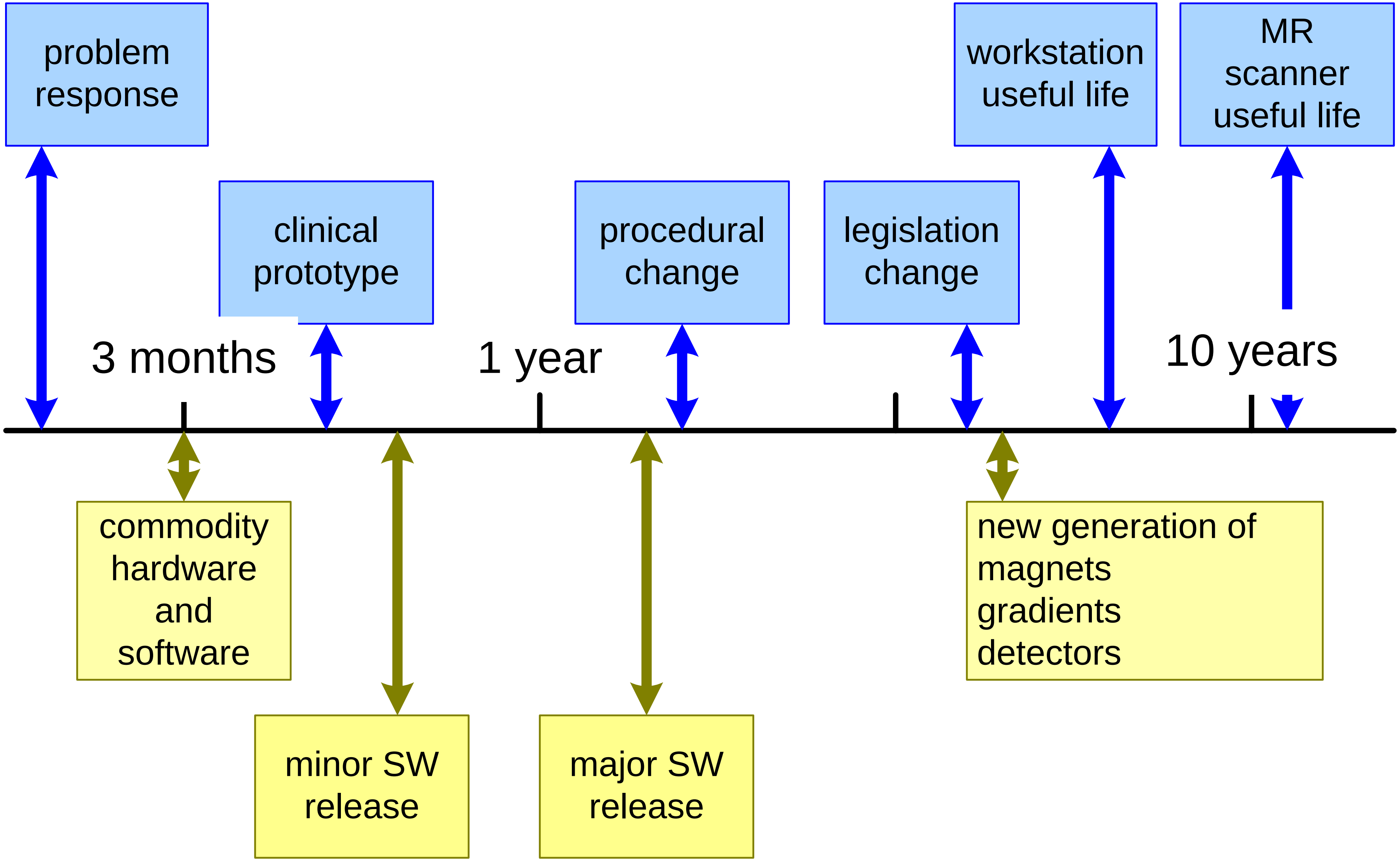
$n_{changes}$ per day	10	100	1000
$effort_{changes}$	1 uur	10 uur	100 uur
#fte	0.1	1	12

Example of Client Level Changes

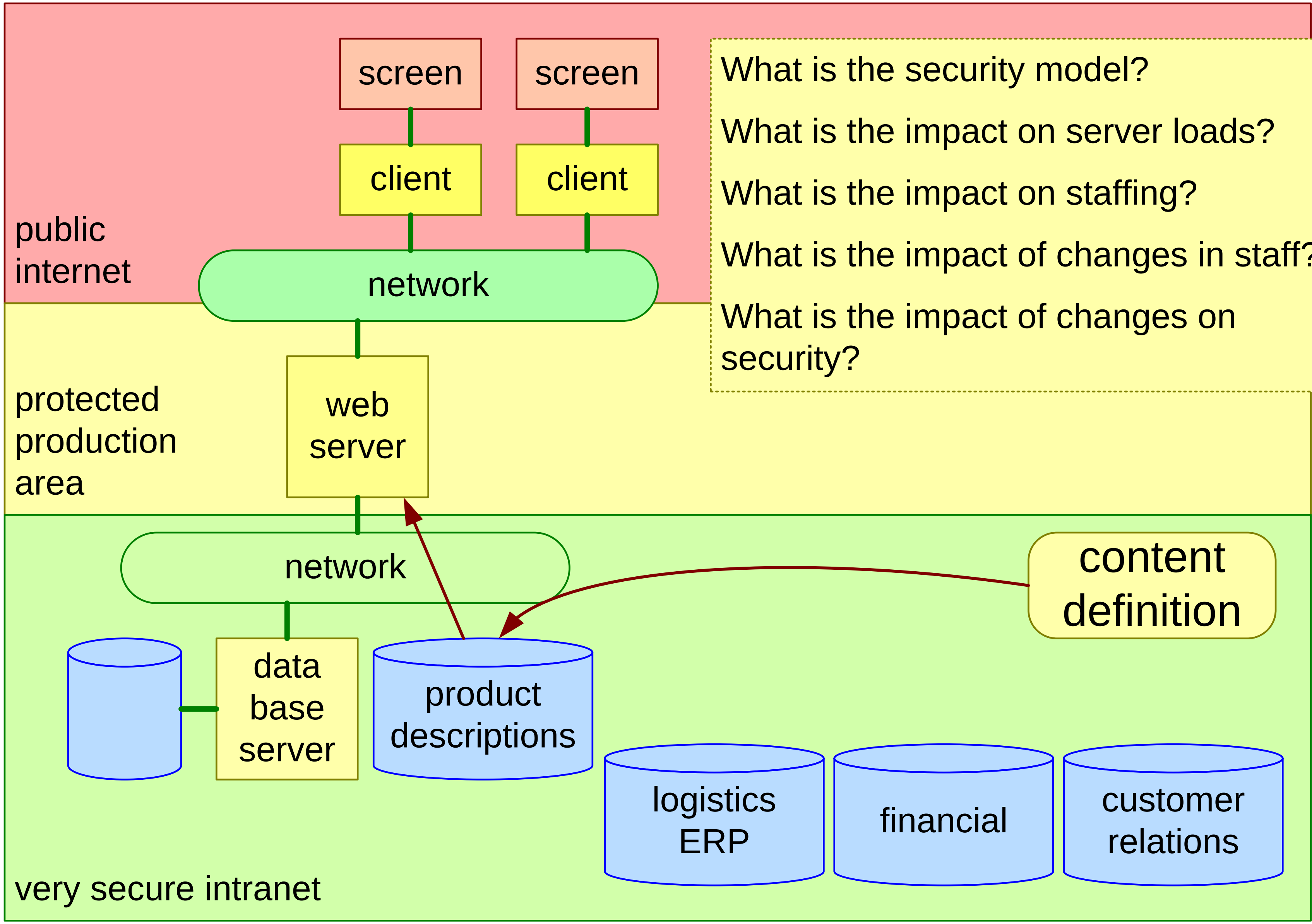


snapshot of www.amazon.com

Example of Time Scale Model for Changes



Web Shop Security and Changes



What is the security model?

What is the impact on server loads?

What is the impact on staffing?

What is the impact of changes in staff?

What is the impact of changes on security?

Web Shop Reliability and Changes

new faults = average fault density * #changes

$$\#errors = \sum_{faults} f(\text{severity}, \text{hit probability}, \text{detection probability})$$

	severity	hit probability	detection probability
Jansen iso *Janssen*	low	high	low
operator iso *sales repr*	high	high	medium

Simplistic Financial Computations for System Architects.

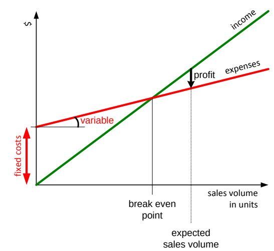
by *Gerrit Muller* USN-SE
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

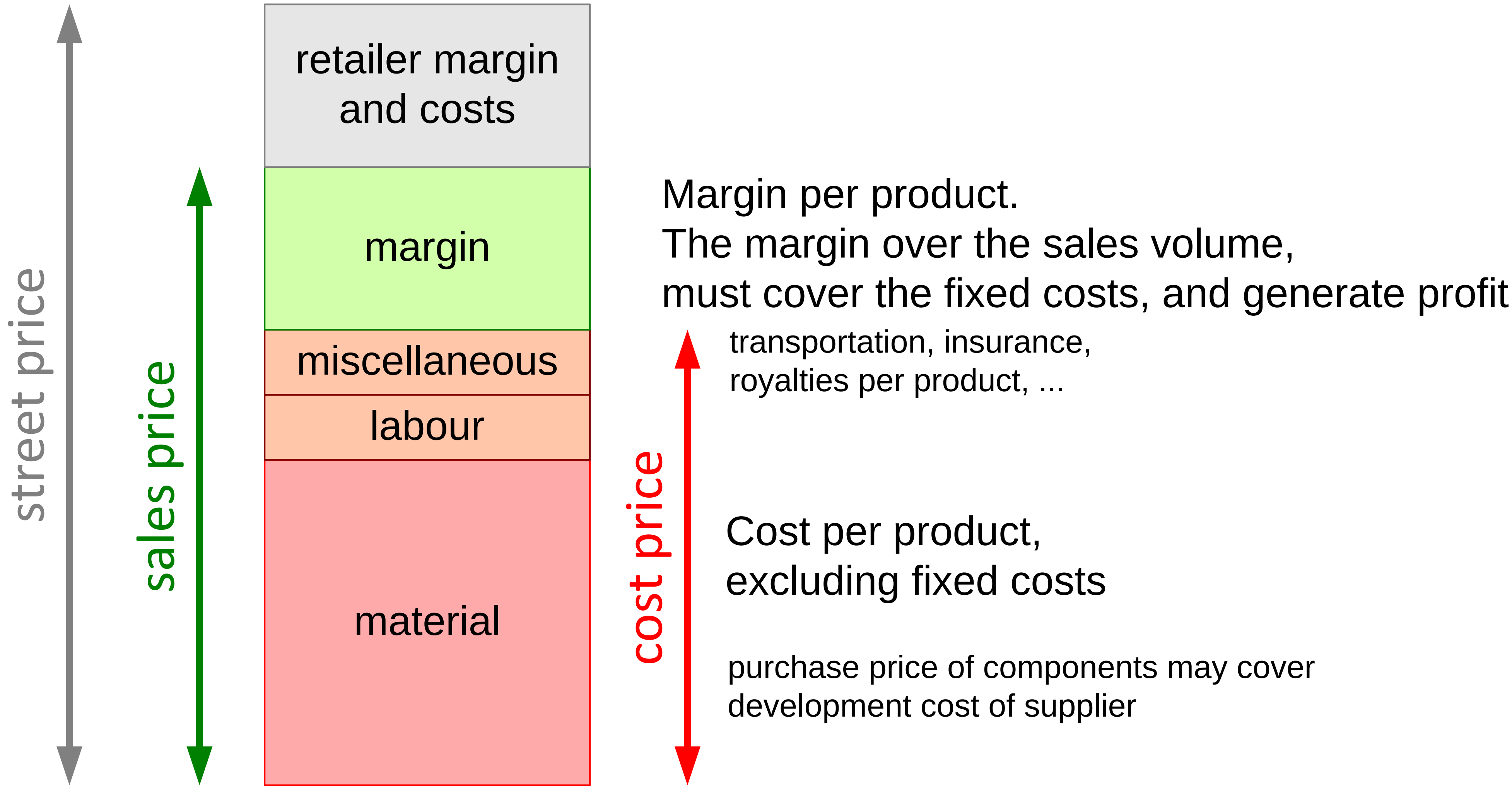
This document explains how simple financial estimates can be made by system architects. These simplistic estimates are useful for an architect to perform sanity checks on proposals and to obtain understanding of the financial impact of proposals. Note that architects will never have full fledged financial controller know how and skills. These estimates are zero order models, but real business decisions will have to be founded on more substantial financial proposals.

Distribution

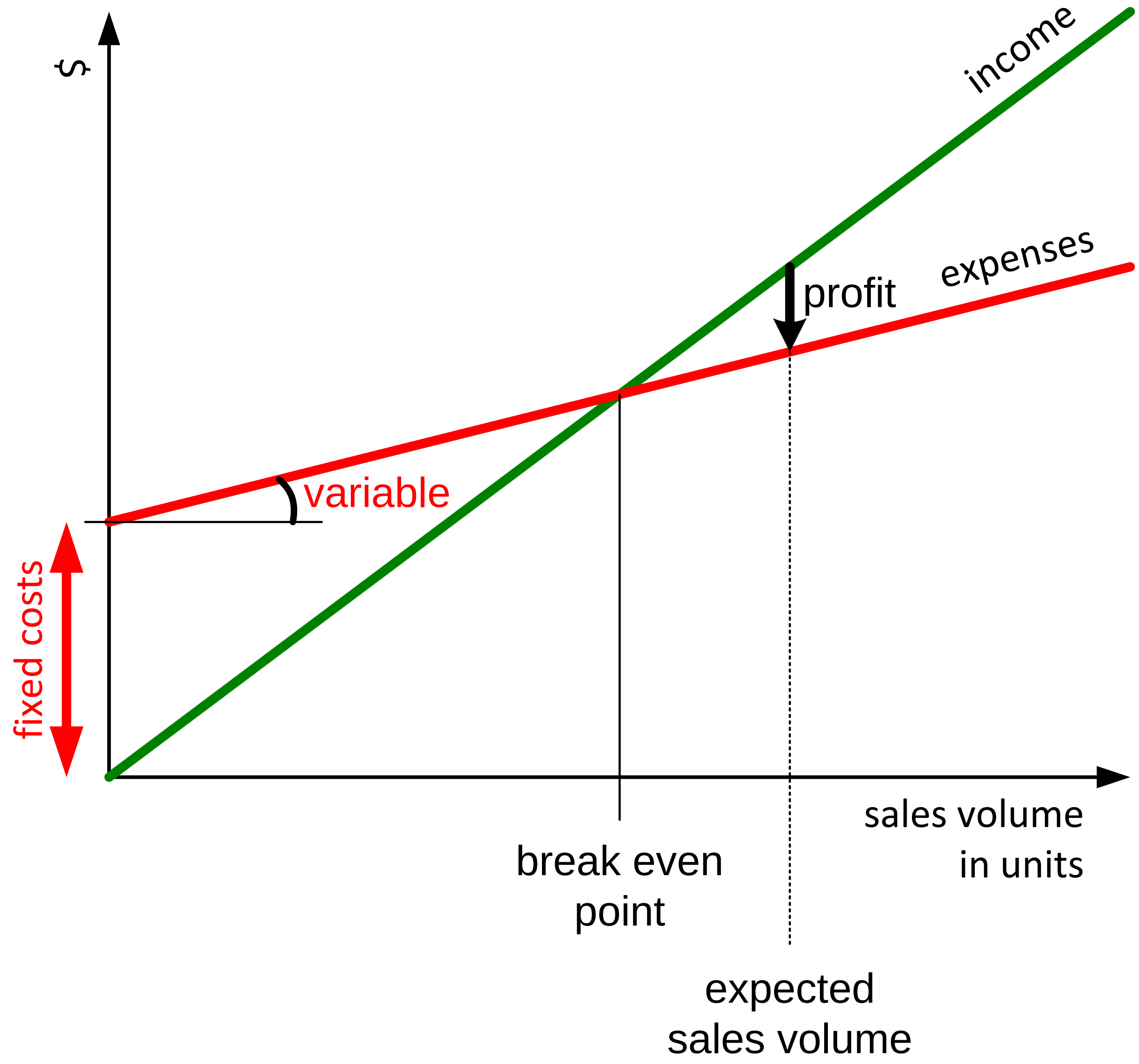
July 3, 2023
status: preliminary draft
version: 1.3



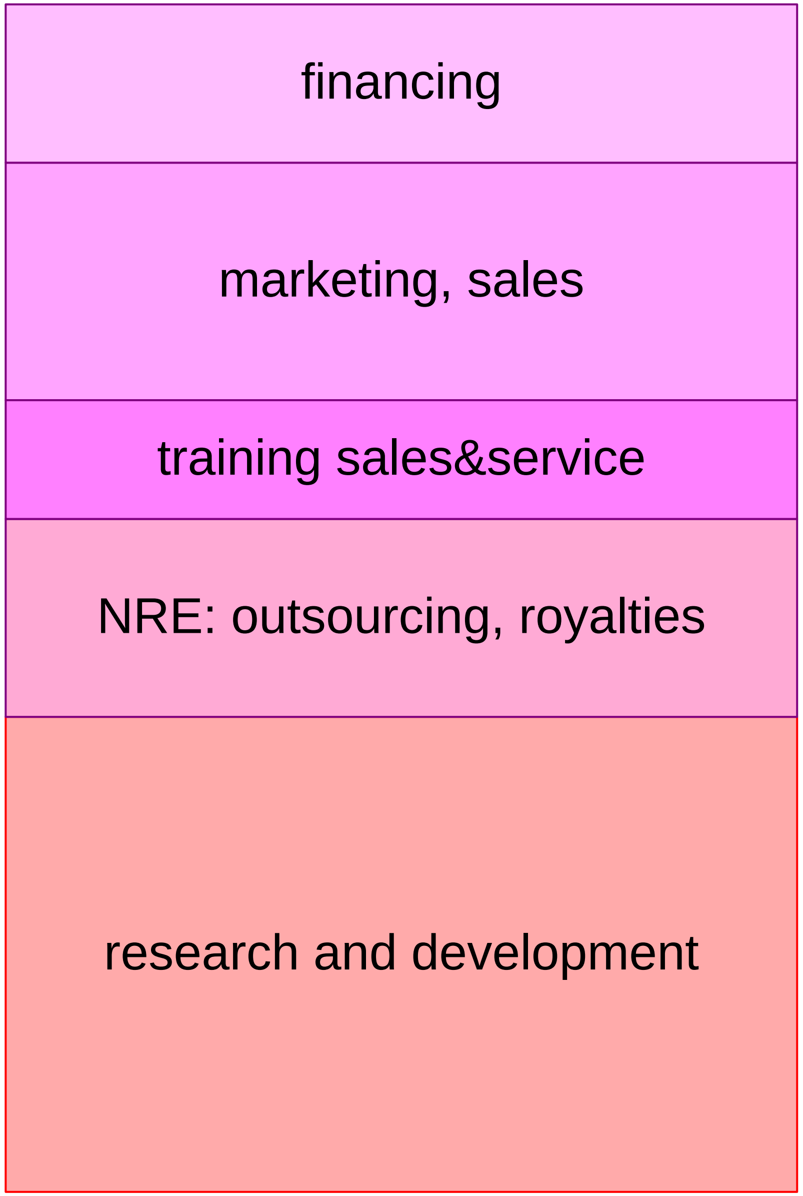
Product Margin = Sales Price - Cost



Profit as function of sales volume



Investments, more than R&D



business dependent:
pharmaceutics industry
sales cost >> R&D cost

strategic choice:
NRE or per product

including:
- staff, training, tools, housing
- materials, prototypes
- overhead
- certification

often a standard staffing rate is used
that covers most costs above:
R&D investment = Effort * rate

Income, more than product sales only

other recurring income		license fees pay per movie
services	$\sum_{services} income_{service}$	content, portal updates maintenance
options, accessories	$\sum_{options} sales\ price_{option} * volume_{option}$	
products	$sales\ price_{product} * volume_{product}$	

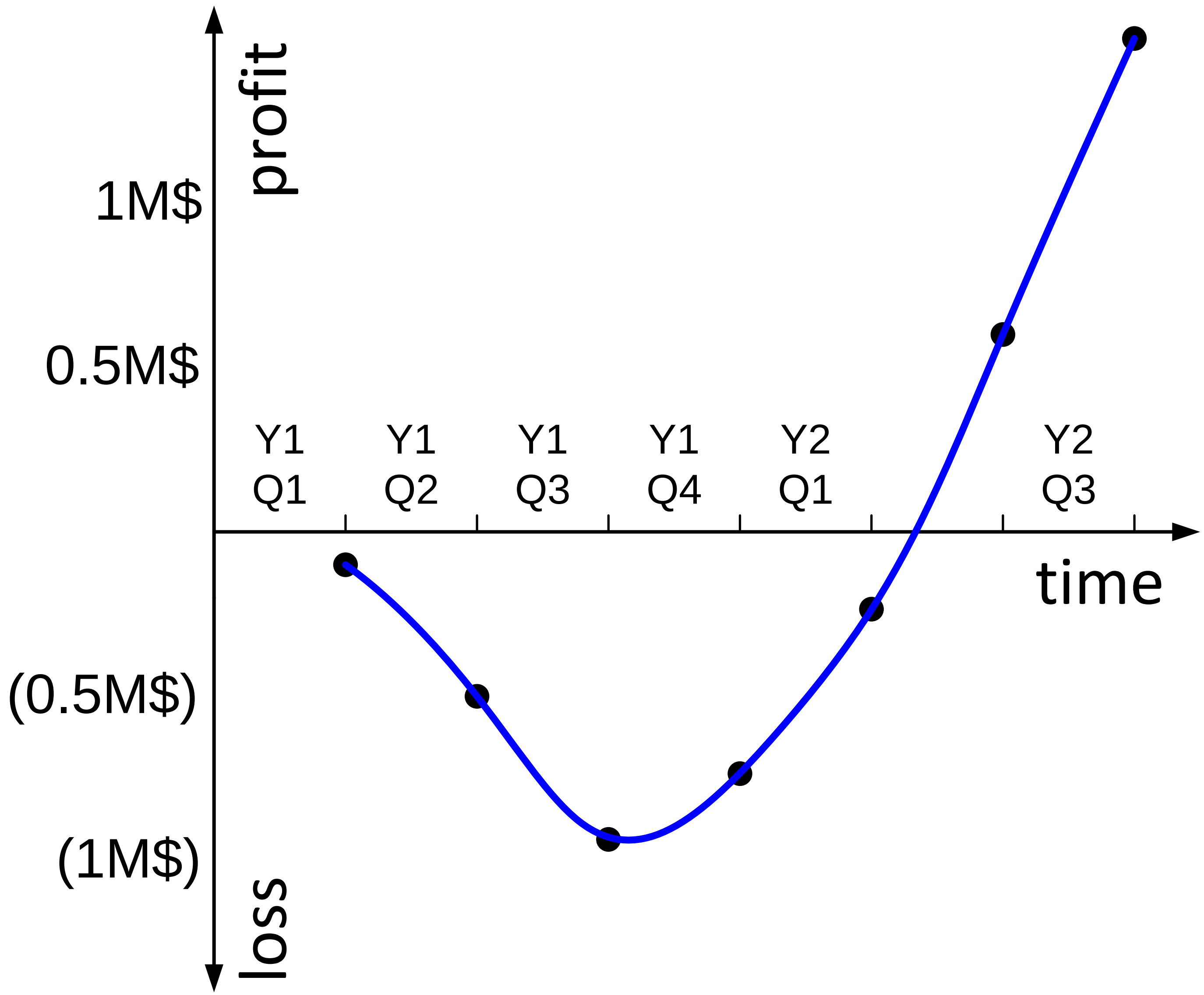
The Time Dimension

	Y1 Q1	Y1 Q2	Y1 Q3	Y1 Q4	Y2 Q1	Y2 Q2	Y2 Q3
investments	100k$	400k$	500k$	100k$	100k$	60k$	20k$
sales volume (units)	-	-	2	10	20	30	30
material & labour costs	-	-	40k$	200k$	400k$	600k$	600k$
income	-	-	100k$	500k$	1000k$	1500k$	1500k$
quarter profit (loss)	(100k$)	(400k$)	(440k$)	200k$	500k$	840k$	880k$
cumulative profit	(100k$)	(500k$)	(940k$)	(740k$)	(240k$)	600k$	1480k$

