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

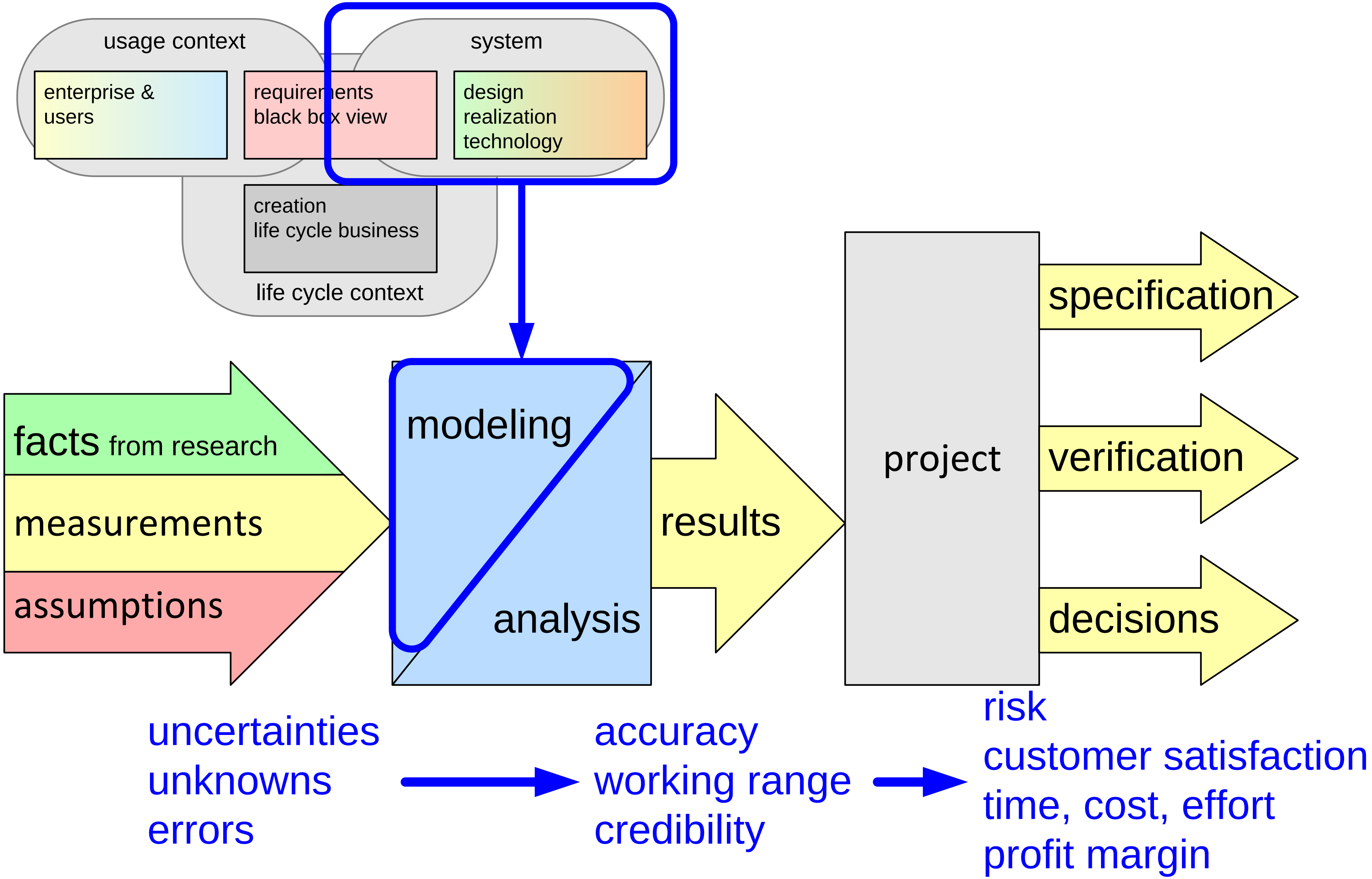
zero order and first order load models

budgeting

exercise

model one NFR in relation to a critical technology choice

Where are we in the Course?



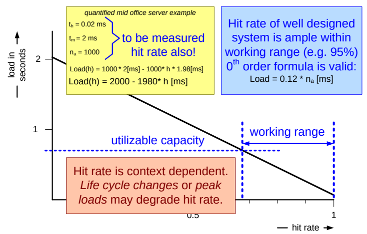
Modeling and Analysis: System Model

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.