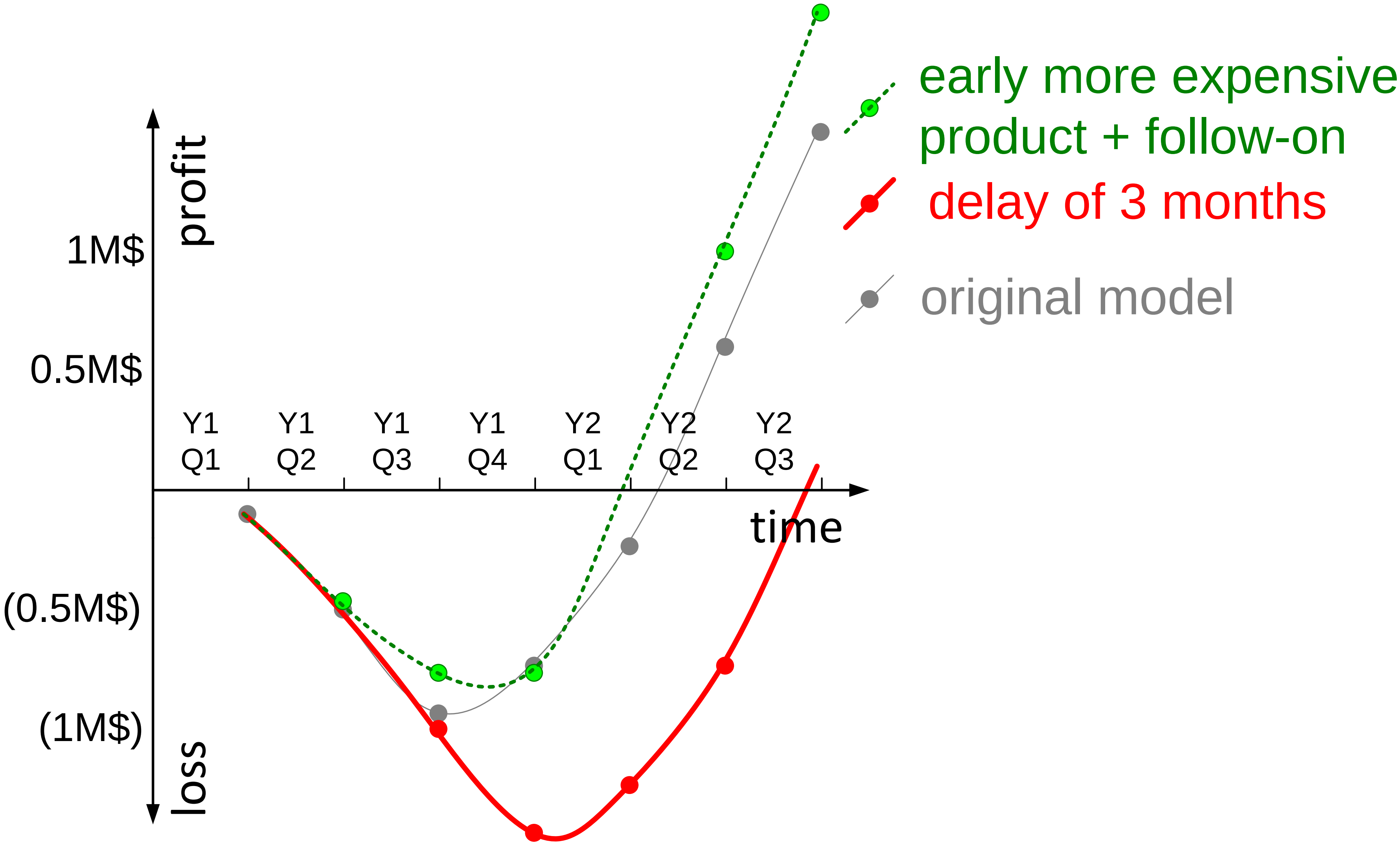
cost price / unit = 20k$
sales price / unit = 50k$

variable cost = sales volume * cost price / unit
income = sales volume * sales price / unit
quarter profit = income - (investments + variable costs)

The “Hockey” Stick

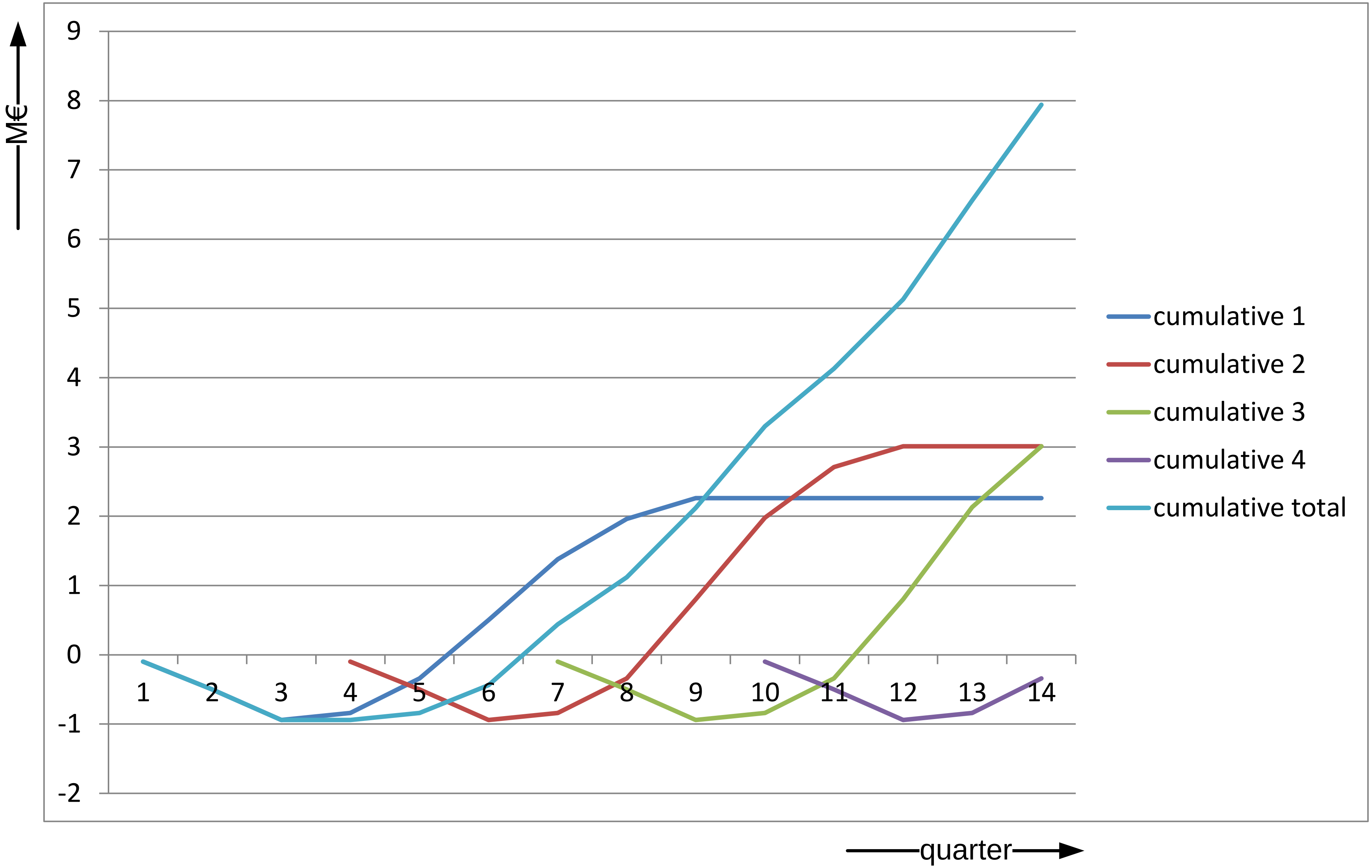


What if ...?



early more expensive
product + follow-on
delay of 3 months
original model
profit
loss
1M$
0.5M$
(0.5M$)
(1M$)
Y1
Q1
Y1
Q2
Y1
Q3
Y1
Q4
Y2
Q1
Y2
Q2
Y2
Q3
time

Stacking Multiple Developments



Fashionable financial yardsticks

Return On Investments (ROI)

Net Present Value

Return On Net Assets (RONA) leasing reduces assets, improves RONA

turnover / fte outsourcing reduces headcount, improves this ratio

market ranking (share, growth) "only numbers 1, 2 and 3 will be profitable"

R&D investment / sales in high tech segments 10% or more

cash-flow fast growing companies combine profits with negative cash-flow, risk of bankruptcy

The application view

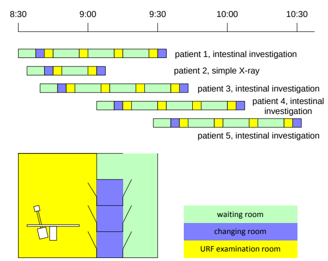
by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

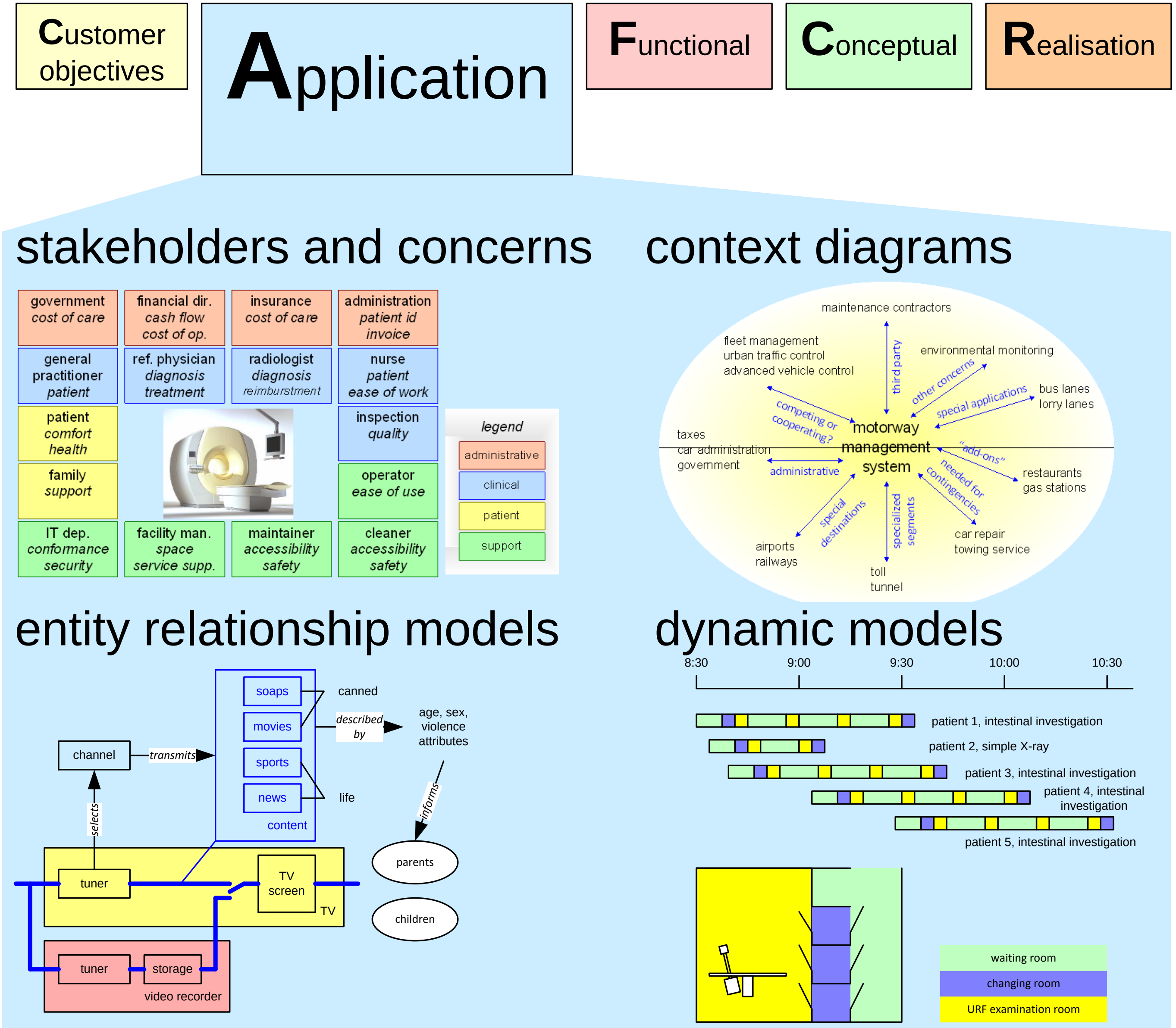
The purpose of the application view is described. A number of methods or models is given to use in this view: stakeholder and concerns, context diagram, static entity relationship models and dynamic flow models.

Distribution

July 3, 2023
status: preliminary draft
version: 0.2



Application view overview



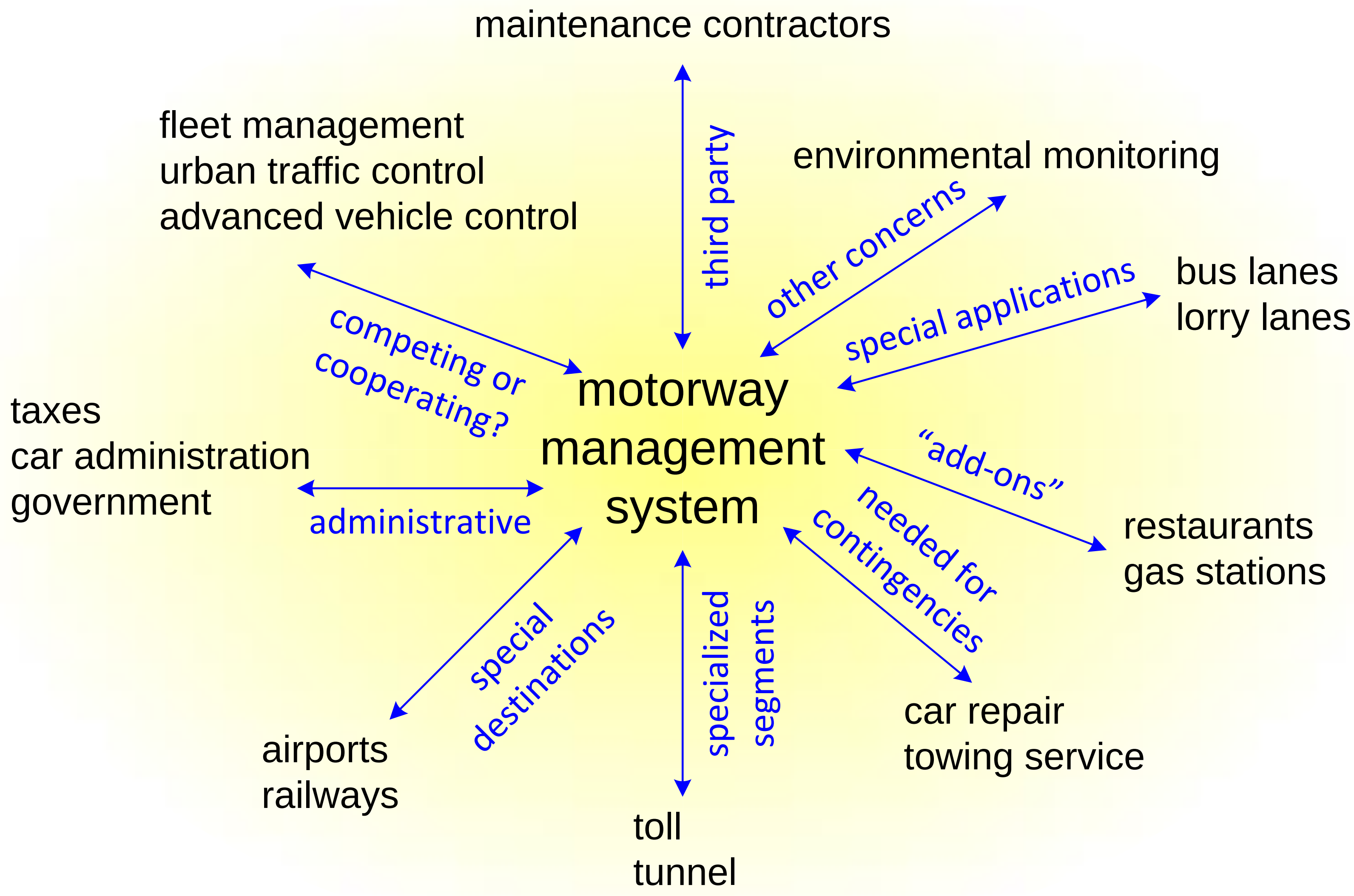
Stakeholders and concerns MRI scanner

government *cost of care*	financial dir. *cash flow* *cost of op.*	insurance *cost of care*	administration *patient id* *invoice*
general practitioner *patient*	ref. physician *diagnosis* *treatment*	radiologist *diagnosis* *reimburstment*	nurse *patient* *ease of work*
patient *comfort* *health*			inspection *quality*
family *support*			operator *ease of use*
IT dep. *conformance* *security*	facility man. *space* *service supp.*	maintainer *accessibility* *safety*	cleaner *accessibility* *safety*

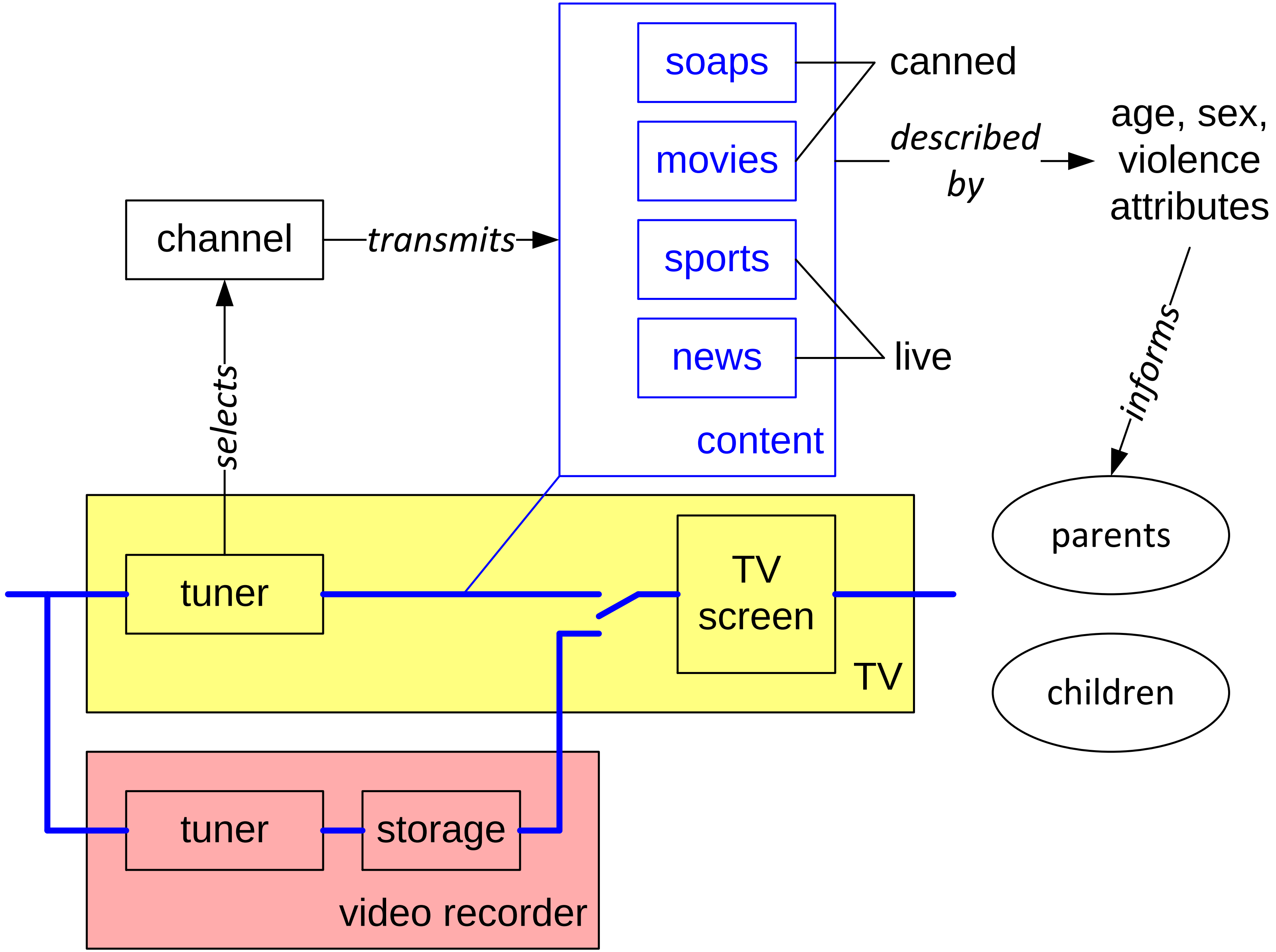
legend

- administrative
- clinical
- patient
- support

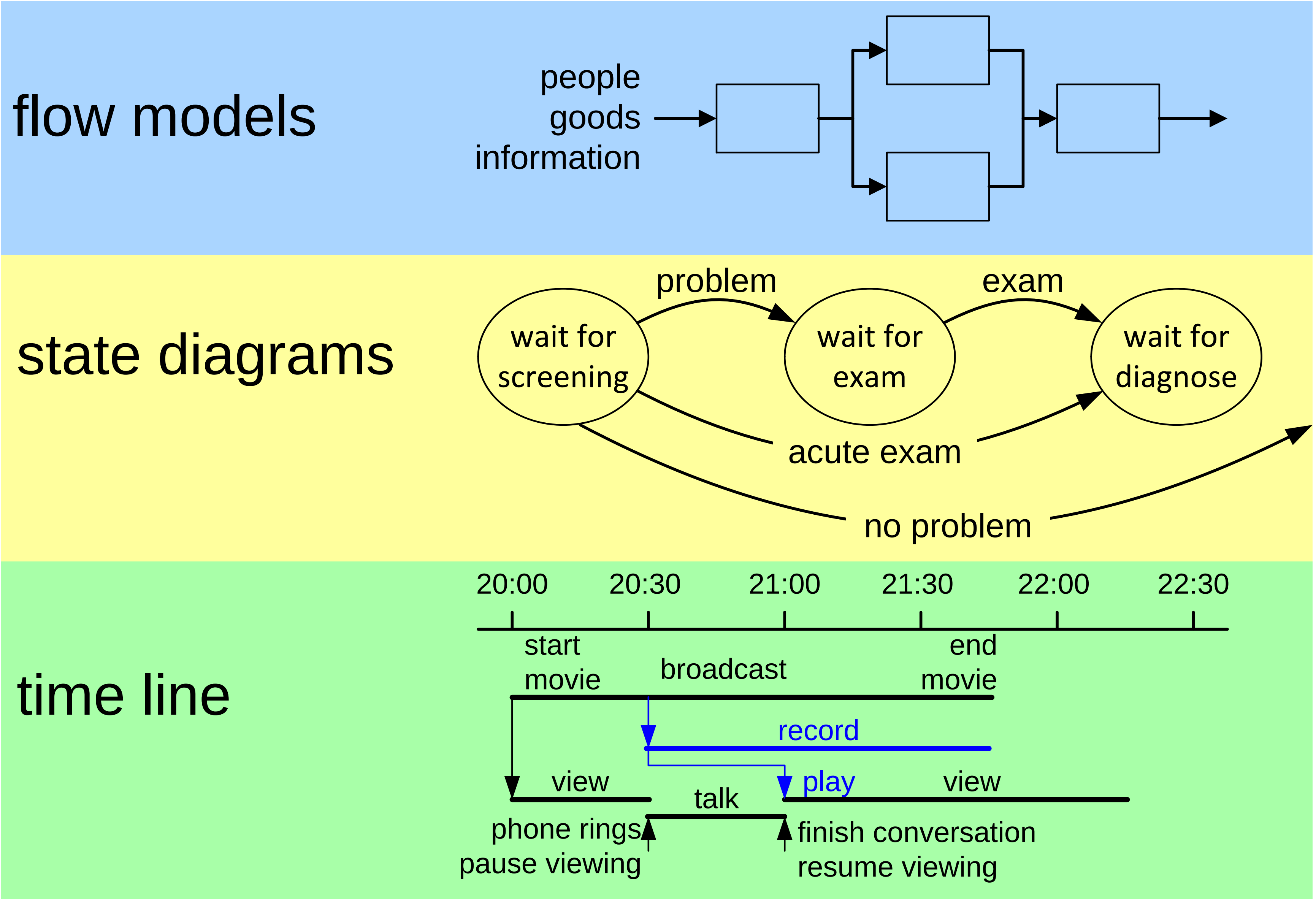
Context of motorway management system



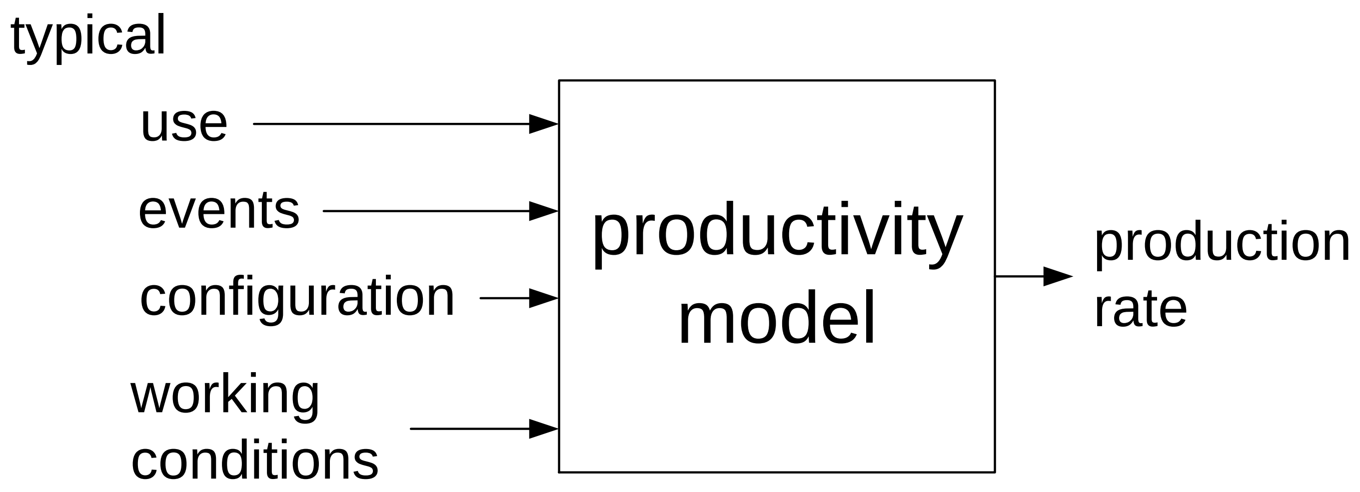
Example of simple TV application model



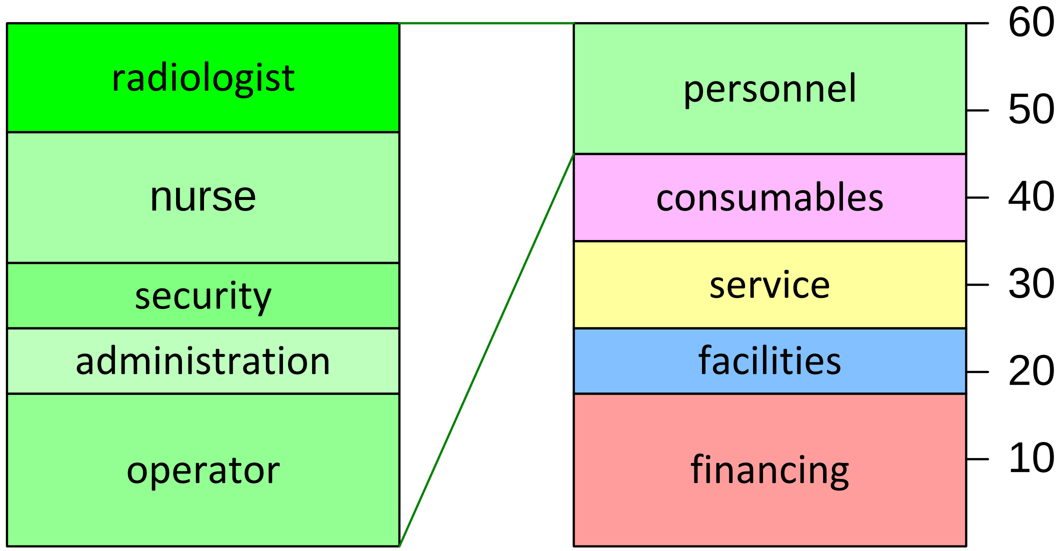
Examples of dynamic models



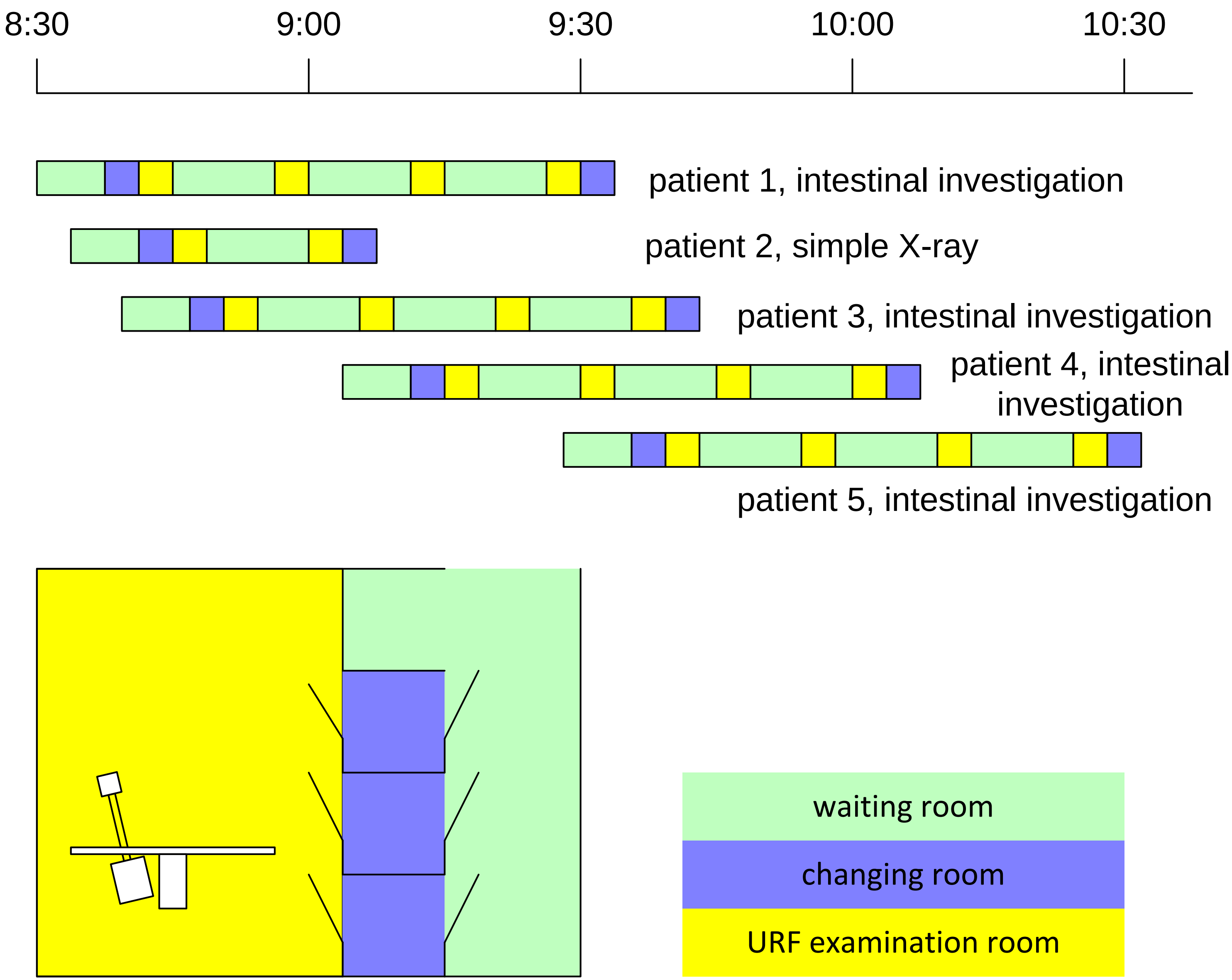
Productivity and Cost models



Cost Of Ownership model



Dynamics of an URF examination room



Exercise

Make a context diagram:

- What other related systems and applications are used?
- How do these relate with our system
- Visualize the context as diagram
- Make supporting diagrams for main application views

Reflection on Exercise

+ Context diagram lift insight to a higher level

~ Our system or application is only a fraction of the customers world

- Application models can become too generic or abstract

Summary Fundamentals of Application

Conclusions

Real requirements are driven by understanding of the customer's application

Complexity of finance is no excuse for ignoring all financial aspects; simplified models provide a lot of insight

Techniques, Models, Heuristics of this module

Simplistic financial models

TBD

Module Modeling and Analysis: Integration and Reasoning

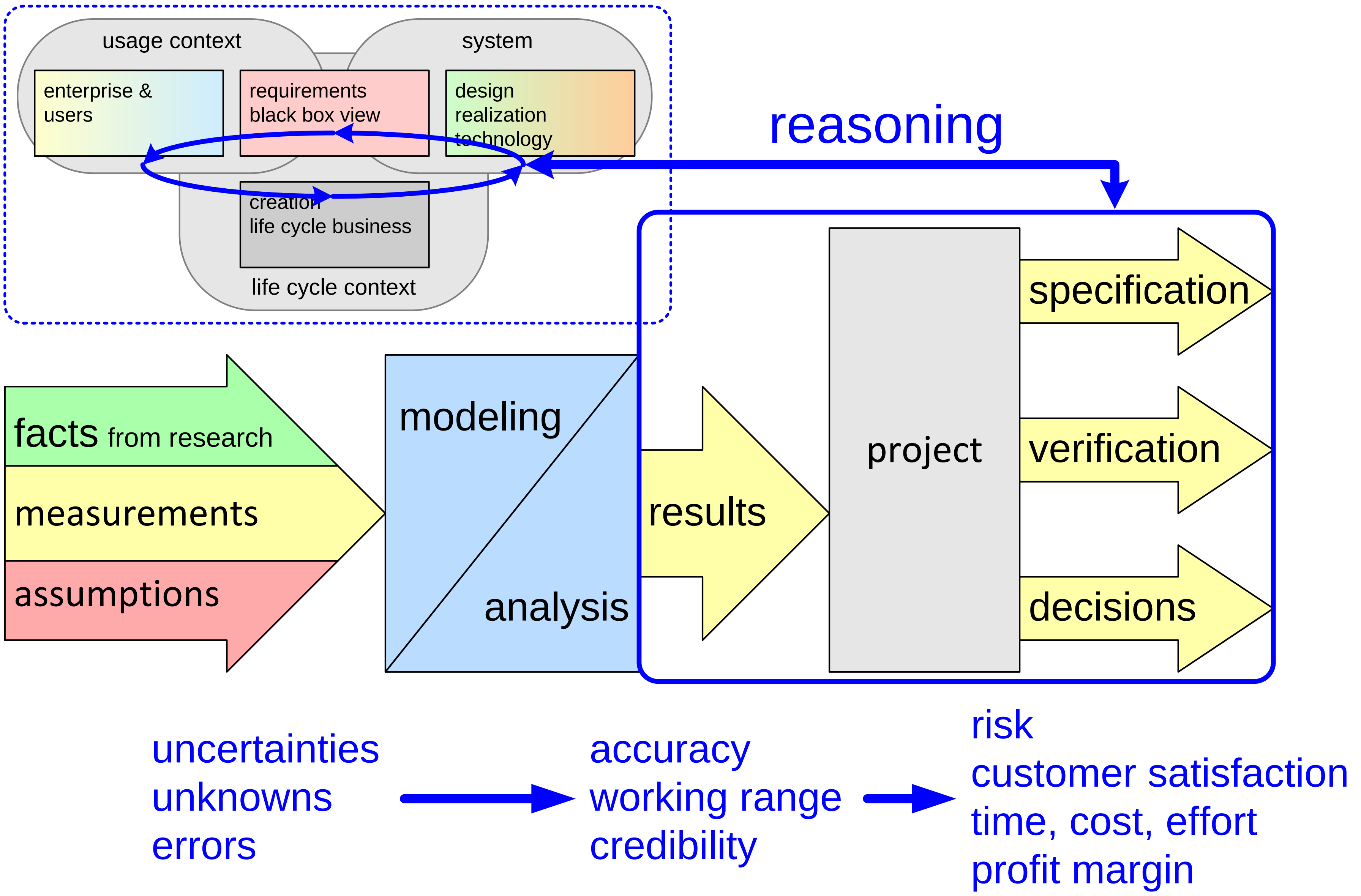
by *Gerrit Muller*   HSN-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module addresses the integration of small or partial models into bigger models. We also discuss how multiple models are used and how to reason using multiple models.

July 3, 2023
status: planned
version: 0.3

Where are we in the Course?



Modeling and Analysis: Reasoning

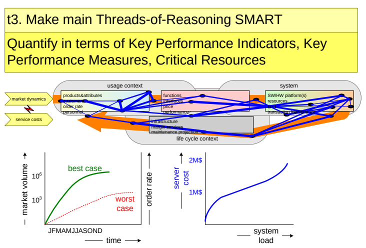
by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

Models are made to facilitate decision making. These decisions range from business decisions, such as Service Level Agreements, to requirements, and to detailed design decisions. The space of decisions is huge and heterogeneous. The proposed modeling approach is to use multiple small and simple models. In this paper we discuss how to reason by means of multiple models.

Distribution

July 3, 2023
status: preliminary draft
version: 0.6



content

From chaos to order: inputs, assumptions, models and decisions

Reasoning approach: stepwise top-down and bottom-up

Life cycles of models in relation to project life cycle

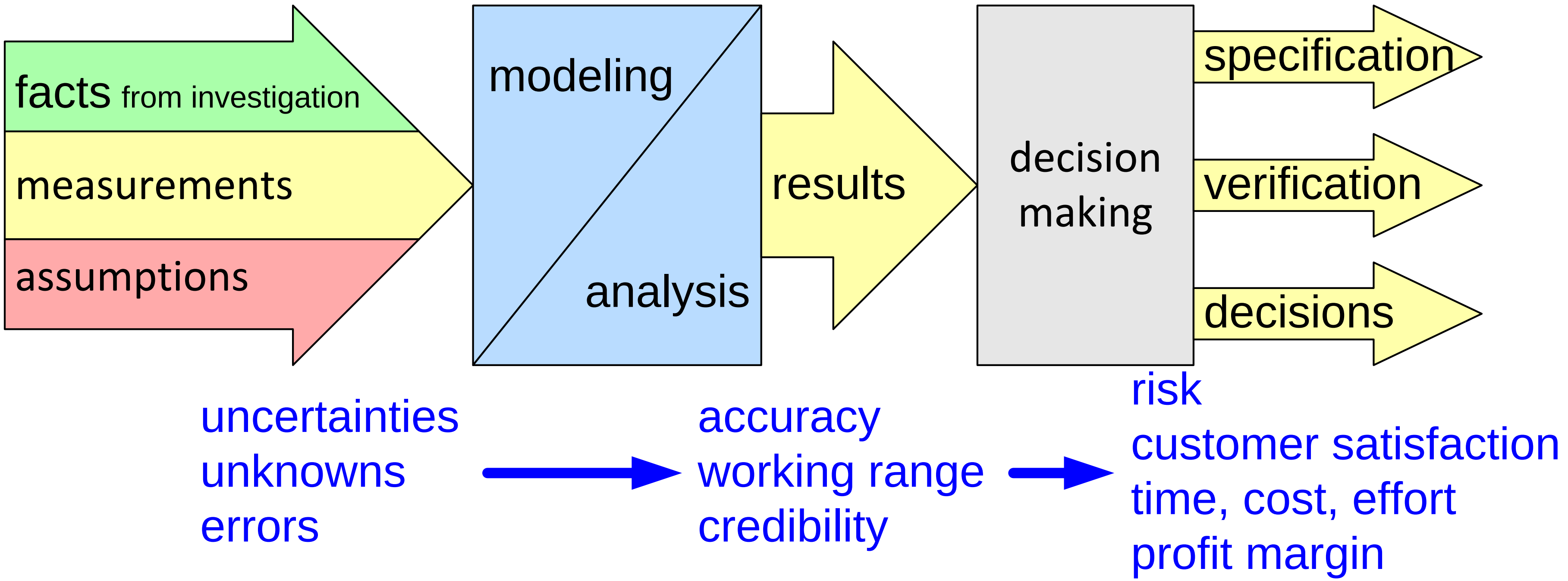
Purpose of Modeling

How to use multiple models to facilitate decisions?

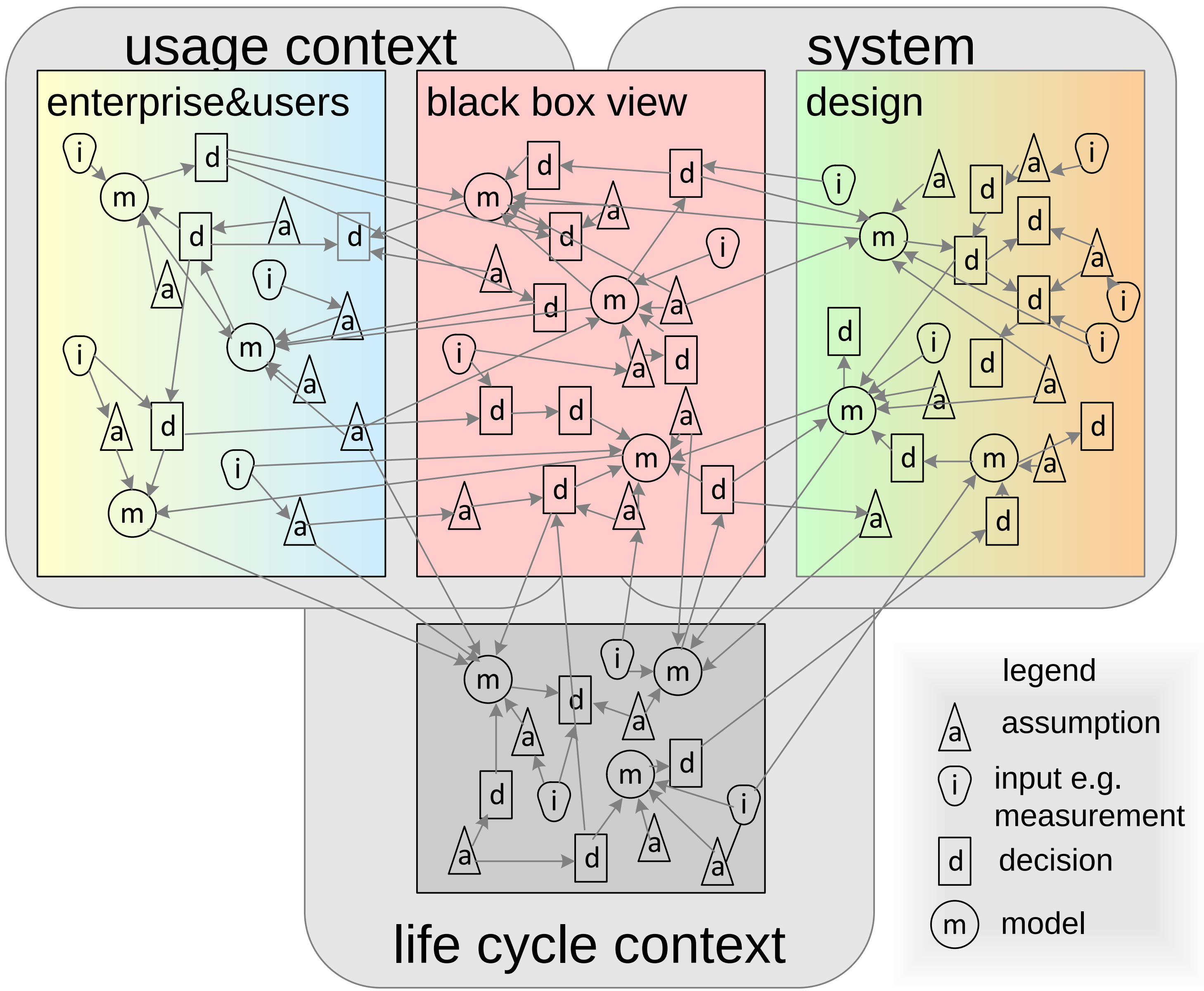
How to get from many fragments to integral insight?