July 3, 2023
status: preliminary draft
version: 0.4



Content of Modeling and Analysis System Model

content

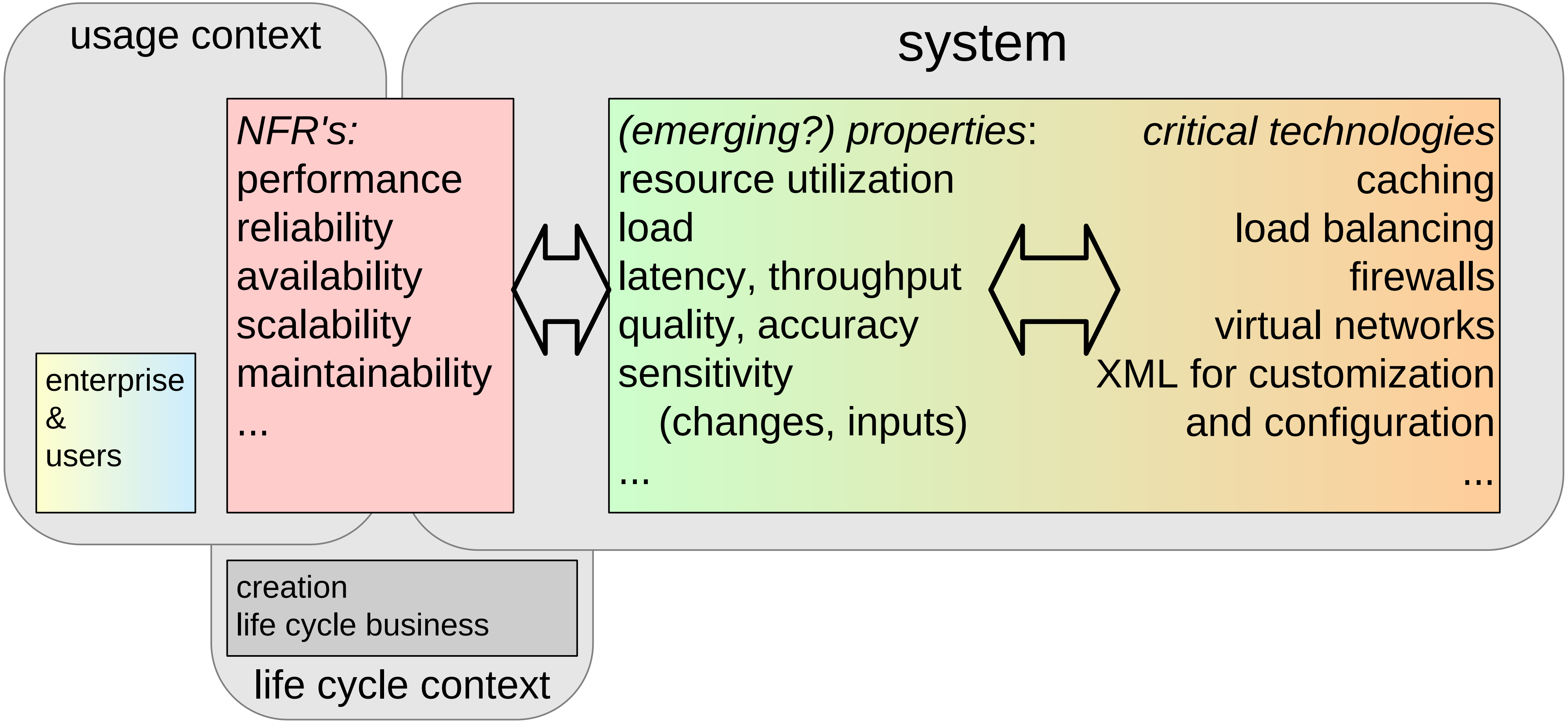
What to model of the system

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

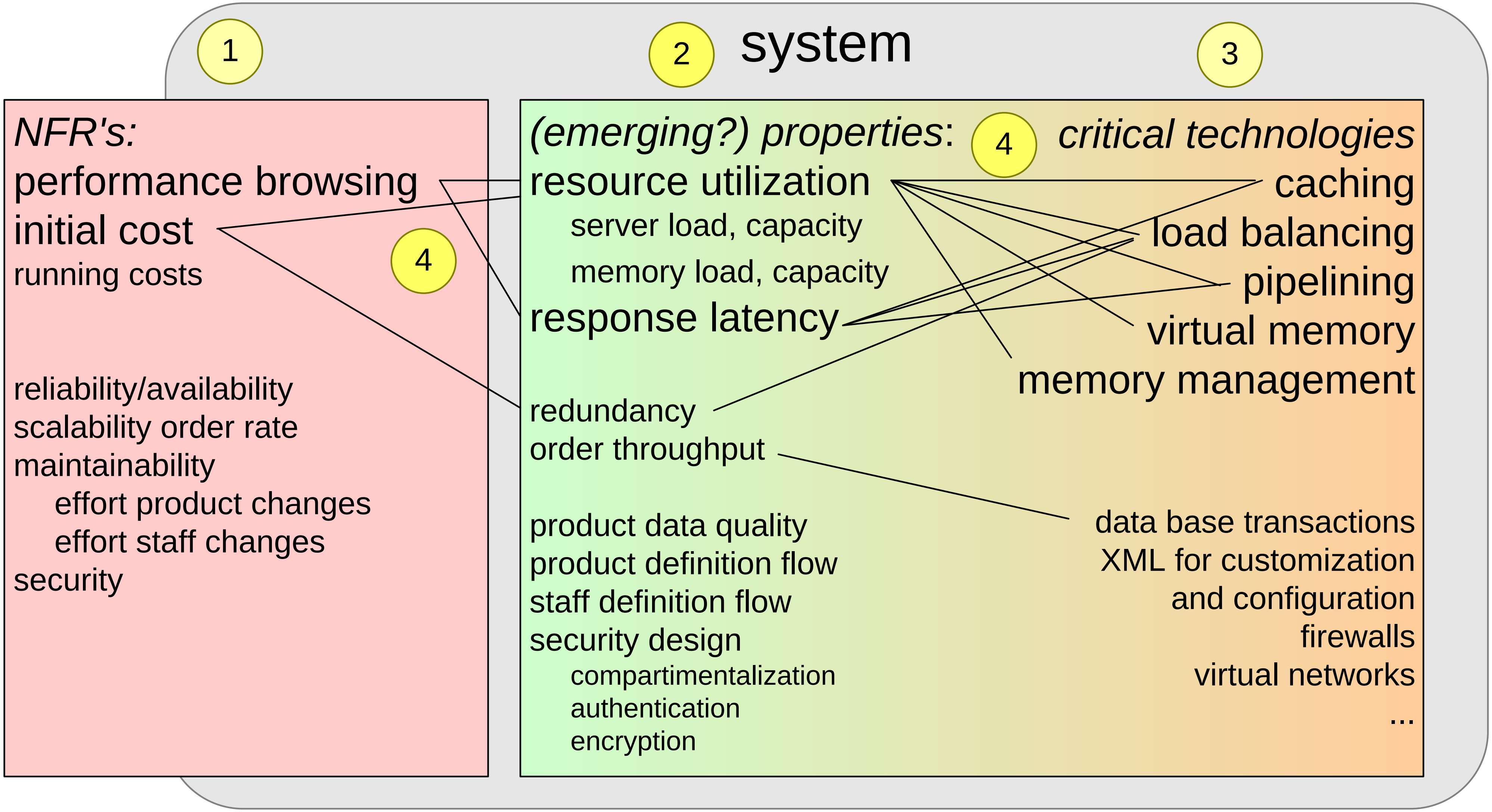
system

1 *NFR's:*
- performance browsing
- initial cost
- running costs
- reliability/availability
- scalability order rate
- maintainability
  - effort product changes
  - effort staff changes
- security

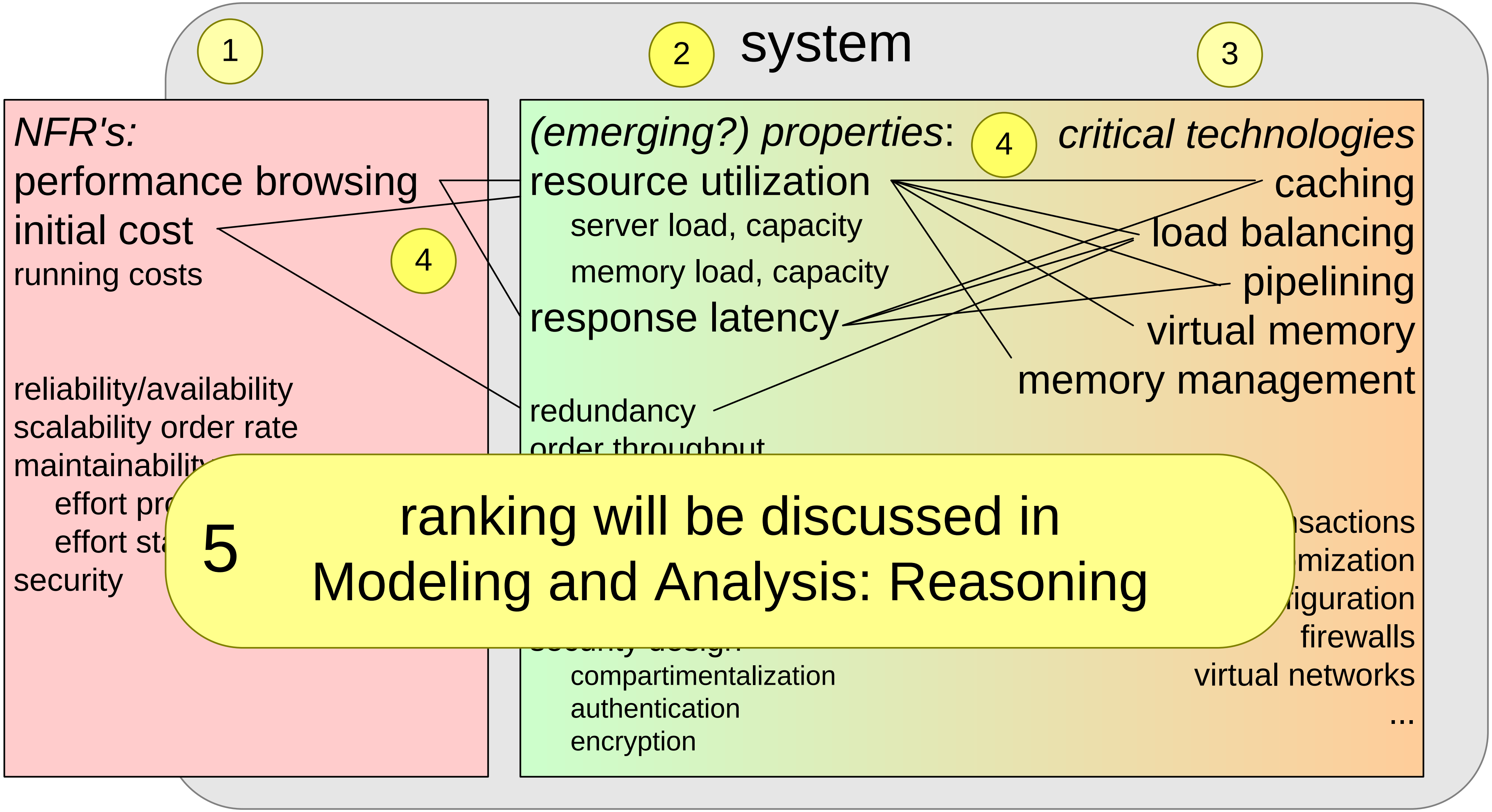
2 *(emerging?) properties*:
- resource utilization
  - server load, capacity
  - memory load, capacity
- response latency
- redundancy
- order throughput
- product data quality
- product definition flow
- staff definition flow
- security design
  - compartimentalization
  - authentication
  - encryption