How many models do we need?

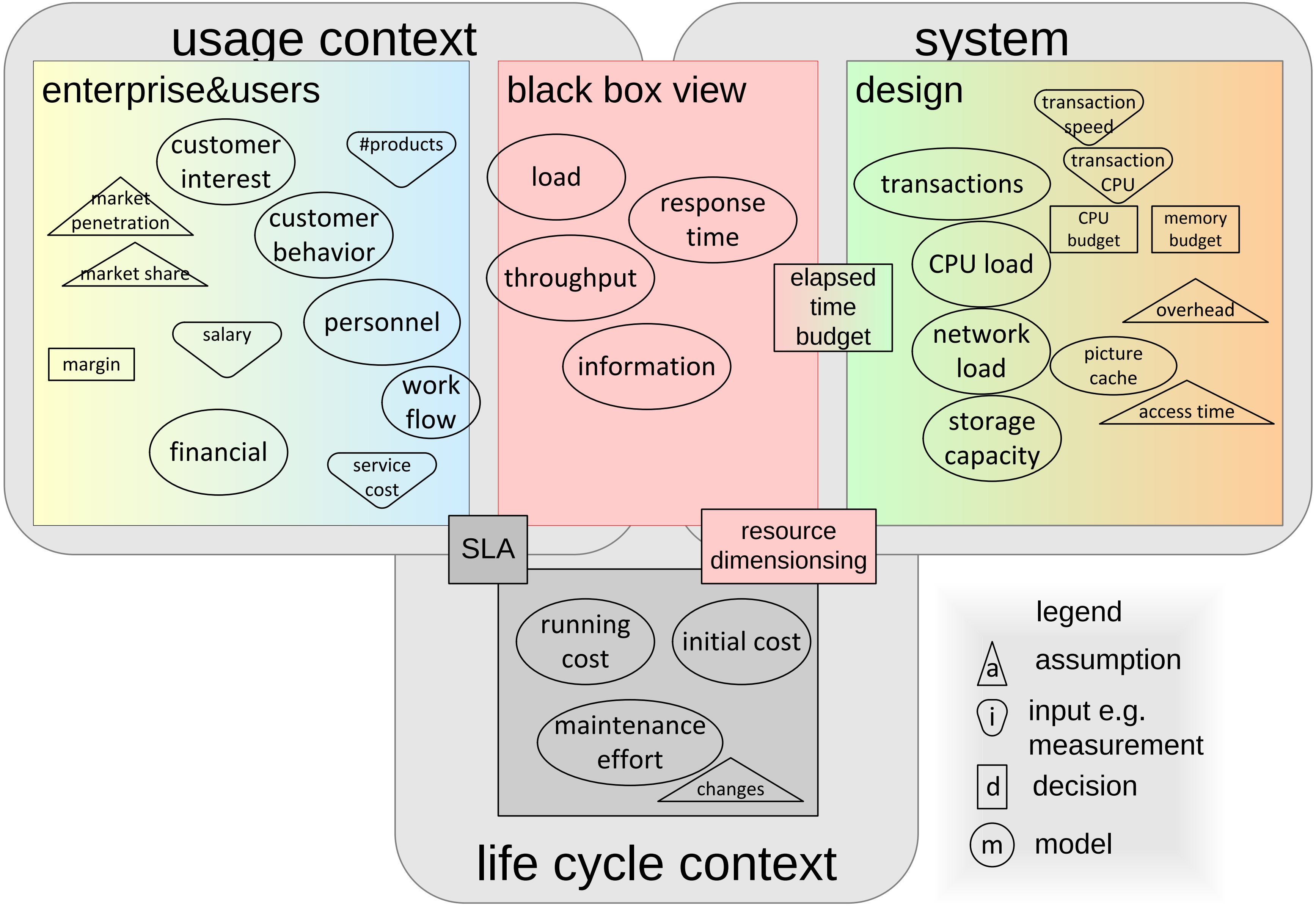
At what quality and complexity levels ?



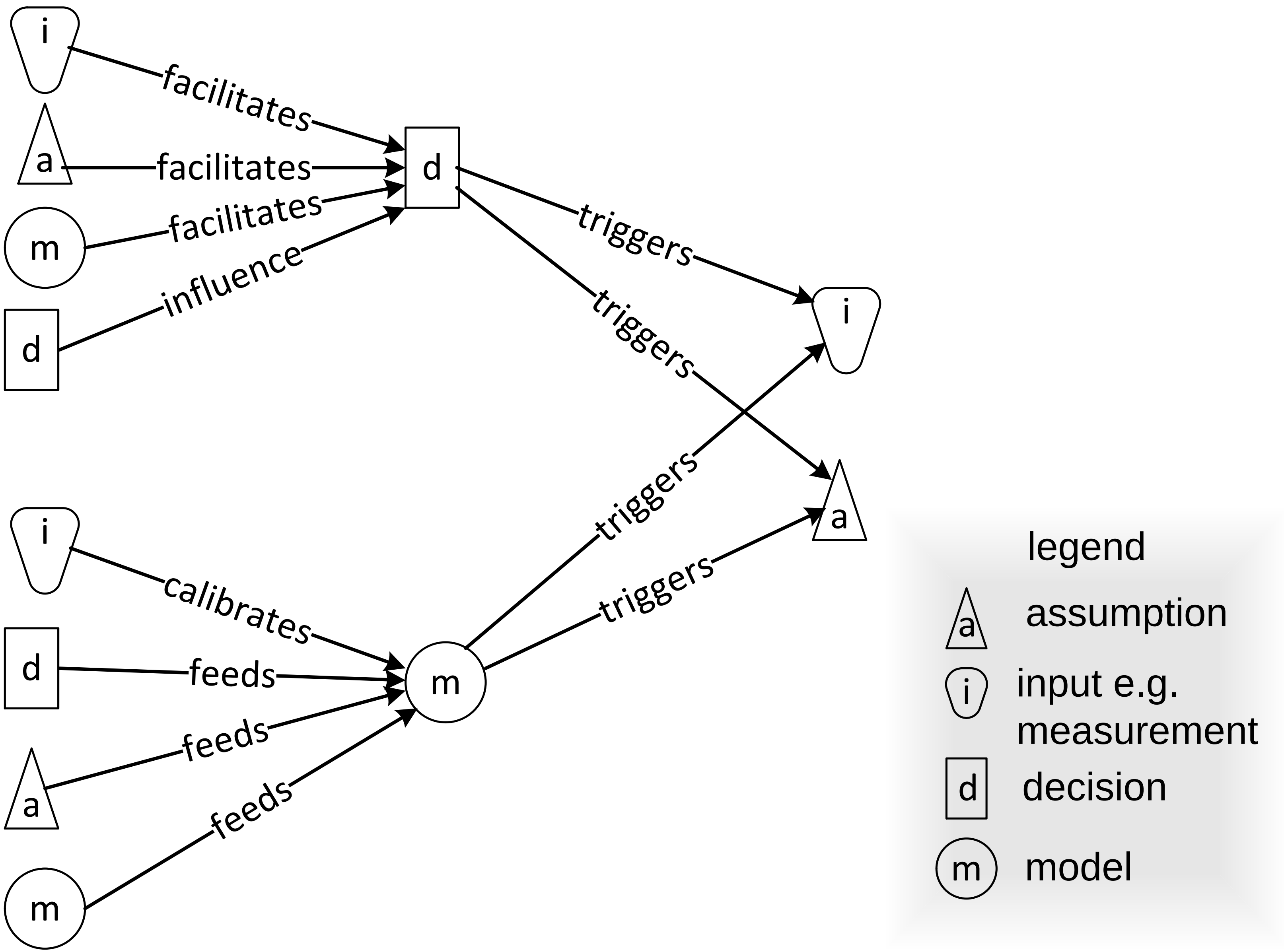
Graph of Decisions and Models



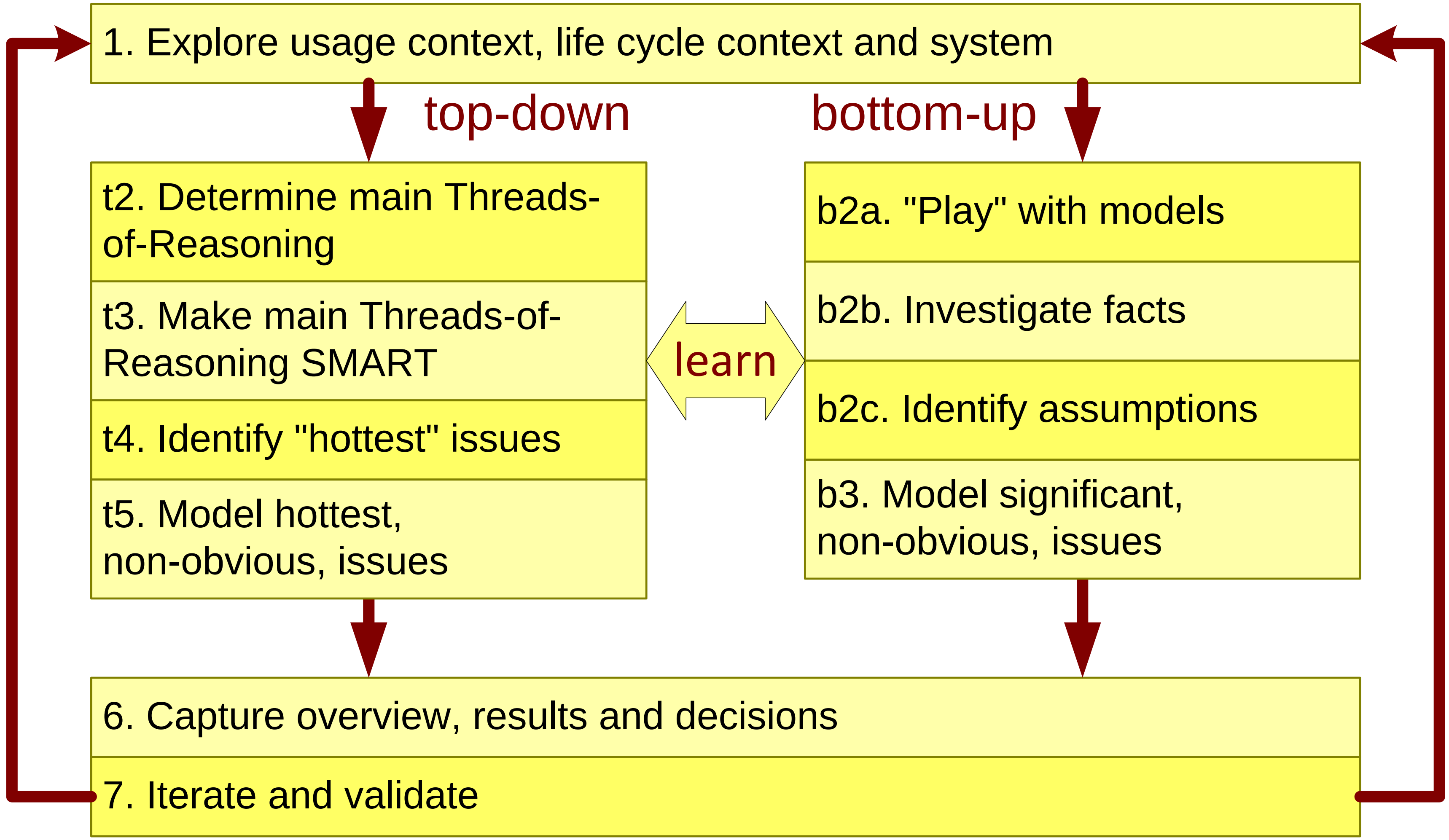
Example Graph for Web Shop



Relations: Decisions, Models, Inputs and Assumptions



Reasoning Approach



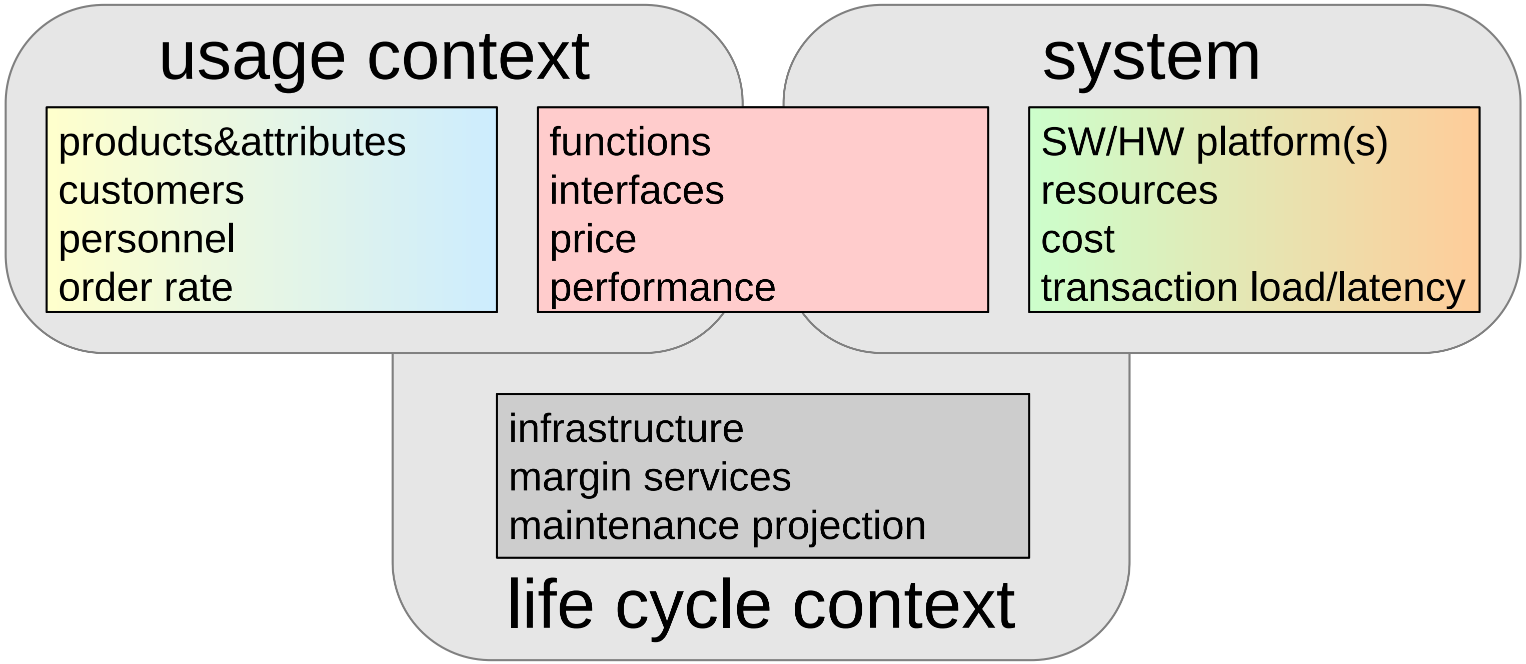
all steps time-boxed between 1 hour and a few days

early in project | later in project

1. Explore

1. Explore usage context, life cycle context and system

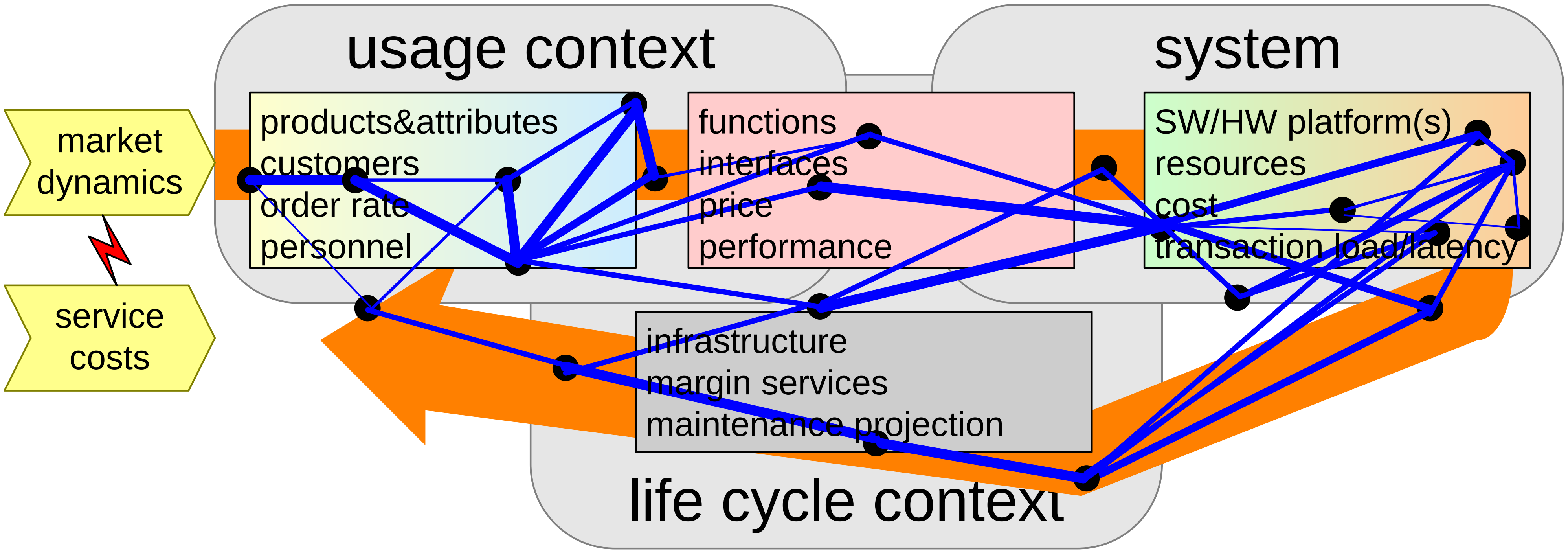
Populate with "known" facts, numbers, issues from preceeding projects, available work products and stakeholders



t2. Determine main Threads-of-Reasoning

Architecting and System Design

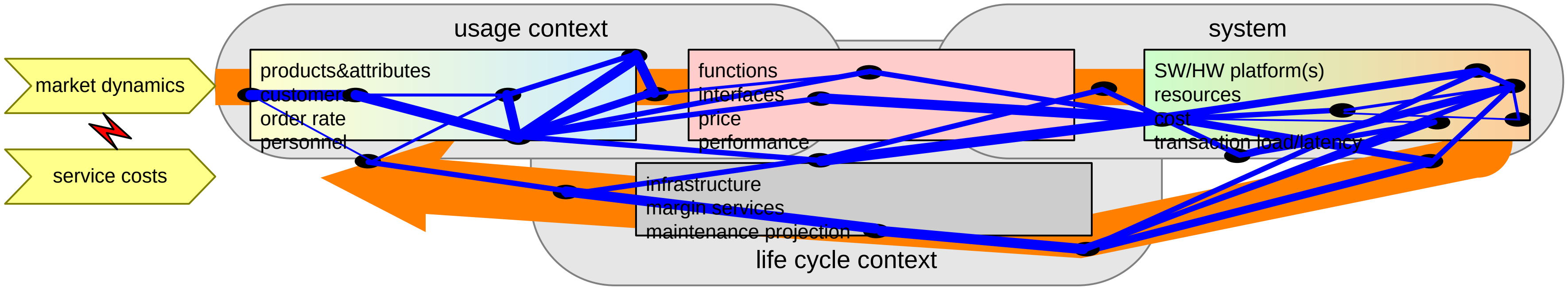
e.g. http://www.gaudisite.nl/ModuleTORSlides.pdf

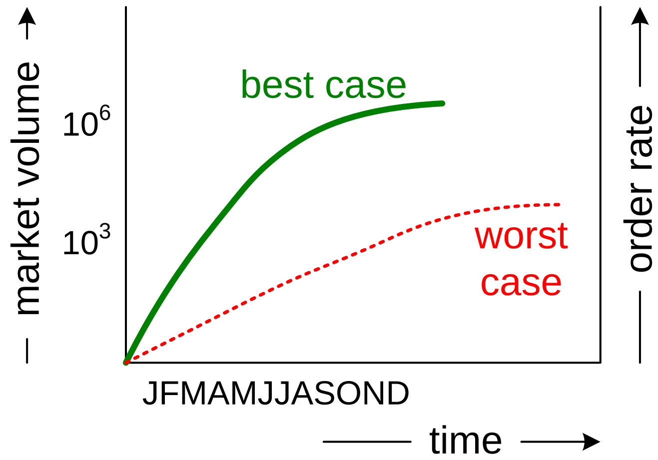


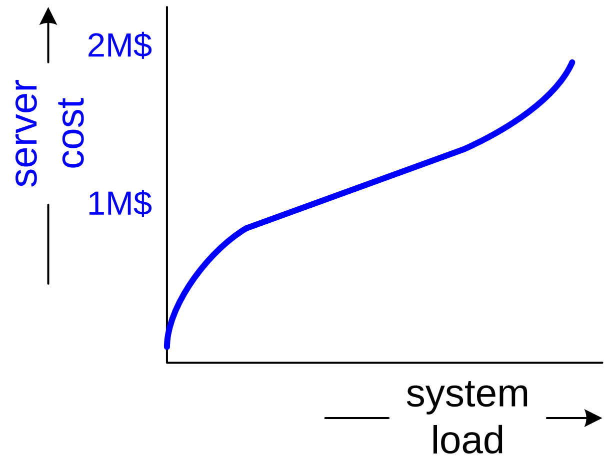
t3. SMART'en Thread-of-Reasoning

t3. Make main Threads-of-Reasoning SMART

Quantify in terms of Key Performance Indicators, Key Performance Measures, Critical Resources