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

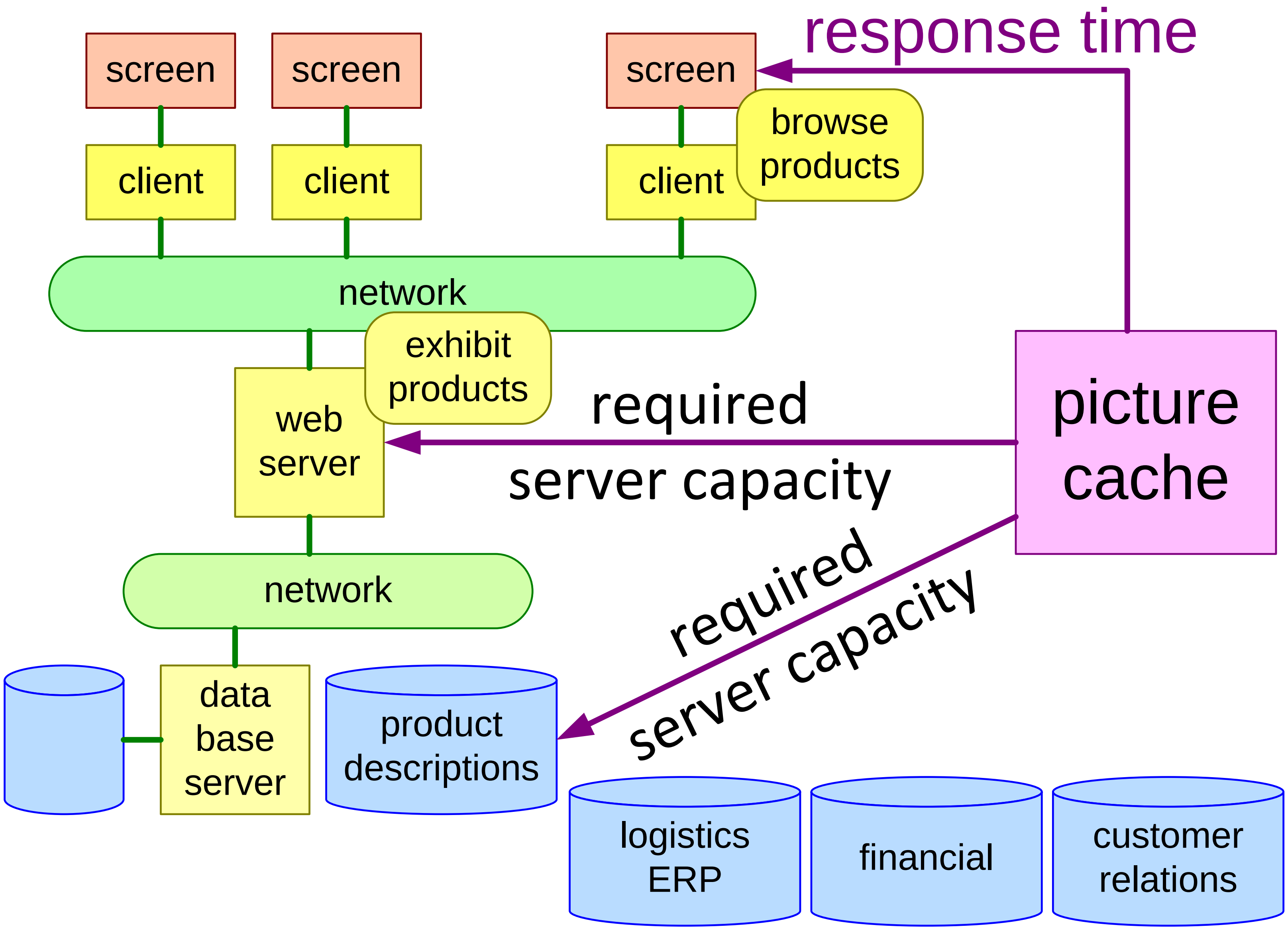
4. Determine Relations



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

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

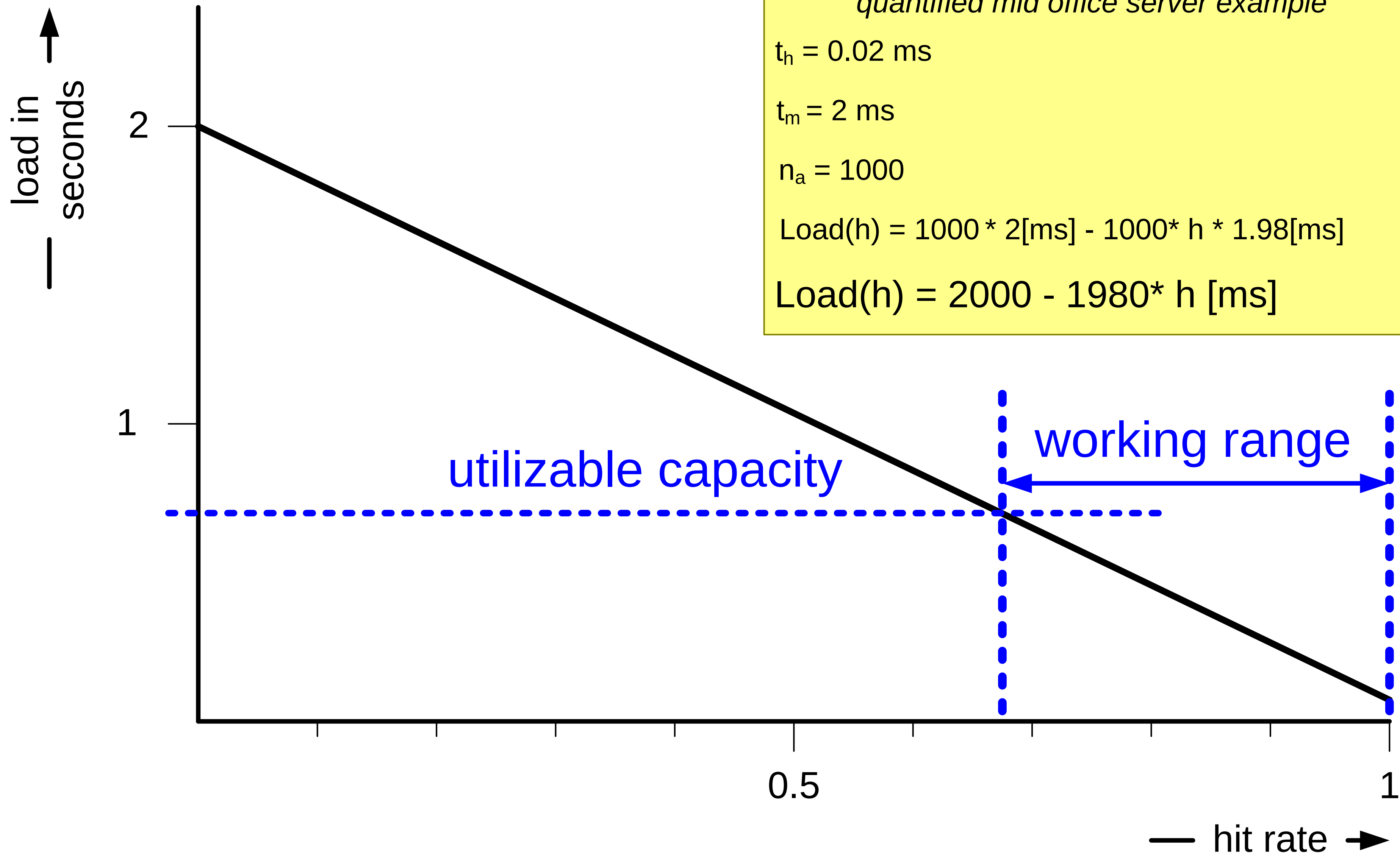
$$n_{a,m} = n_a * (1-h)$$

n_a = total accesses