Intermezzo: the acronym SMART

- Specific

quantified

- Measurable

verifiable

acronym consensus

- Assignable (Achievable, Attainable,
  Action oriented, Acceptable, Agreed-upon, Accountable)
- Realistic (Relevant, Result-Oriented)
- Time-related (Timely, Time-bound, Tangible, Traceable)

variation of meaning

t4: Identify Hottest

t4. Identify "hottest" issues

assess explored landscape:

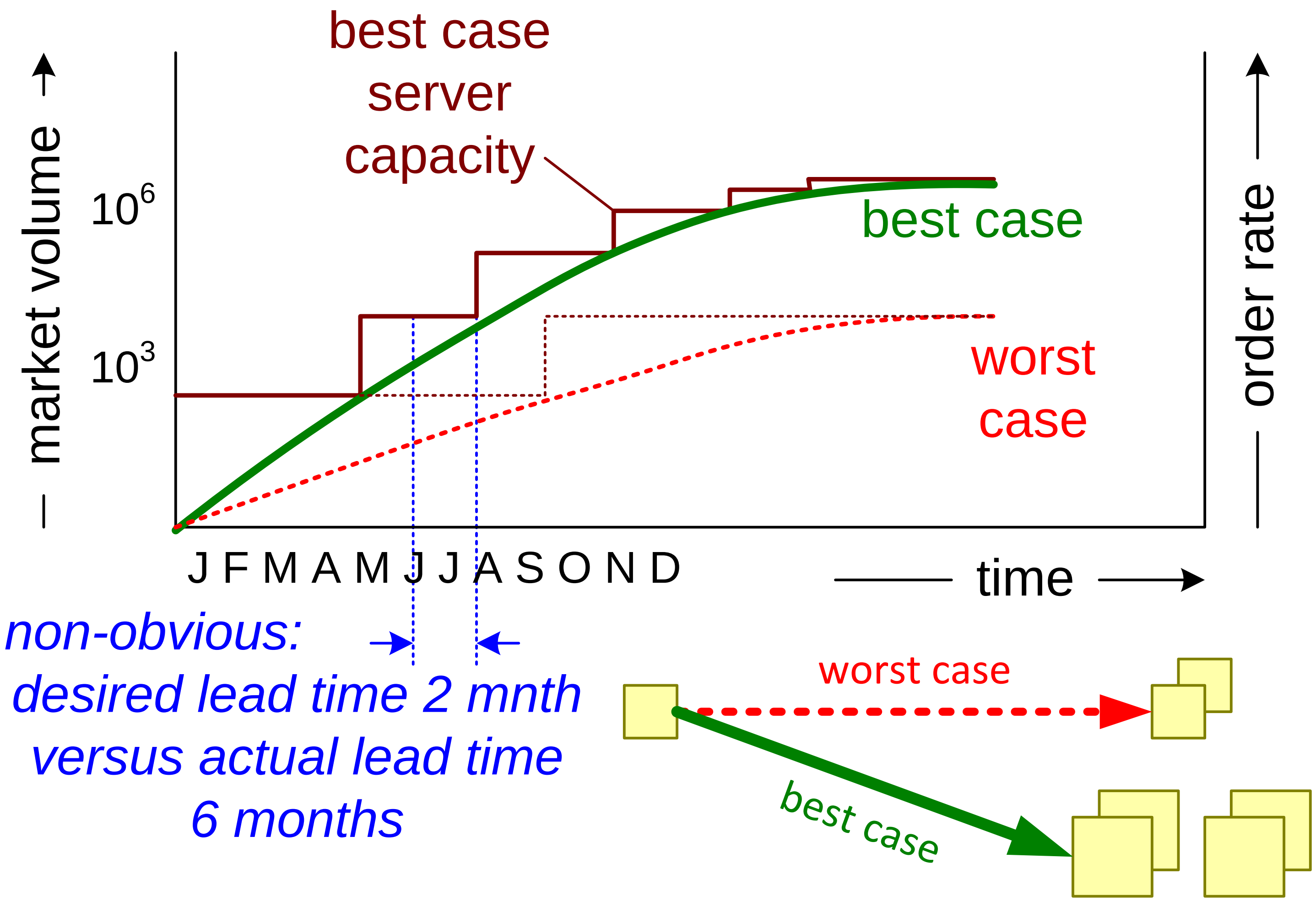
- highest (perceived) risk
- most important/valuable
- most discussed
- historic evidence
- urgency

rank issues according to aggregated assessment

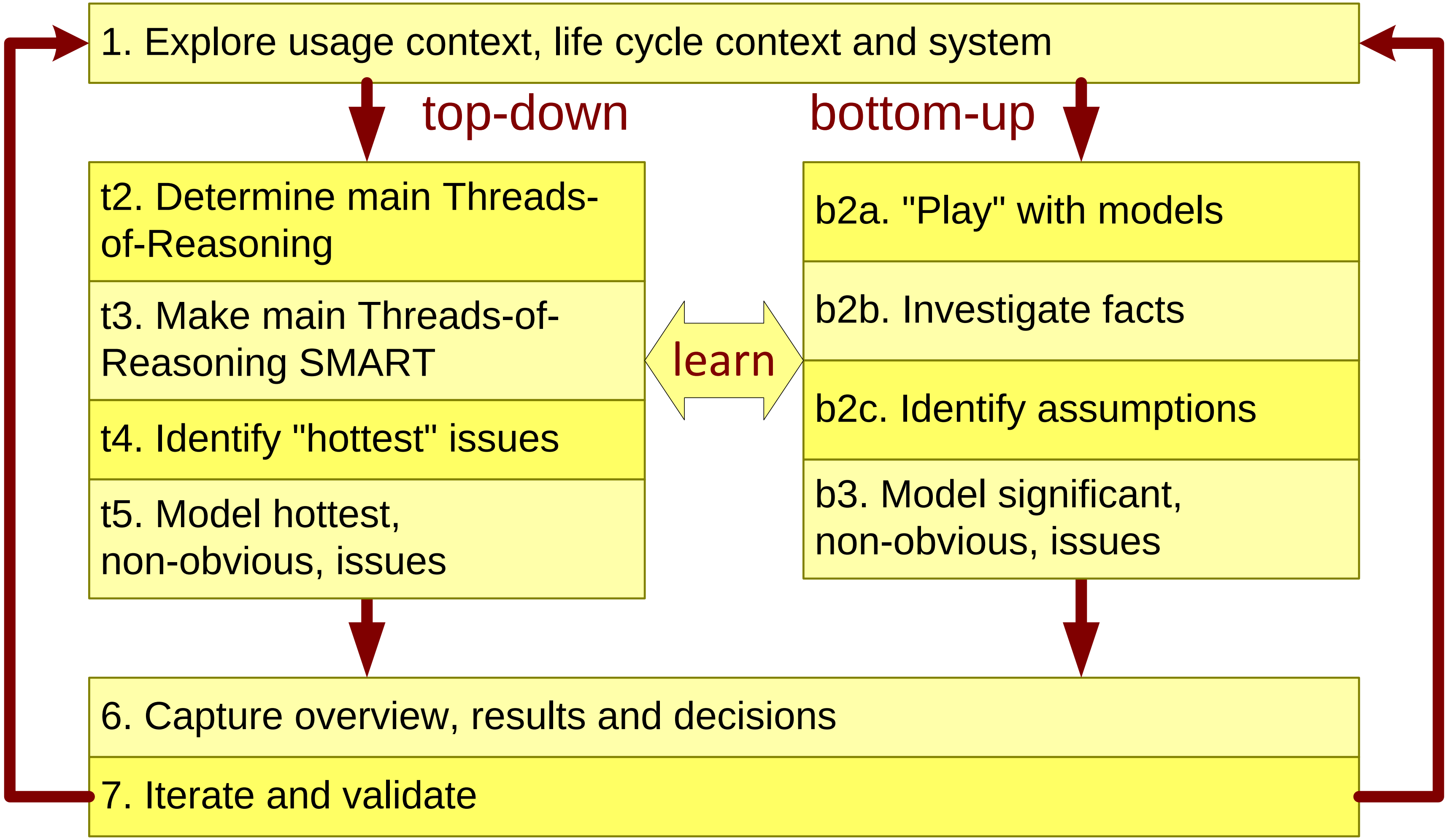
1..5 scale,
1 = low risk
5 = high risk
et cetera

	risk	value	discussion	history	urgency	ranking
server cost	2	3	2	1	3	
order rate	4	5	5	3	5	1
transactions	3	3	3	4	2	3
response time	3	5	1	4	2	2
availability	2	5	1	3	3	4
network bandwidth	1	1	3	1	3	
storage capacity	1	1	1	2	3	

t5. Model hottest, non-obvious, issues



From *top-down* to *bottom-up*



all steps time-boxed between 1 hour and a few days

early in project

later in project

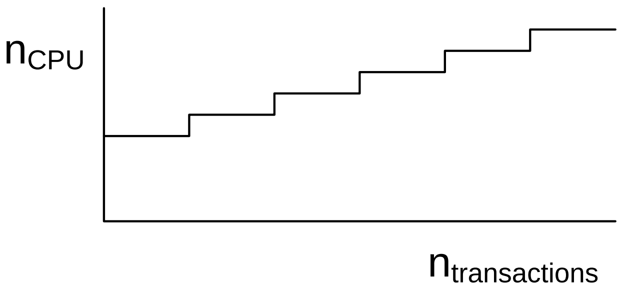
b2abc: Bottom-up

b2a. "Play" with models	b2b. Investigate facts	b2c. Identify assumptions
vary inputs vary model structure to understand *model applicability*, *design quality* and *specification feasibility*	market research measurements preceeding systems micro benchmarks literature, supplier info	What is the basis for model structure, design decision, specification, quantification et cetera? *Most assumptions are implicit and hidden!*

$n_{CPU} = t_{required\ total} / t_{1\ CPU}$

$t_{required\ total} = n_{transactions} * t_{1\ transaction} + t_{other}$

$t_{1\ transaction} = 1$ ms (on 1 CPU)



http://www.tpc.org/tpcc/results/tpcc_perf_results.asp

IBM System p5 595

TPC-C Throughput 4,033,378

Total # of Processors: 32

Total # of Cores: 64

$1/t_{1\ transaction} = 4*10^6 / 60 / 64$

min to sec

cores

$t_{1\ transaction} \sim= 1$ ms

server load dominated by transactions

transaction load scales linear

TPC-C is representative

what is the effect of other TPC-C workload?

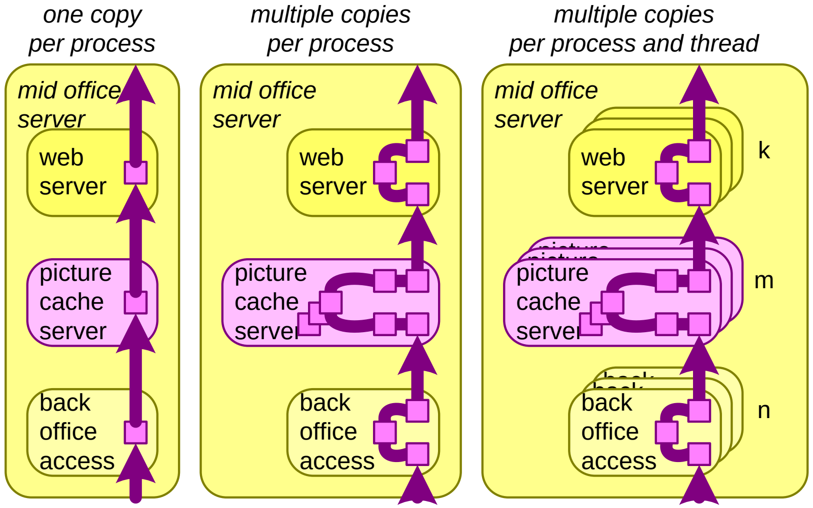
Bottom-up, more detailed steps

Make a list of technologies, components and resources to be used transactions, data base engine, memory, disk
Make a list of important qualities performance, reliability, security, maintainability
Make a characterization matrix of technologies, components and resources versus qualities 1..5 scale, 1 = low risk 5 = high risk et cetera
Perform step 2abc on most critical class 4 and 5 risks

b3: Model Significant Issues

b3. Model significant, non-obvious, issues

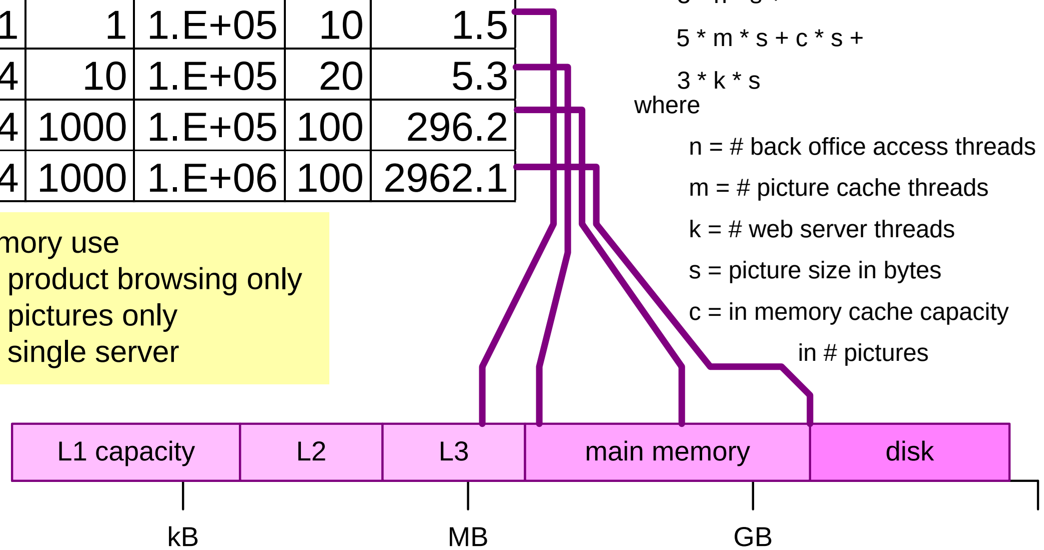
for example, memory use in server(s) for picture transfers and buffering



n	m	k	s	c	MB
1	1	1	1.E+05	10	1.5
2	4	10	1.E+05	20	5.3
2	4	1000	1.E+05	100	296.2
2	4	1000	1.E+06	100	2962.1

picture memory =
3 * n * s +
5 * m * s + c * s +
3 * k * s
where
n = # back office access threads
m = # picture cache threads
k = # web server threads
s = picture size in bytes
c = in memory cache capacity in # pictures

memory use
product browsing only
pictures only
single server



What is the performance impact of memory use on other processing?

Learning Concurrent Bottom-up and Top-down

top-down ⟷ learn ⟷ bottom-up

top-down: what is *hidden* in details?

top-down: do we address the *relevant* decomposition?

bottom-up: do we address relevant details?

bottom-up: what details have *significant* impact?

order rate
(and evolution over time)
is highly speculative

needs input

induces risk

transactions
dominate
server load&cost

Example top-down and bottom-up

top-down:

what is impact of

catalogue size and changes?

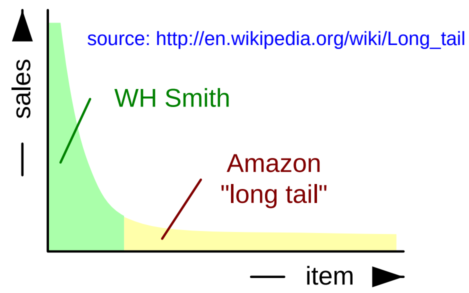
bottom-up:

what is relevant concurrency (k), cache size (c), or picture size (s)?

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



	n	m	k	s	c	MB	storage type
	1	1	1	100,000	10	1.5	L3
	2	4	10	100,000	20	5.3	main
highly concurrent	2	4	1000	100,000	100	296	main
large pictures	2	4	1000	1,000,000	100	2,962	main+disk
many pictures	2	4	10	100,000	100,000	9,540	main+disk
all at once	2	4	1000	1,000,000	100,000	98,234	disk

picture memory = 3 * n * s + 5 * m * s + c * s +3 * k * s
where

n = # back office access threads

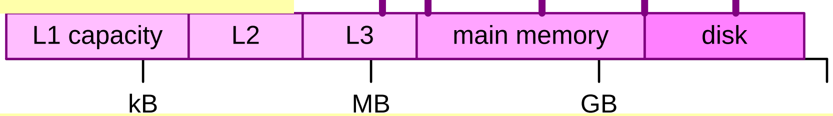
m = # picture cache threads

k = # web server threads

s = picture size in bytes

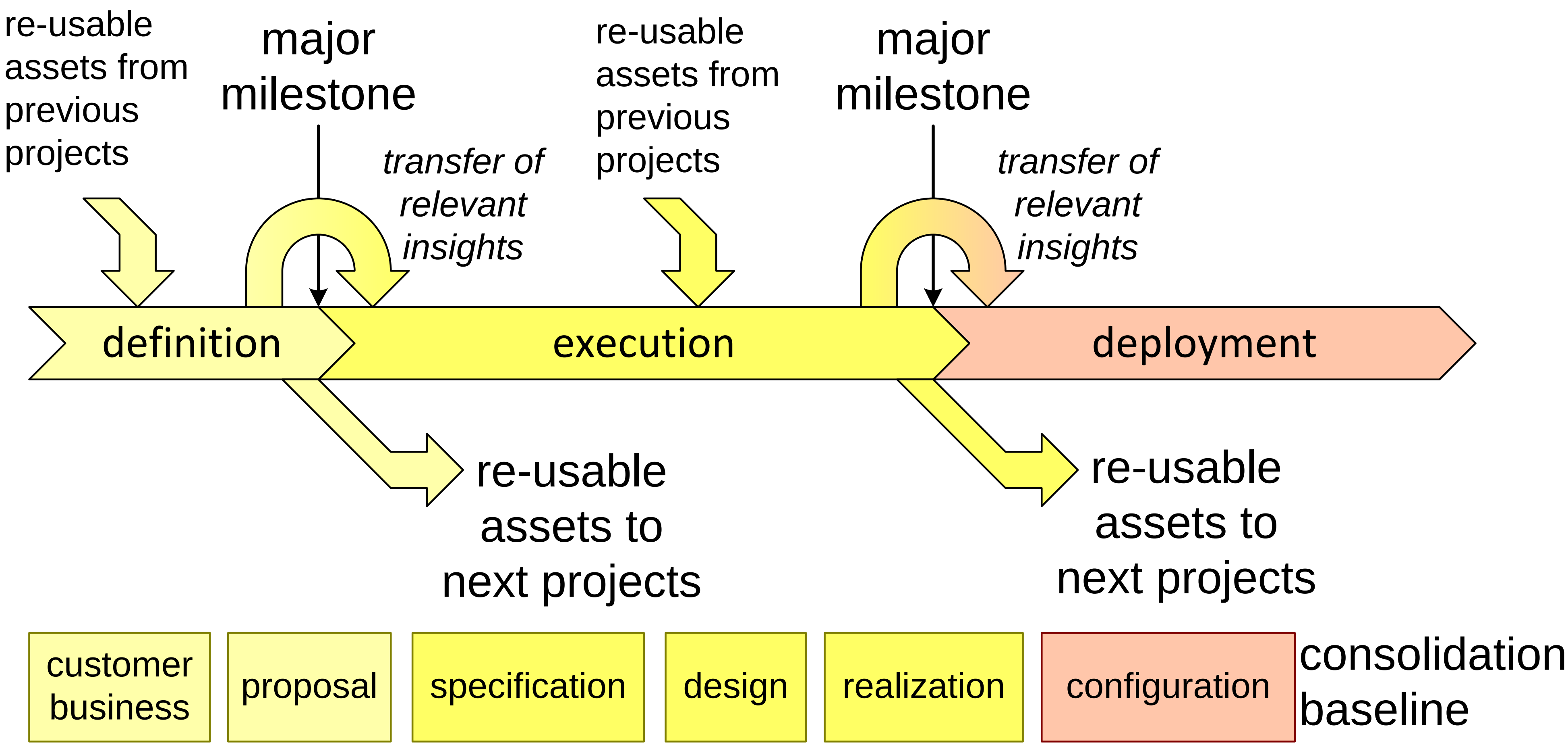
c = in memory cache capacity in # pictures

memory use
product browsing only
pictures only
single server

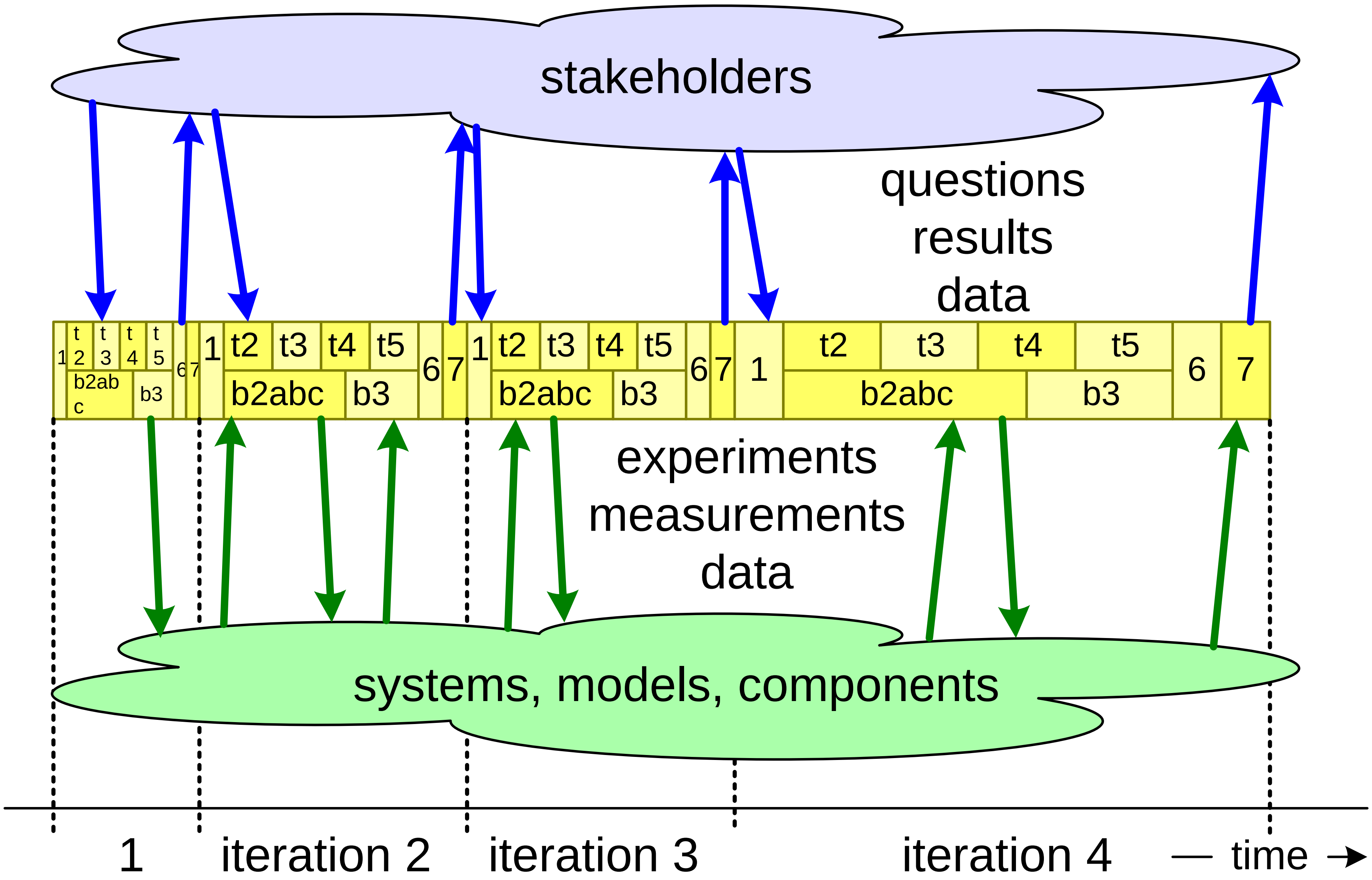


What is the performance impact of memory use on other processing?

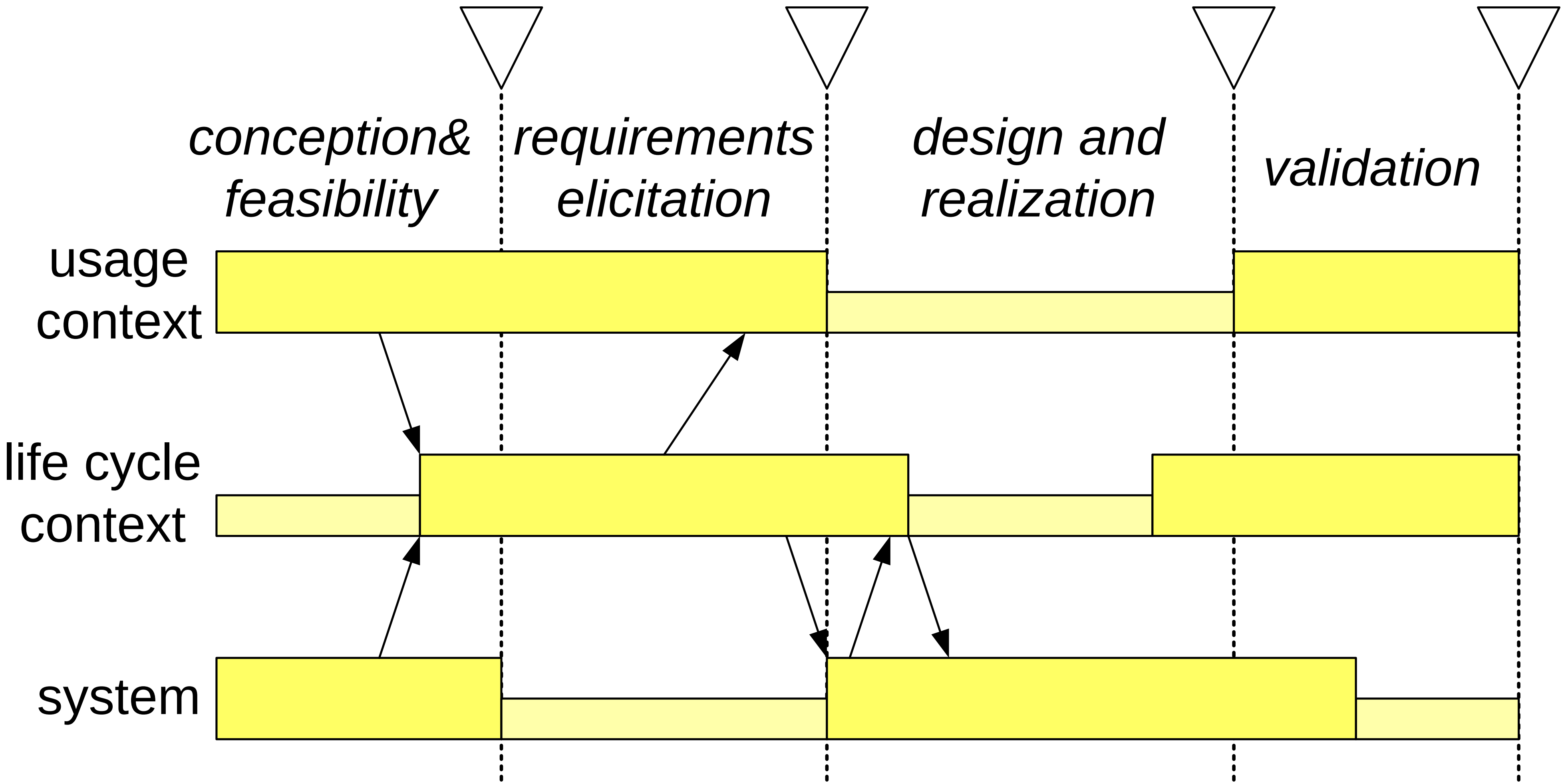
6. Capture overview, results and decisions



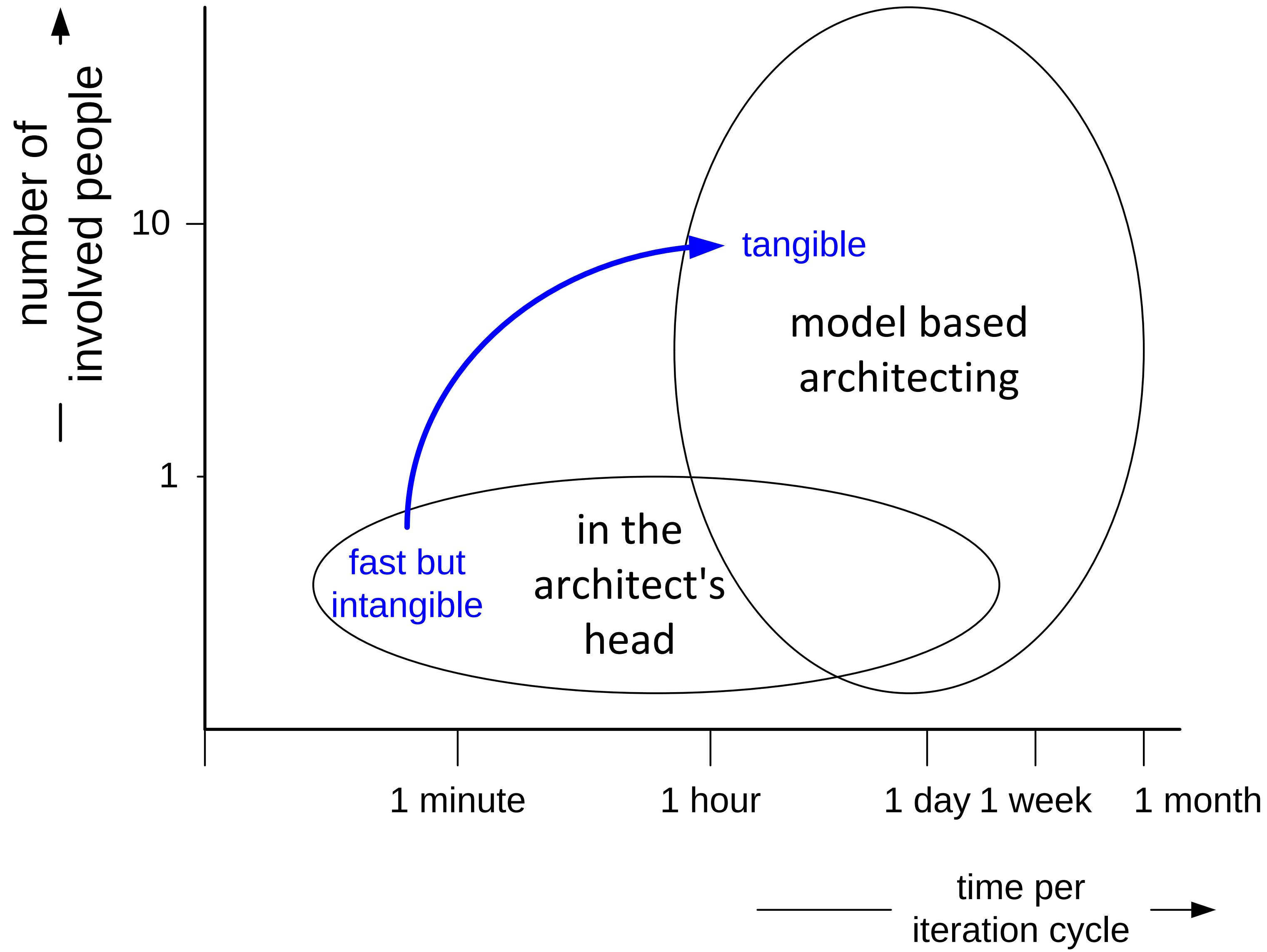
7. Iterate and Validate



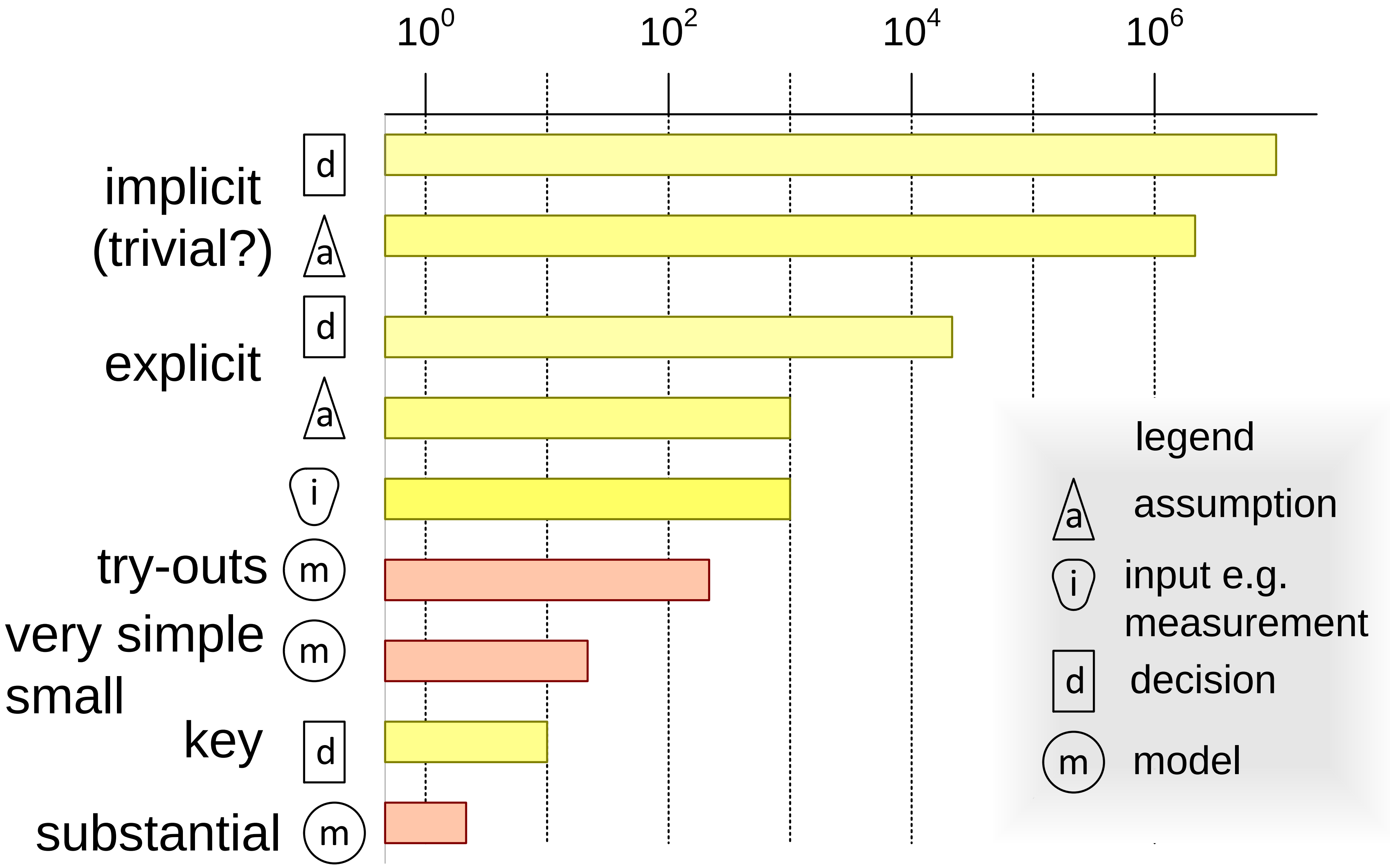
Focus is Shifting during Project



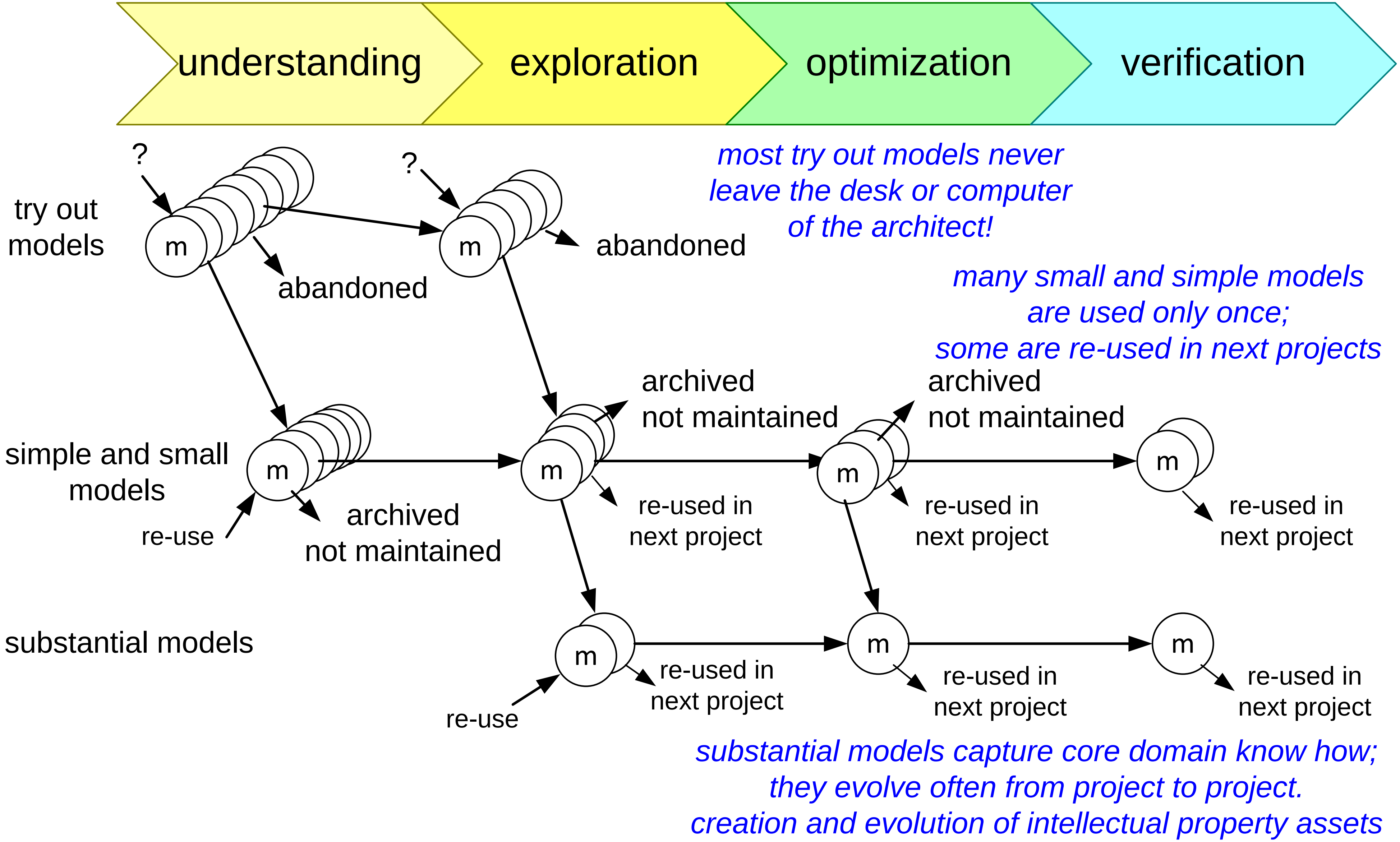
Models Support Communication



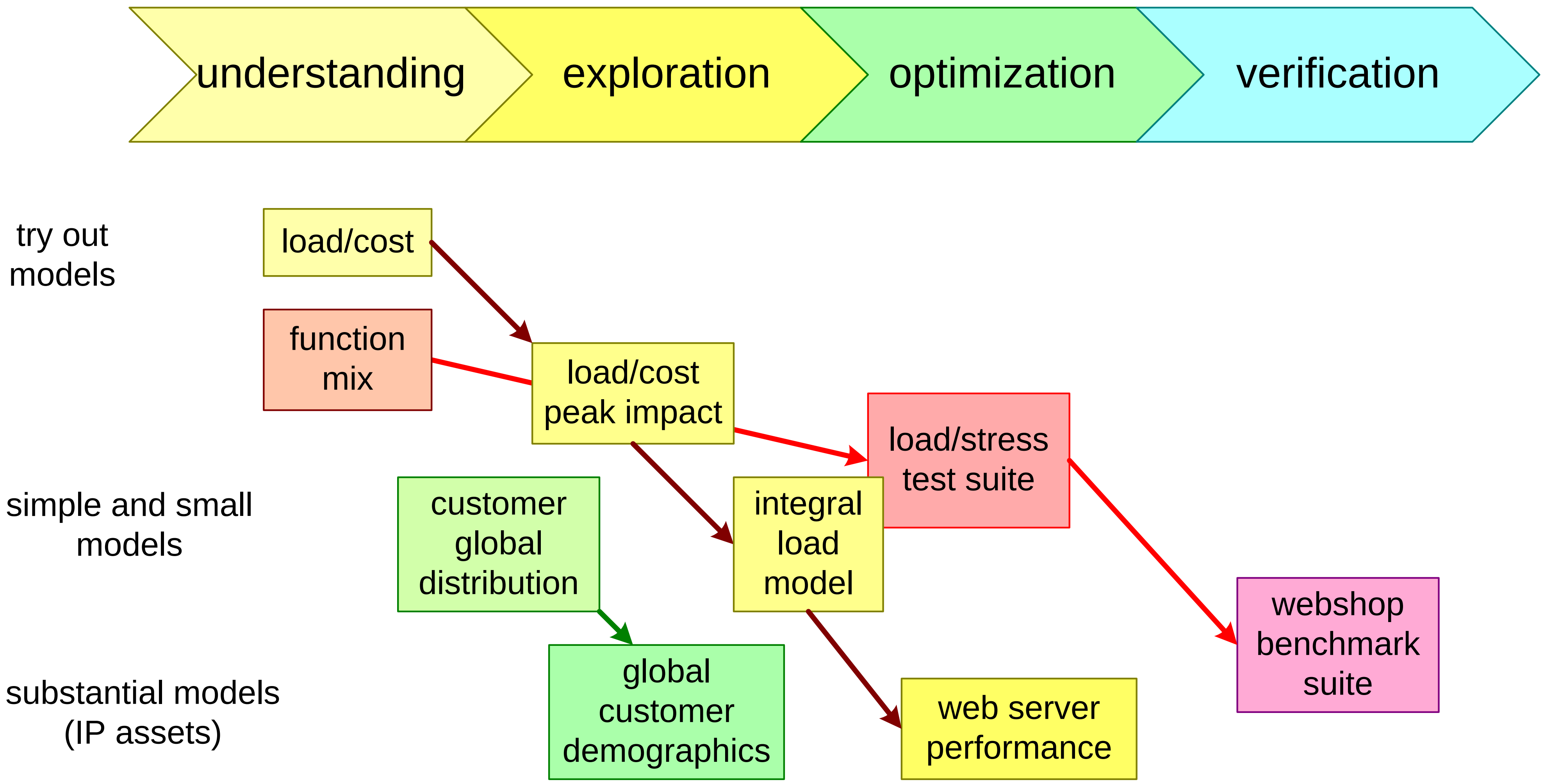
Frequency of Assumptions, Decisions and Modeling



Life Cycle of Models



Examples of Life Cycle of Models



Conclusions

Top-down and bottom-up provide complementary insights

Key words for selection: hottest, non-obvious, significant, relevant

Multiple small models are used in combination

Some models evolve from very simple to more substantial

Techniques, Models, Heuristics of this module

Threads-of-reasoning

SMART

Key Performance Indicators, Key Performance Measures, Critical Resources

Ranking matrices

Modeling and Analysis: Modularity and Integration

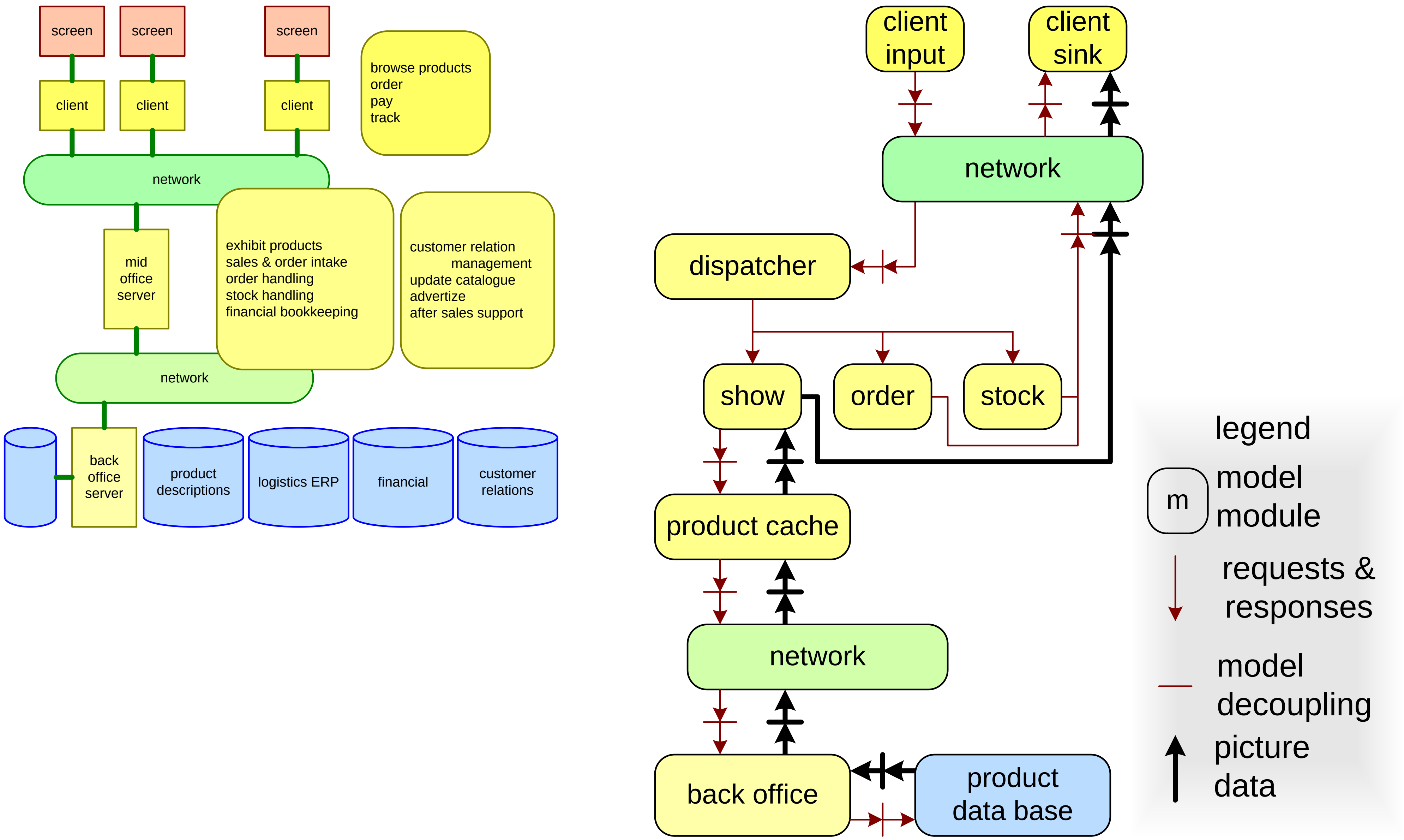
by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract
More substantial models are created step by step. We will discuss the order of creation and modularity considerations. The modules have to be integrated into the desired substantial model.