h = hit rate

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

Quantification: From Formulas to Insight



quantified mid office server example

t_h = 0.02 ms

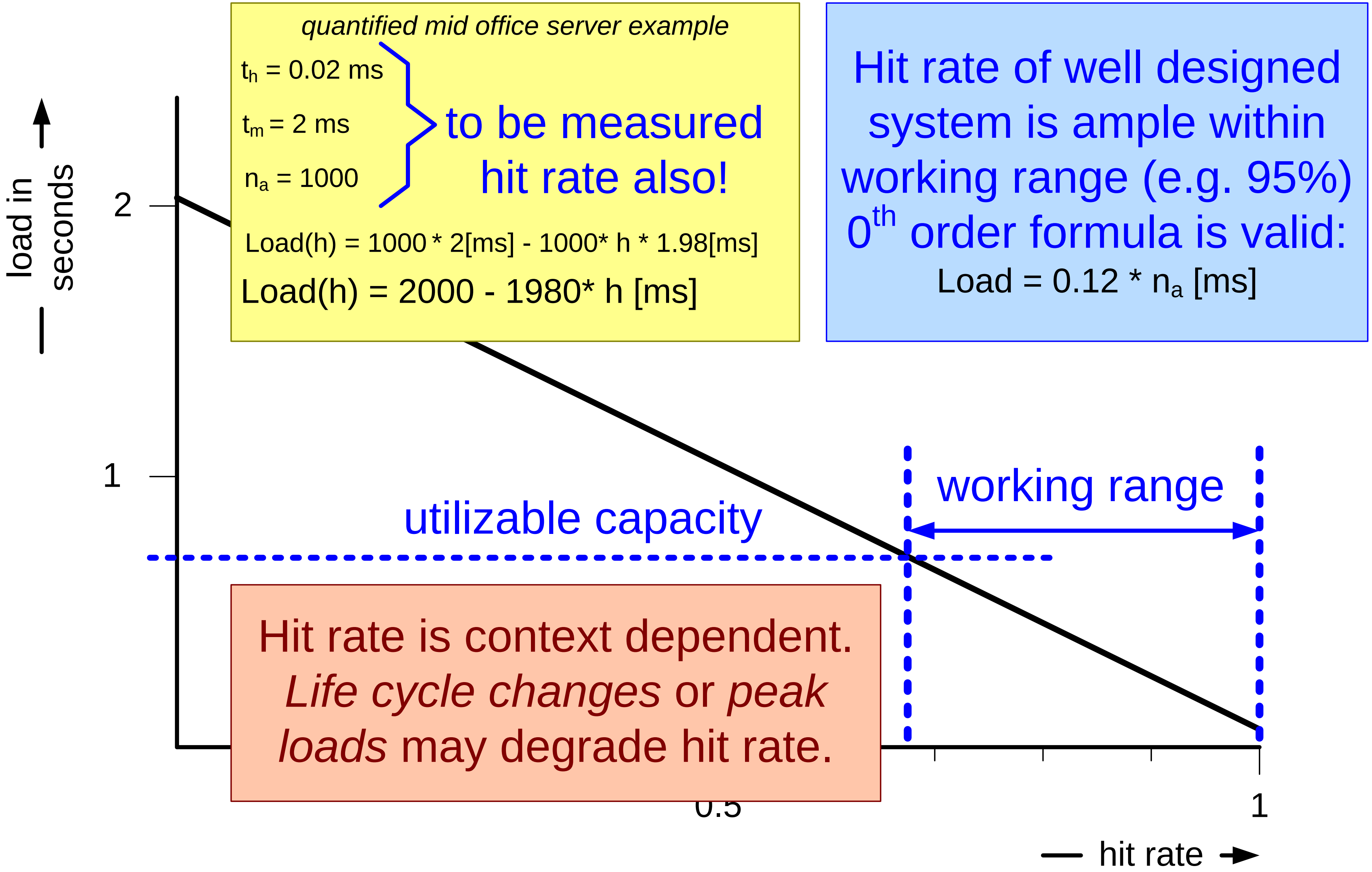
t_m = 2 ms

n_a = 1000

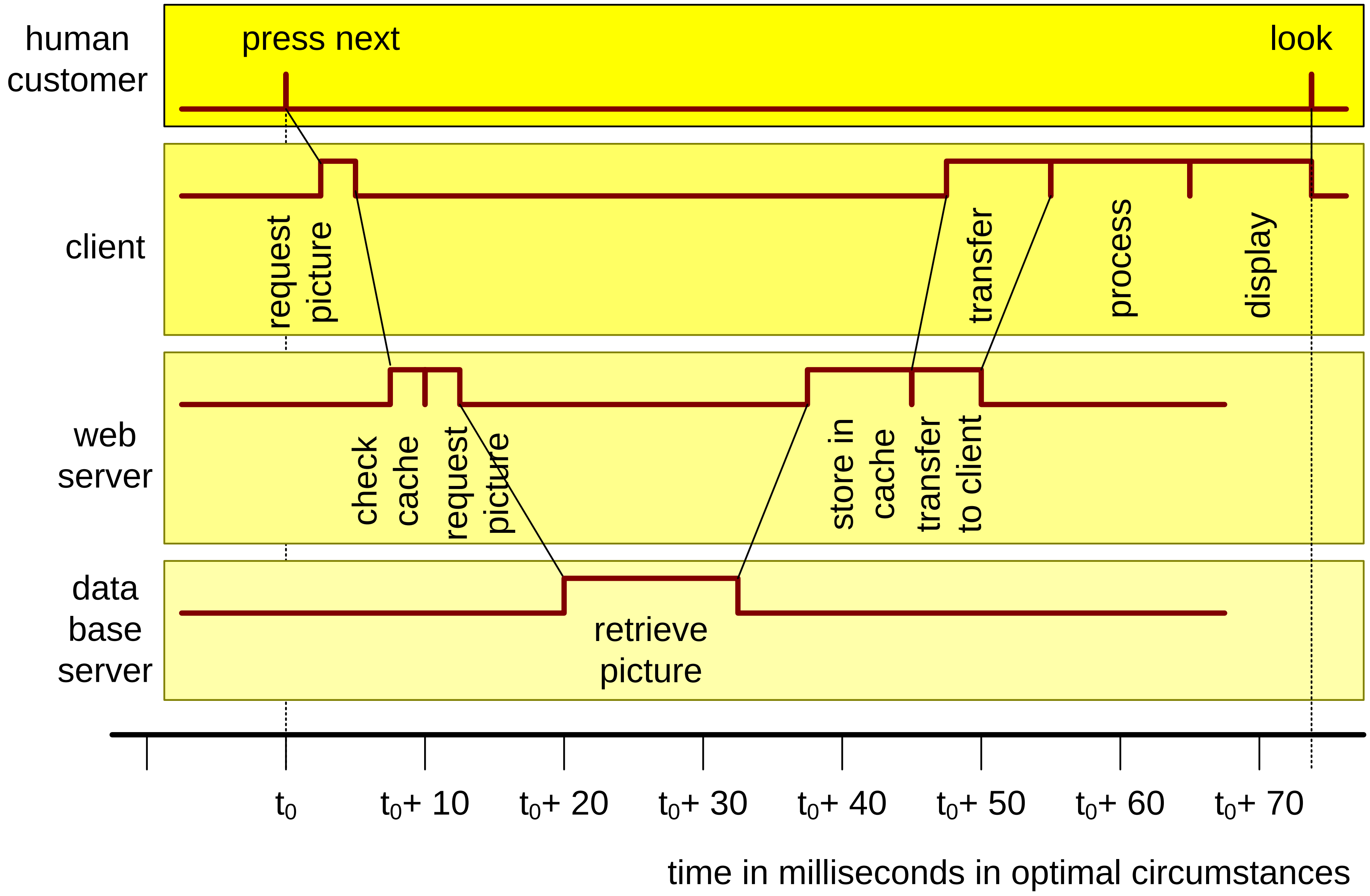
Load(h) = 1000 * 2[ms] - 1000* h * 1.98[ms]

Load(h) = 2000 - 1980* h [ms]

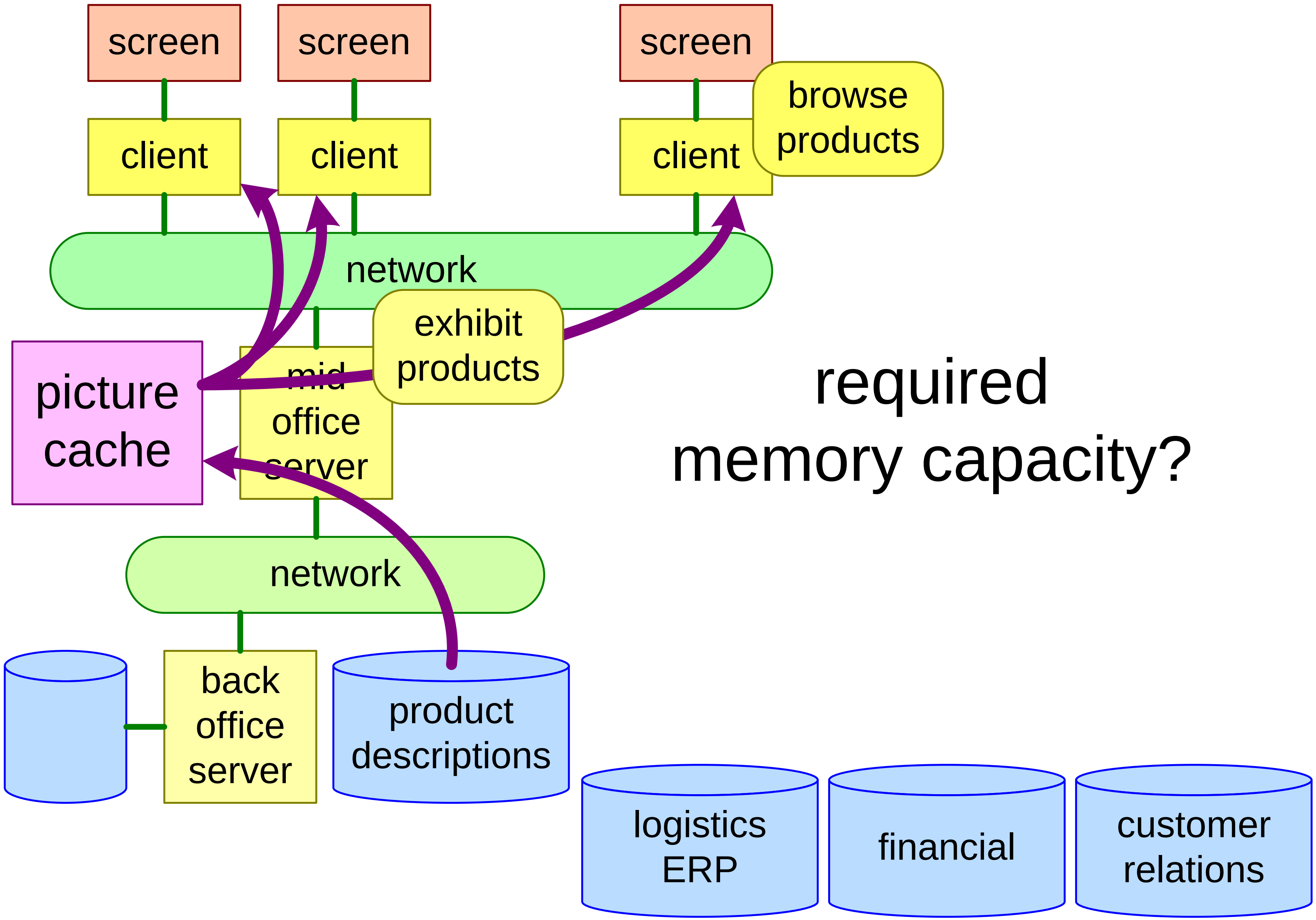
Hit Rate Considerations



Response Time

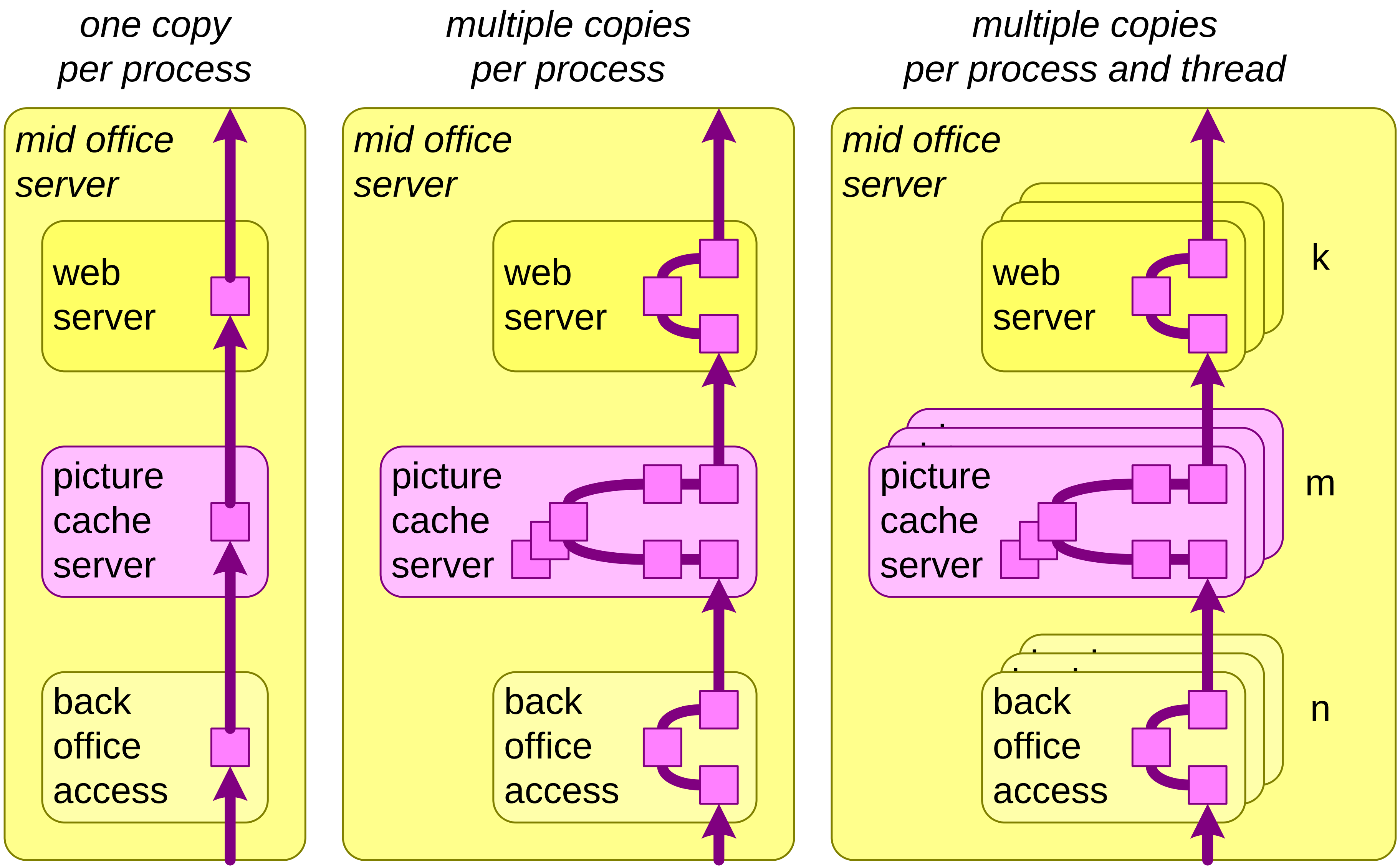


What Memory Capacity is Required for Picture Transfers?