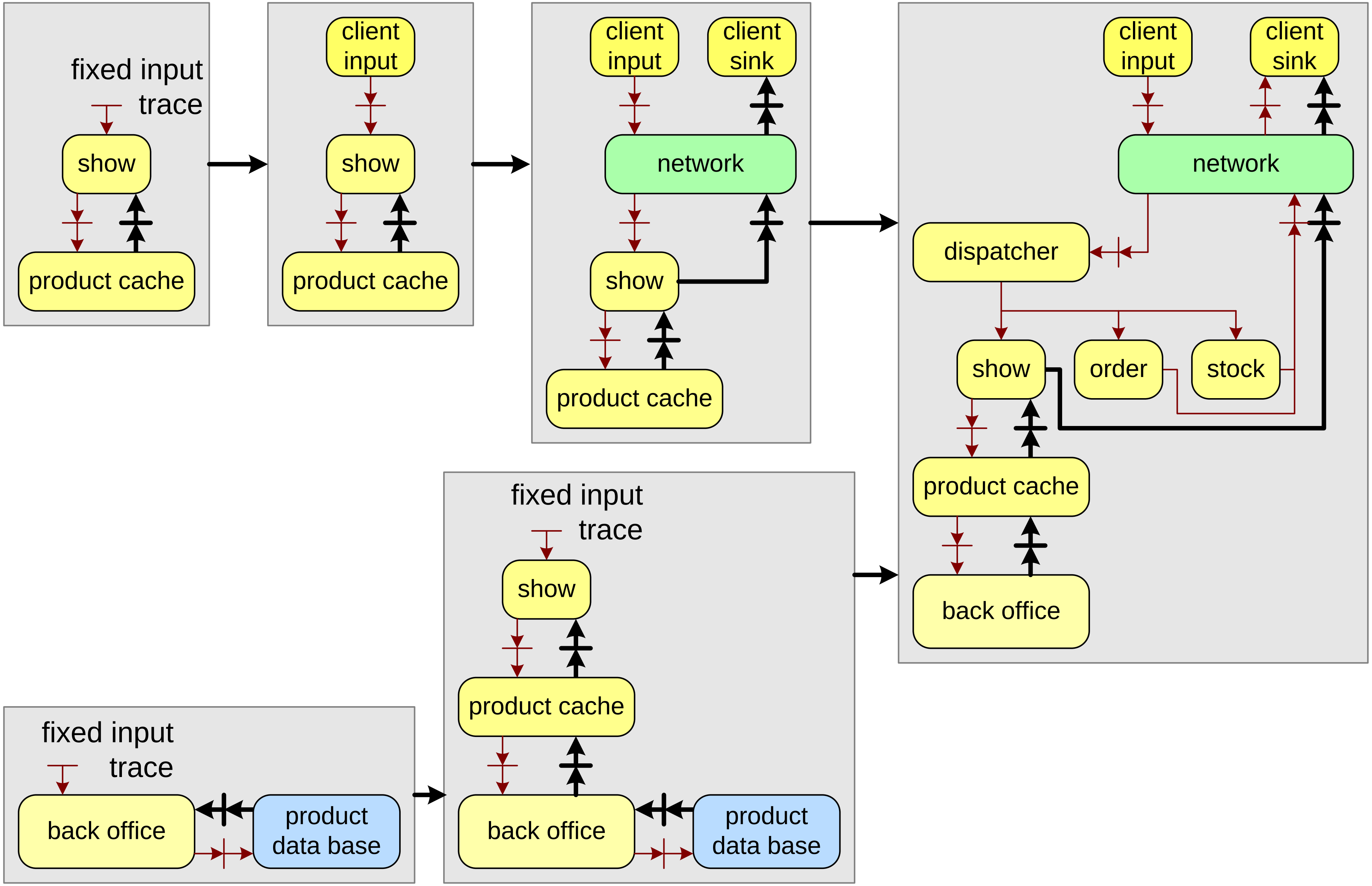
July 3, 2023
status: planned
version: 0

logo
TBD

Example of (Partial) Flow Simulator



Example of Incremental Model Creation



Approach for Incremental Model Creation

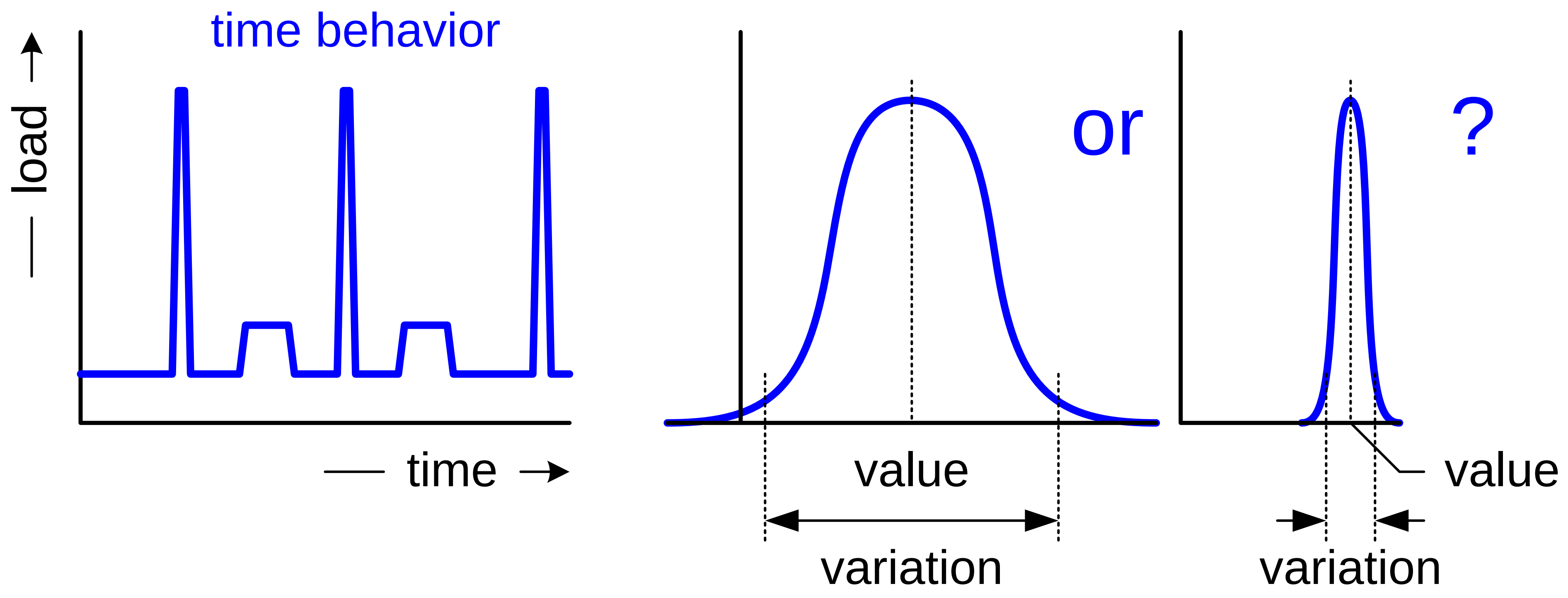
Start with the hottest issue	what creates the most discussion or uncertainty?
Ensure immediate feedback	does this model help to answer the questions that we have?
Keep flexible decoupling point	e.g. human readable/editable files
Extend model only for a good purpose	don't integrate models because it *can* be done
Create effective visual outputs	simple animations, graphs, tables, ...
Refactor regularly	based on increasing insight, feedback and purpose

Attention Points for Every Integration Step

Does the output of the integrated model match your expectation?

Can you explain the model behavior?

Can you explain the variation of the output?



Module Modeling and Analysis: Analysis and Using Models

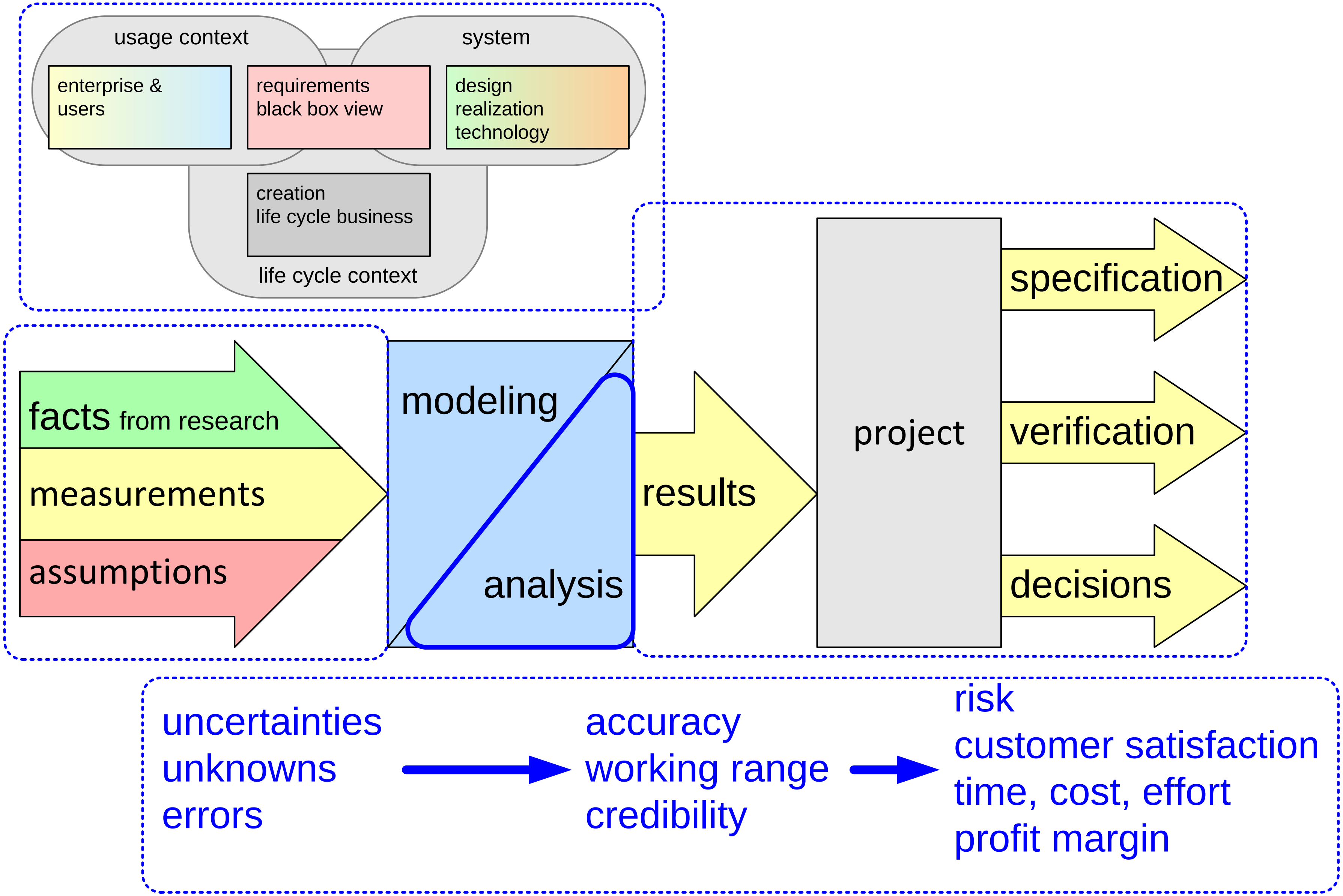
by *Gerrit Muller* HSN-NISE
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

This module addresses the analysis of models and discusses how to use models.

July 3, 2023
status: planned
version: 0.1

Where are we in the Course?



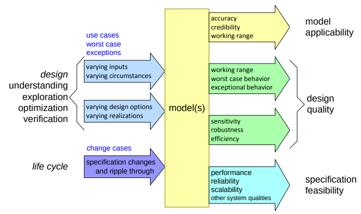
Modeling and Analysis: Analysis

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

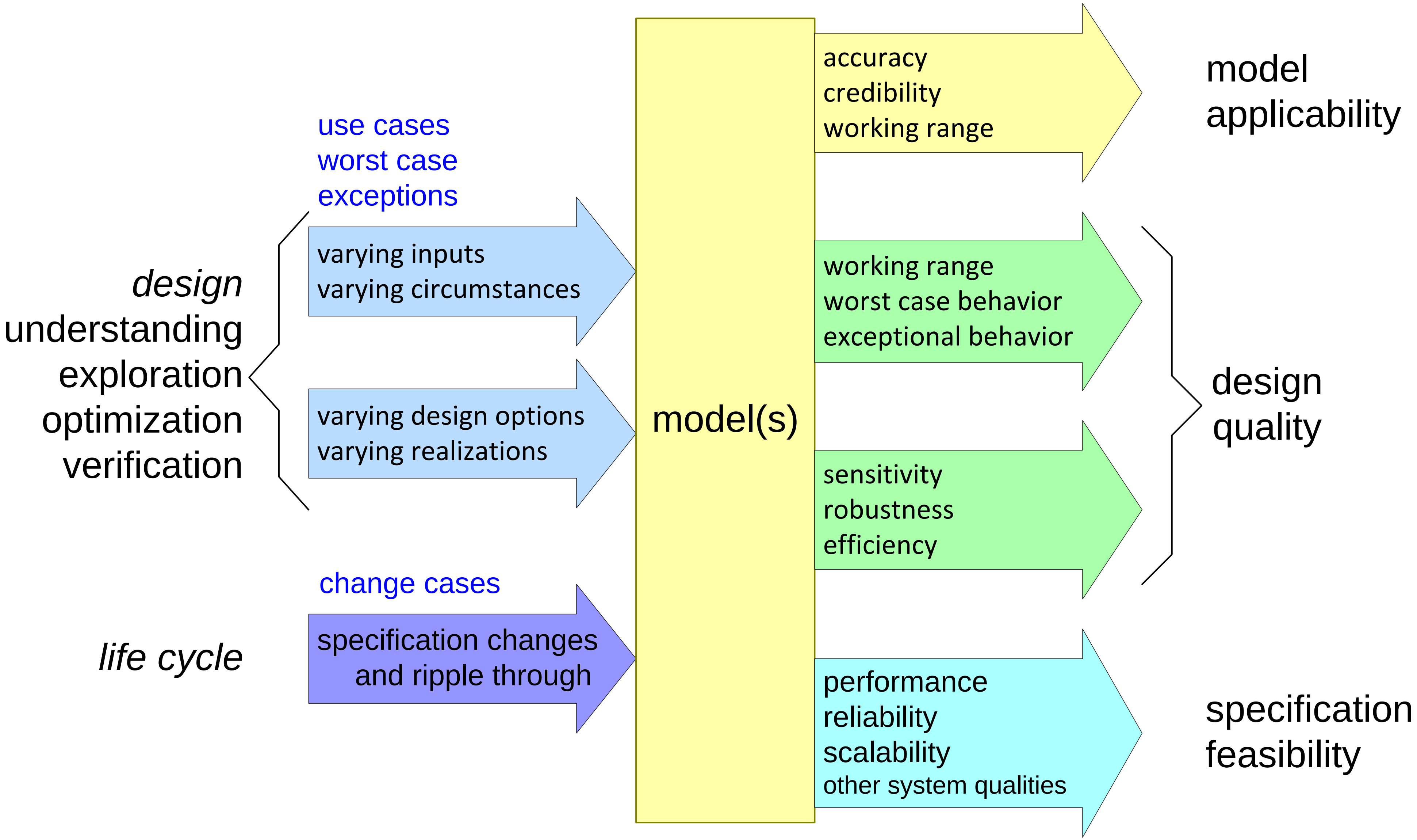
Abstract

Models only get value when they are actively used. We will focus in this presentation on analysis aspects: accuracy, credibility, sensitivity, efficiency, robustness, reliability and scalability.

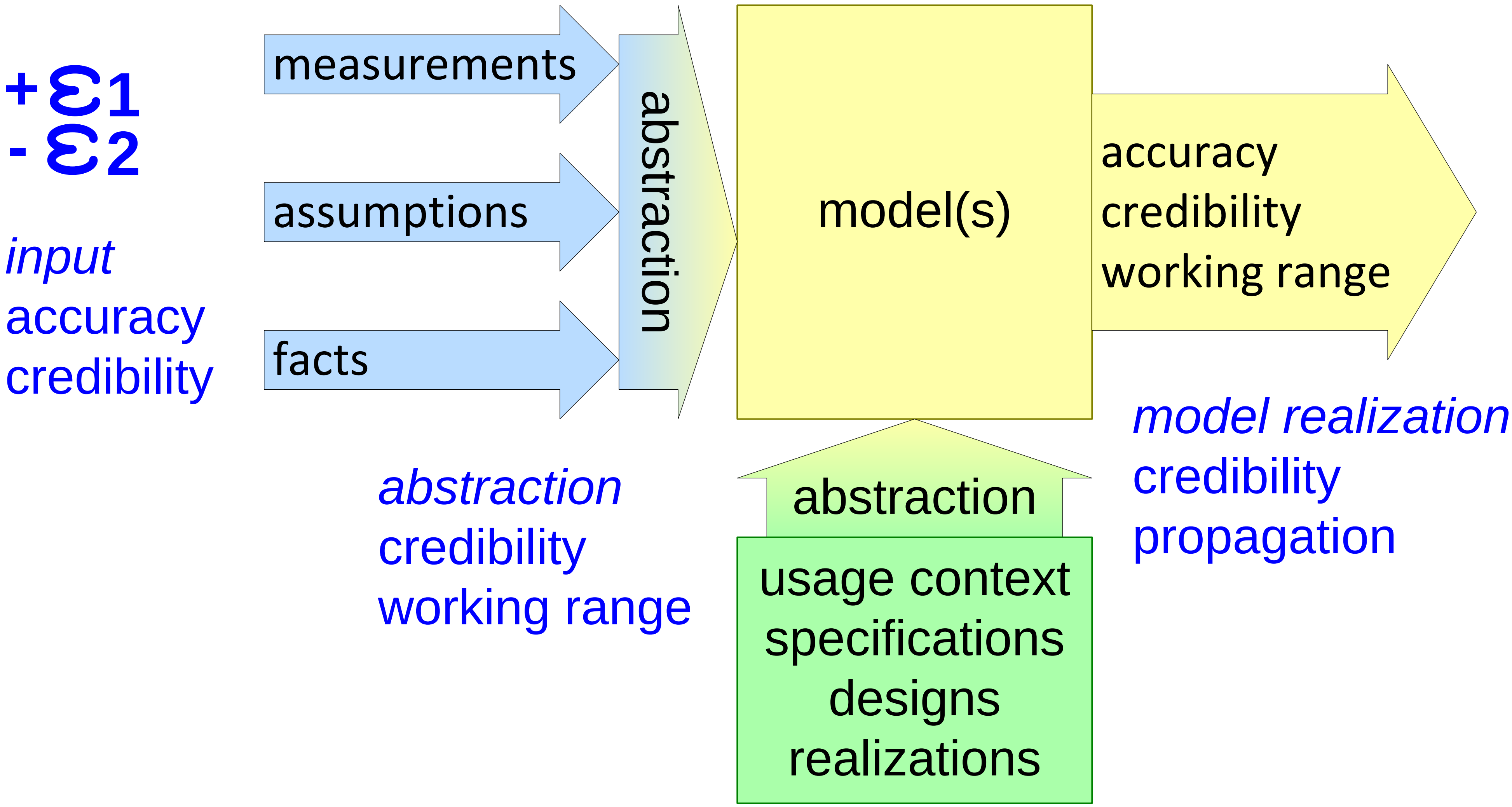
July 3, 2023
status: planned
version: 0.2



What Comes out of a Model



Applicability of the Model



How to Determine Applicability

try out models

be aware of accuracy, credibility and working range

simple and small models

1. Estimate accuracy of results

   based on most significant inaccuracies of inputs and assumed model propagation behavior