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

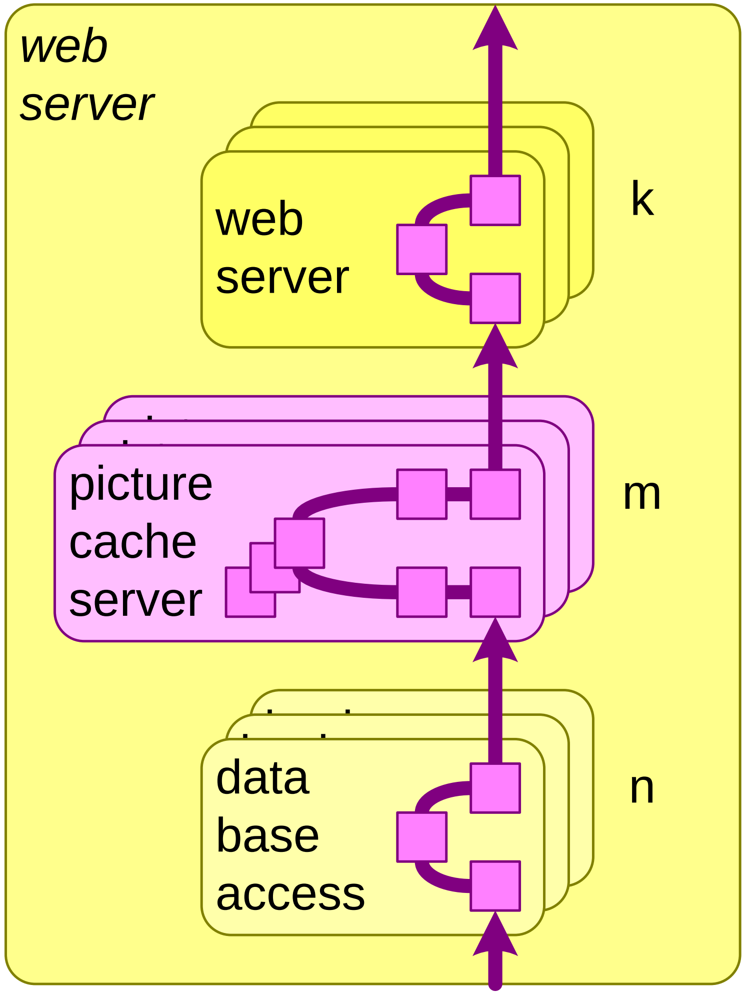
m = # picture cache threads

k = # web server threads

s = picture size in bytes

c = in memory cache capacity in # pictures

multiple copies per process and thread



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

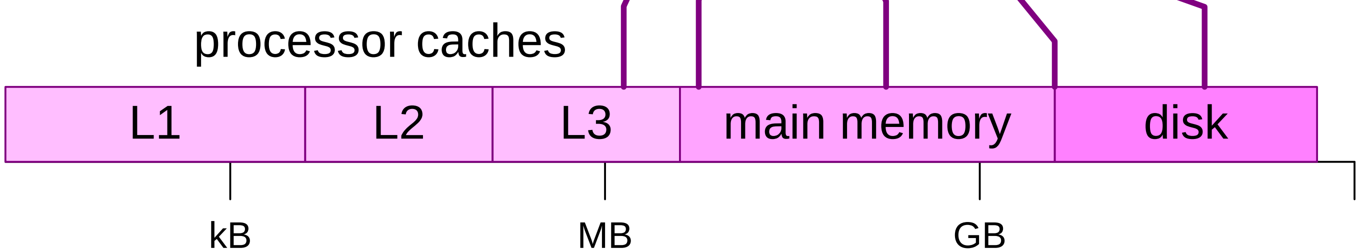
m = # picture cache threads

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

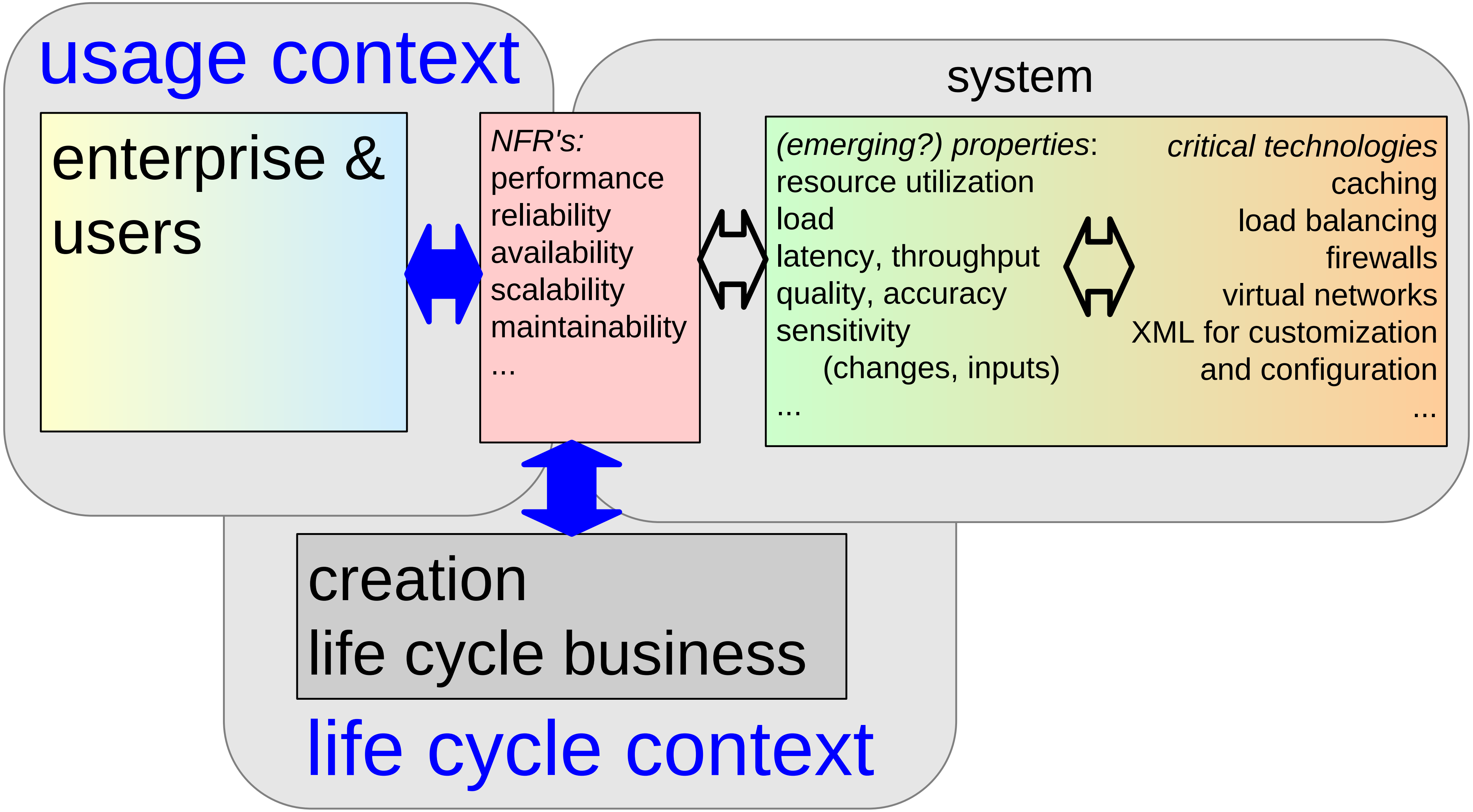
static

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

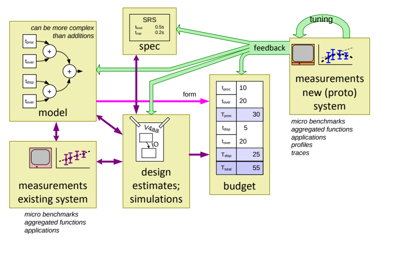
Modeling and Analysis: Budgeting

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.

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***

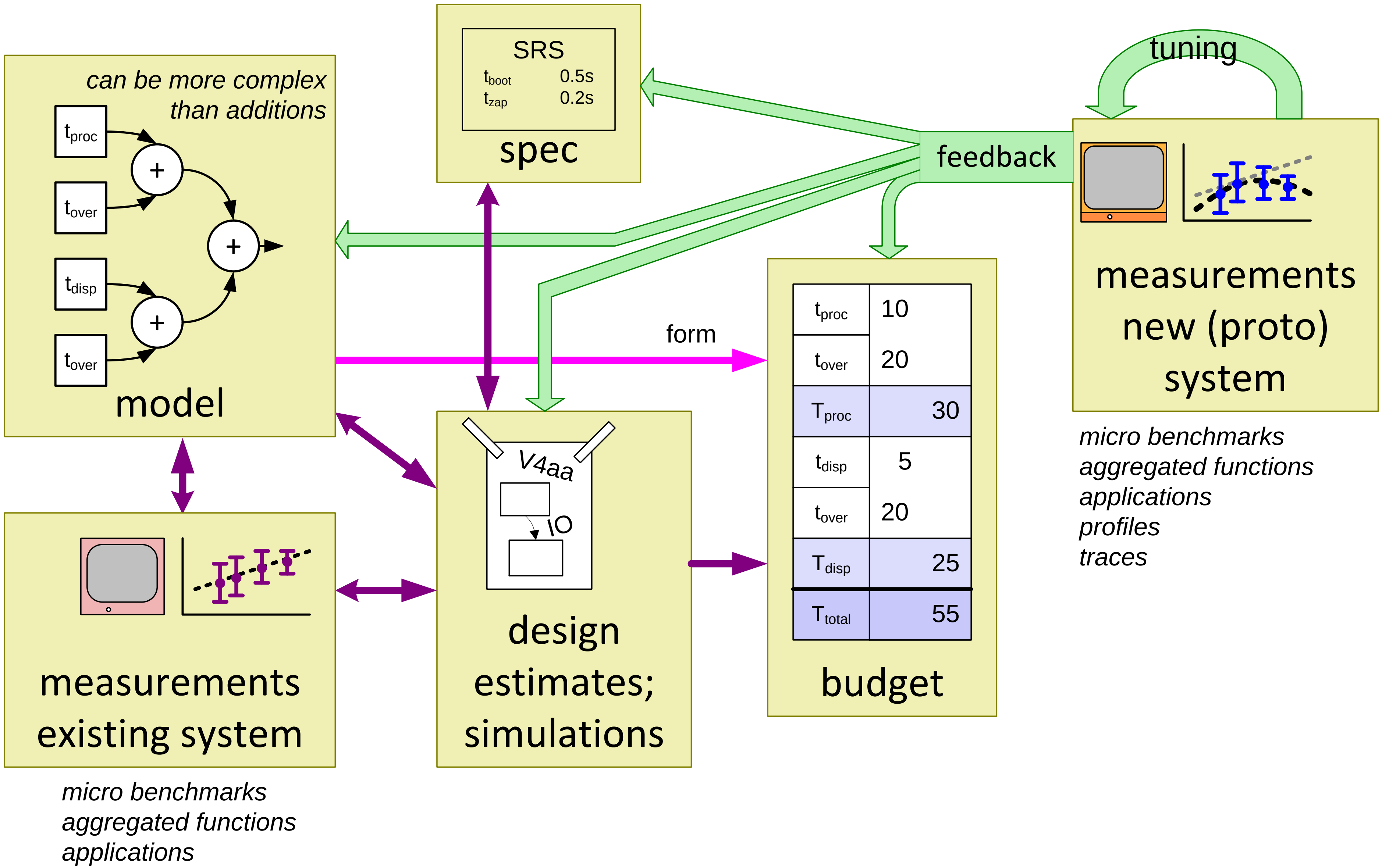
by ***parts*** *of the* ***solution***

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

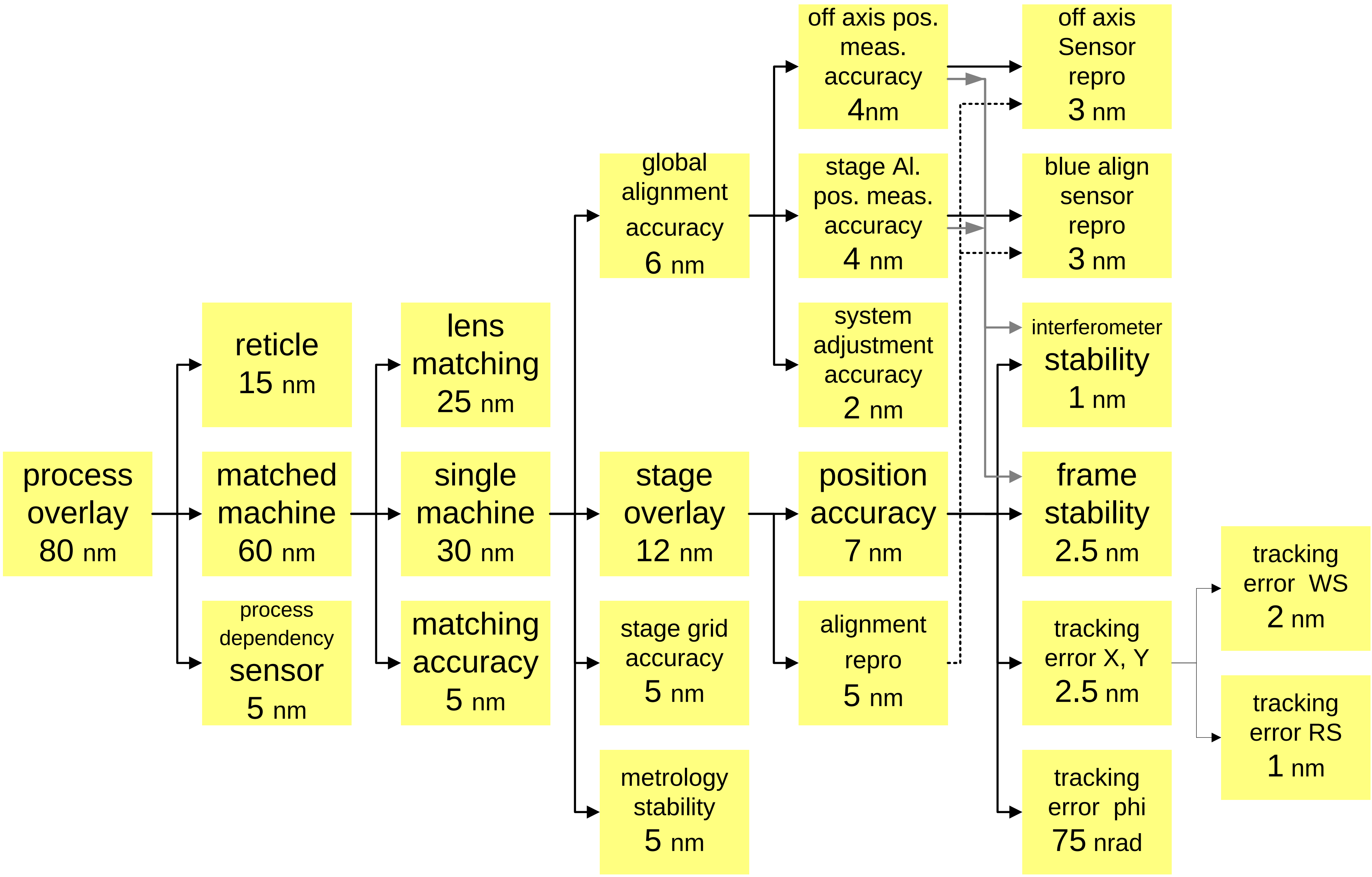
Visualization of Budget Based Design Flow