2. Identify top 3 credibility risks

   identify biggest uncertainties in inputs, abstractions and realization

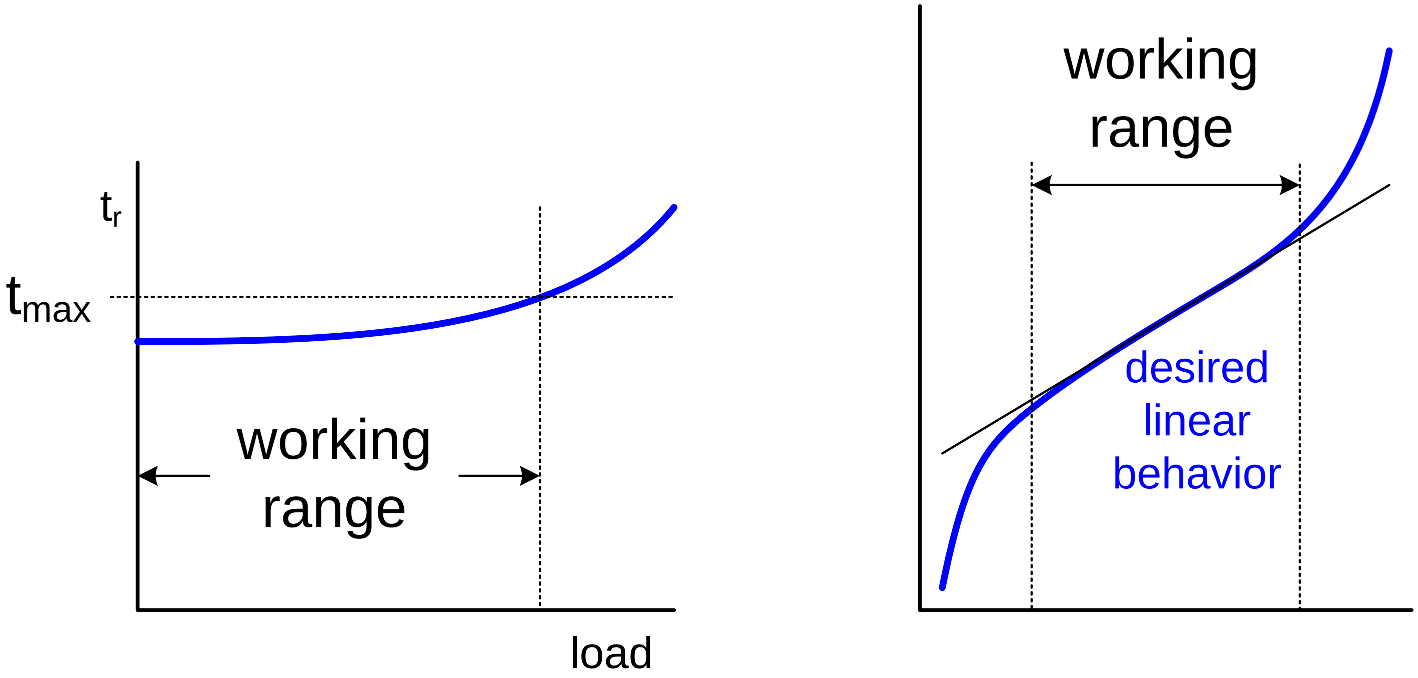
3. Identify relevant working range risks

   identify required (critical) working ranges and compare with model working range

substantial models

systematic analysis and documentation of accuracy, credibility and working range

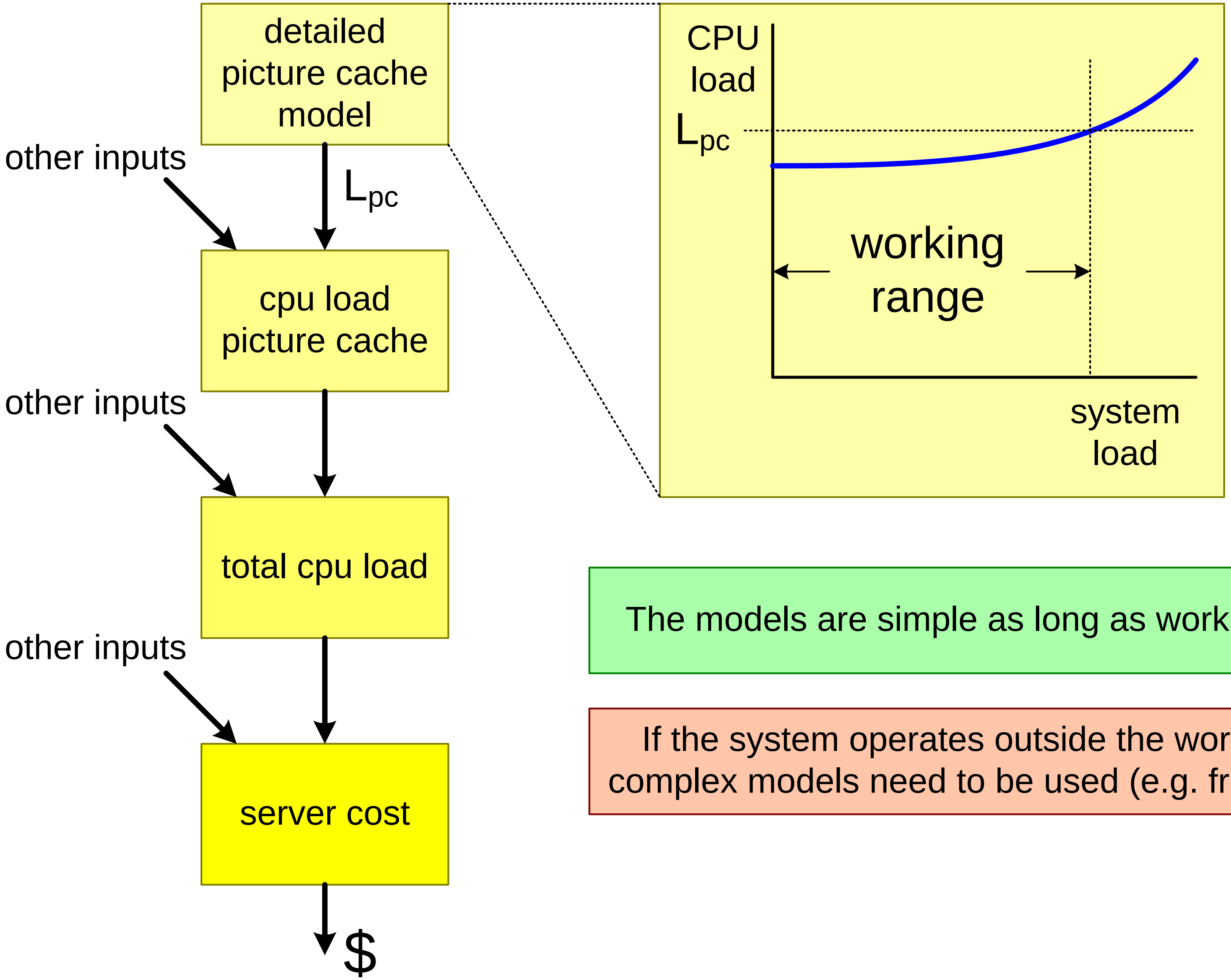
Working Range examples



A system design assumption is often:
the performance of this function
{ is constant | is linear | doesn't exceed x | ...}

The working range is the interval where this assumption holds

Example of Picture Cache Working Range



The models are simple as long as working ranges are obeyed

If the system operates outside the working range then more complex models need to be used (e.g. from 0^e order to 1^e order)

Common Pitfalls

discrete events in continuous world

discretization artefacts
e.g. stepwise simulations

(too) systematic input data

random data show different behavior
e.g. memory fragmentation

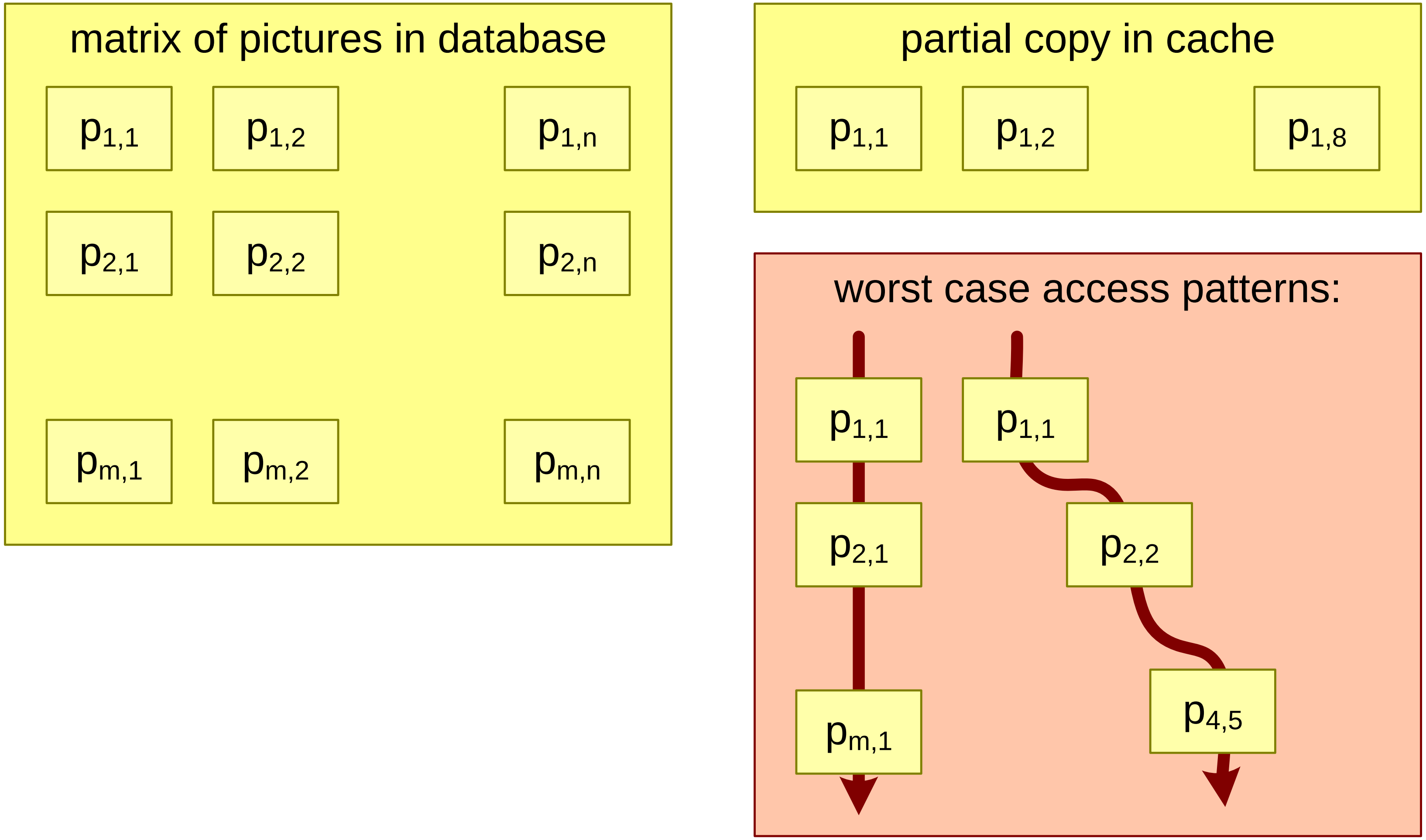
fragile model

small model change results in large shift in results

self fulfilling prophecy

price erosions + cost increase (inflation) -> bankruptcy

Example of Worst Case Picture Cache



What is the system behavior and performance for worst case access patterns?

Worst Case Questions

Which design assumptions have a big impact on system performance?

What are the worst cases for these assumptions?

How does the system behave in the worst case?

a. poor performance within spec

b. poor performance not within spec

c. failure -> reliability issue

FMEA-like Analysis Techniques

	(systematic) brainstorm	analysis and assessment probability severity propagation	improve spec, design, process, procedure, ...
safety hazard analysis	potential hazards	damage	measures
reliability FMEA	failure modes exceptional cases	effects	measures
security	vulnerability risks	consequences	measures
maintainability	change cases	impact, effort, time	decisions
performance	worst cases	system behavior	decisions

Brainstorming Phases

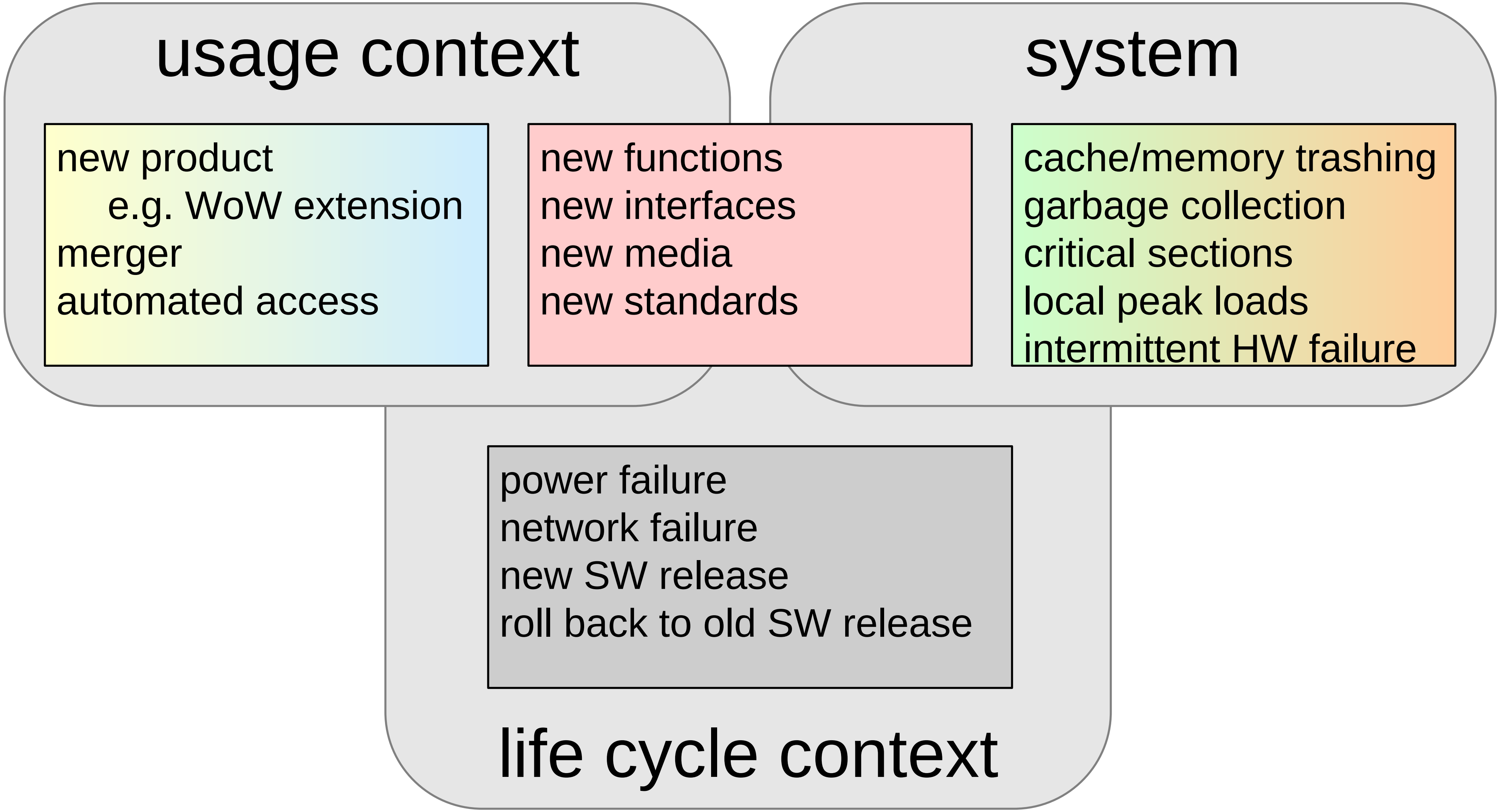
wave 1: the obvious

wave 2: more of the same

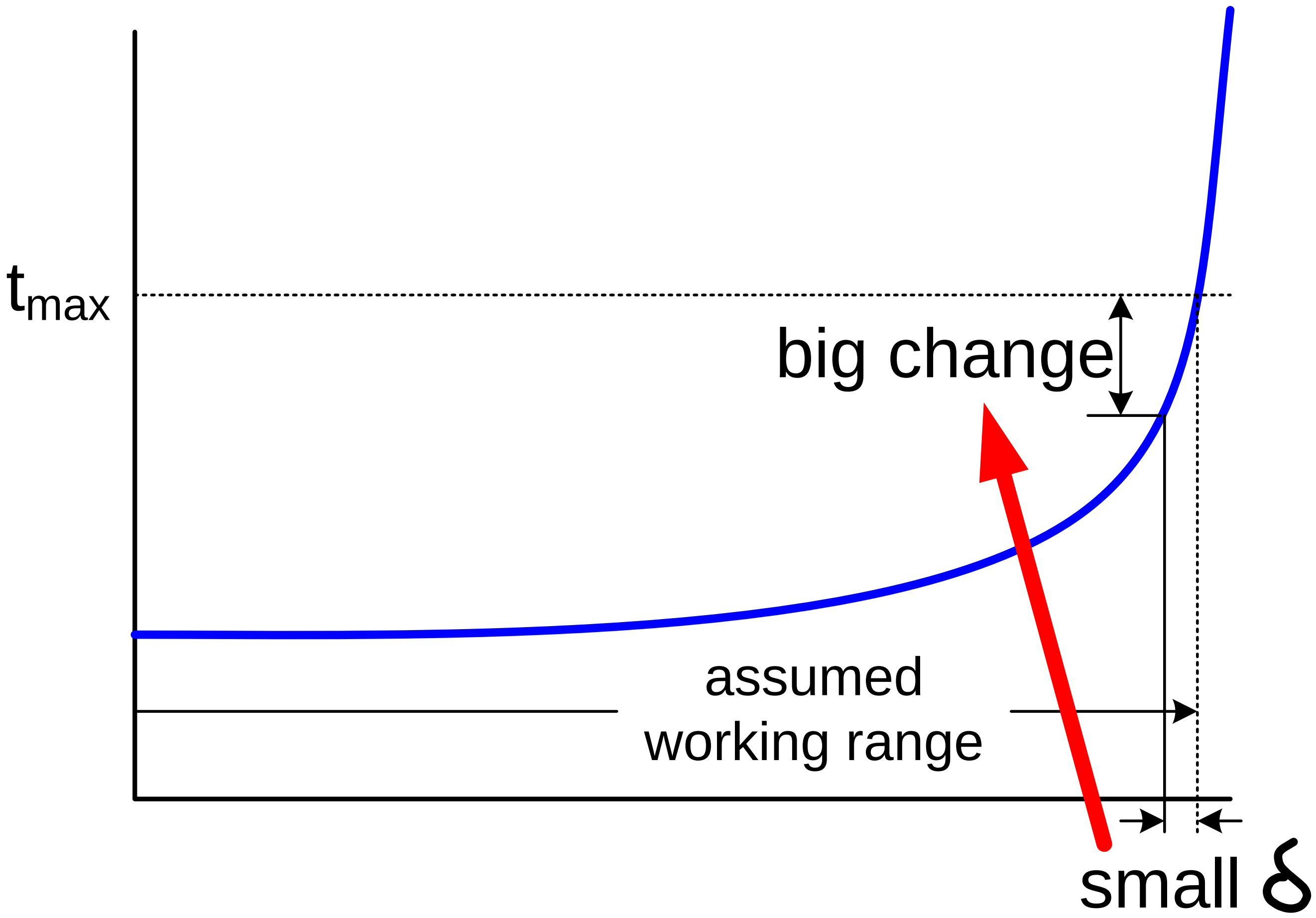
wave 3: the exotic, but potentially important

don't stop too early with brainstorming!

Different Viewpoints for Analysis



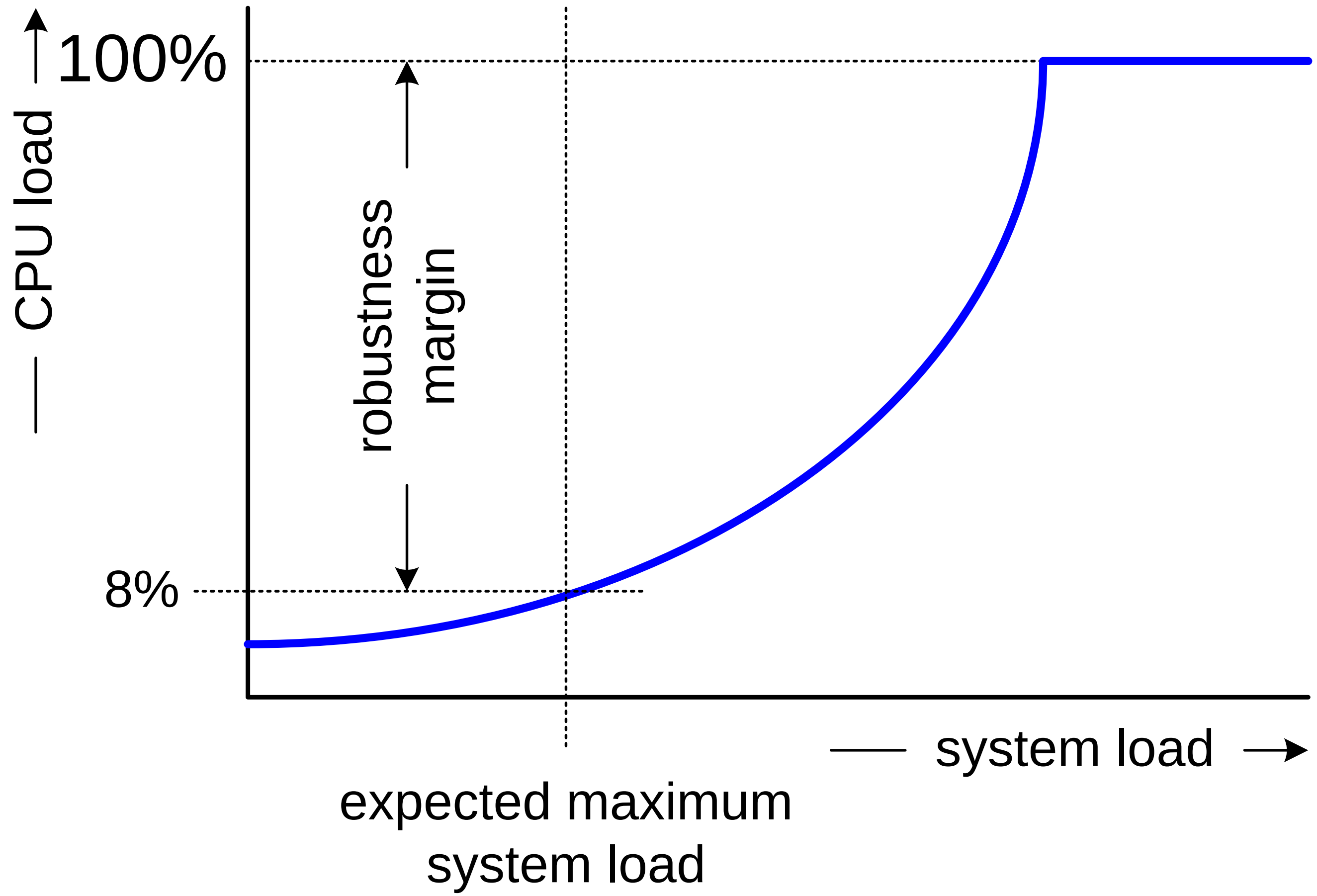
Example Sensitivity



sensitivity: how sensitive is the system output for small changes in input or realization?

Example of CPU Utilization and Efficiency

CPU utilization is "only" 8%
what is the efficiency?



Efficiency is Context Dependent!