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	

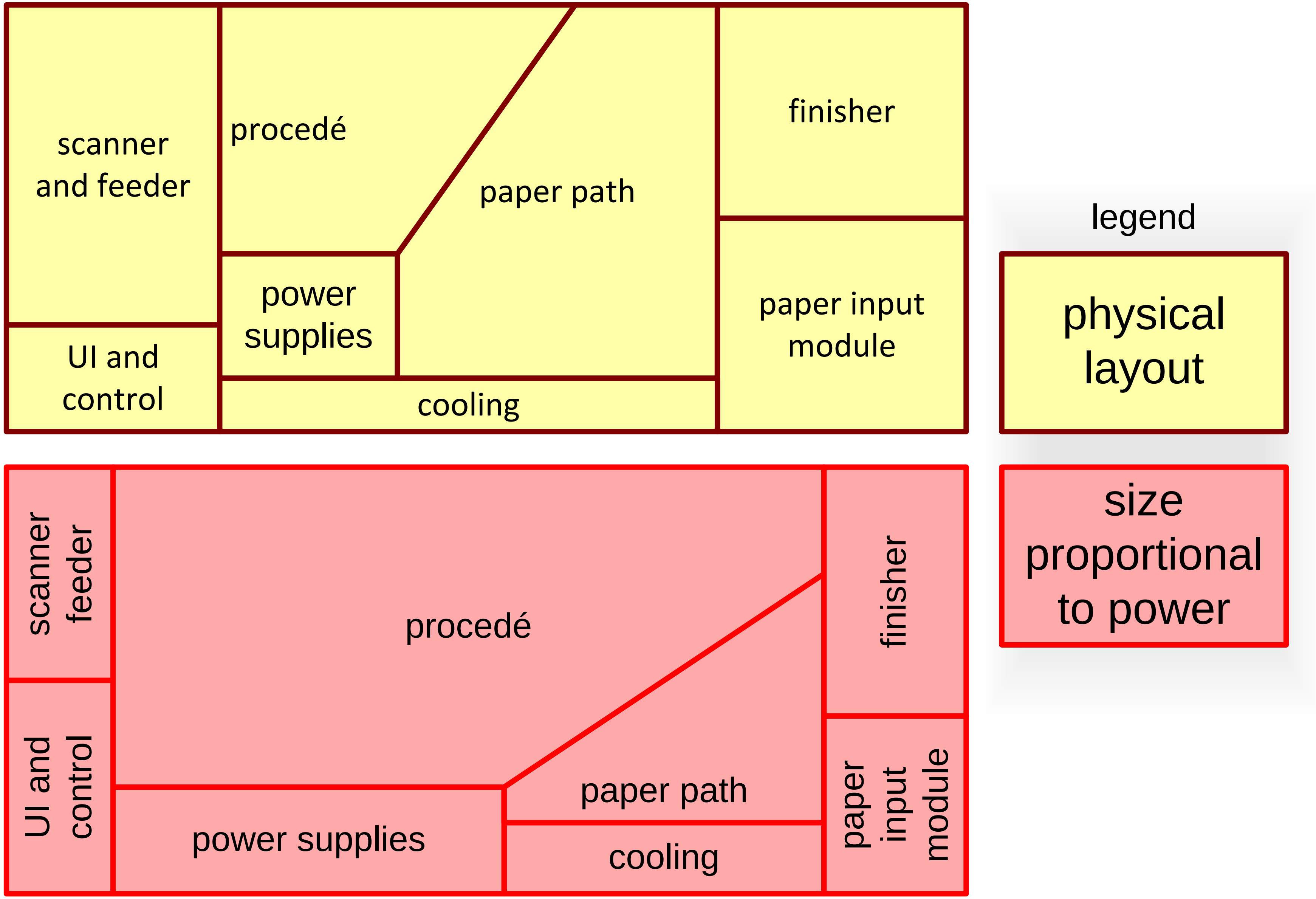
Budgets Applied on Waferstepper Overlay



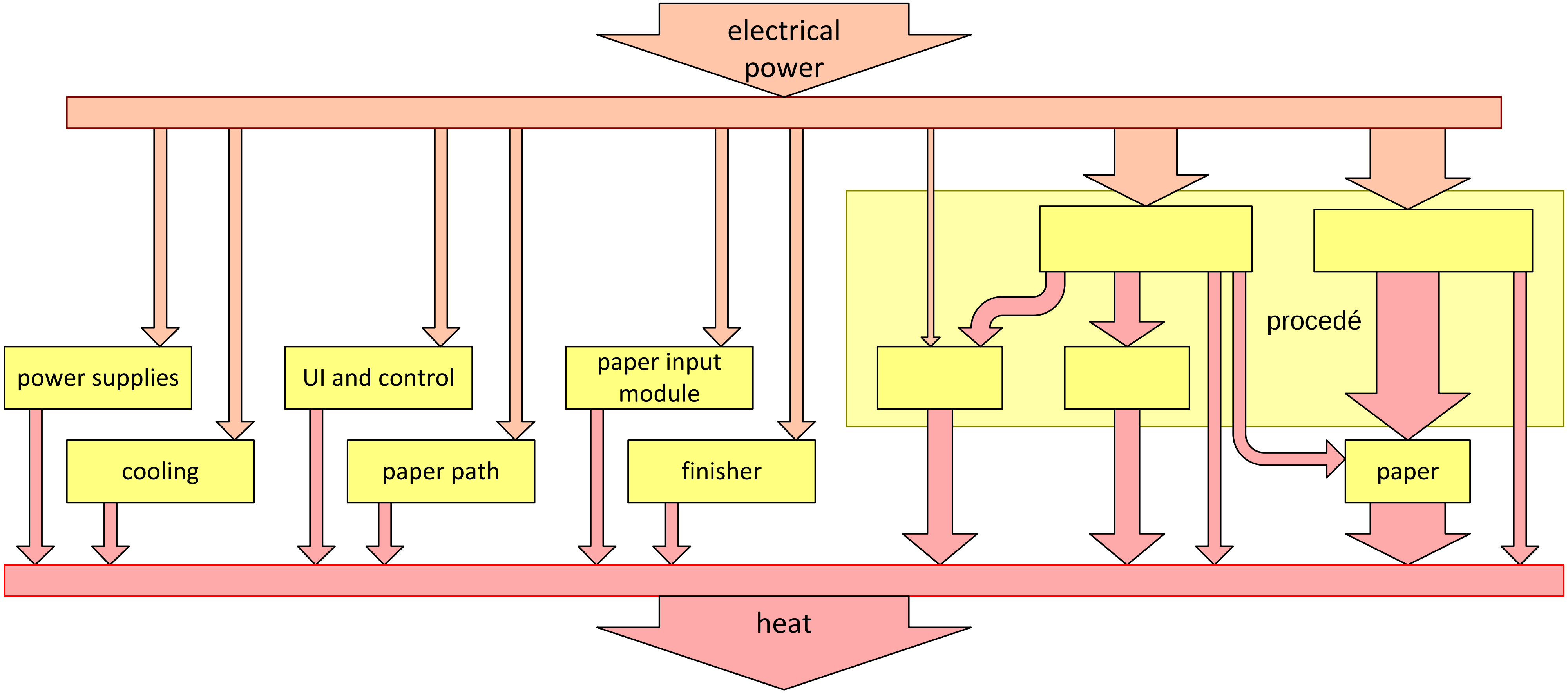
Budgets Applied on Medical Workstation Memory Use

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.