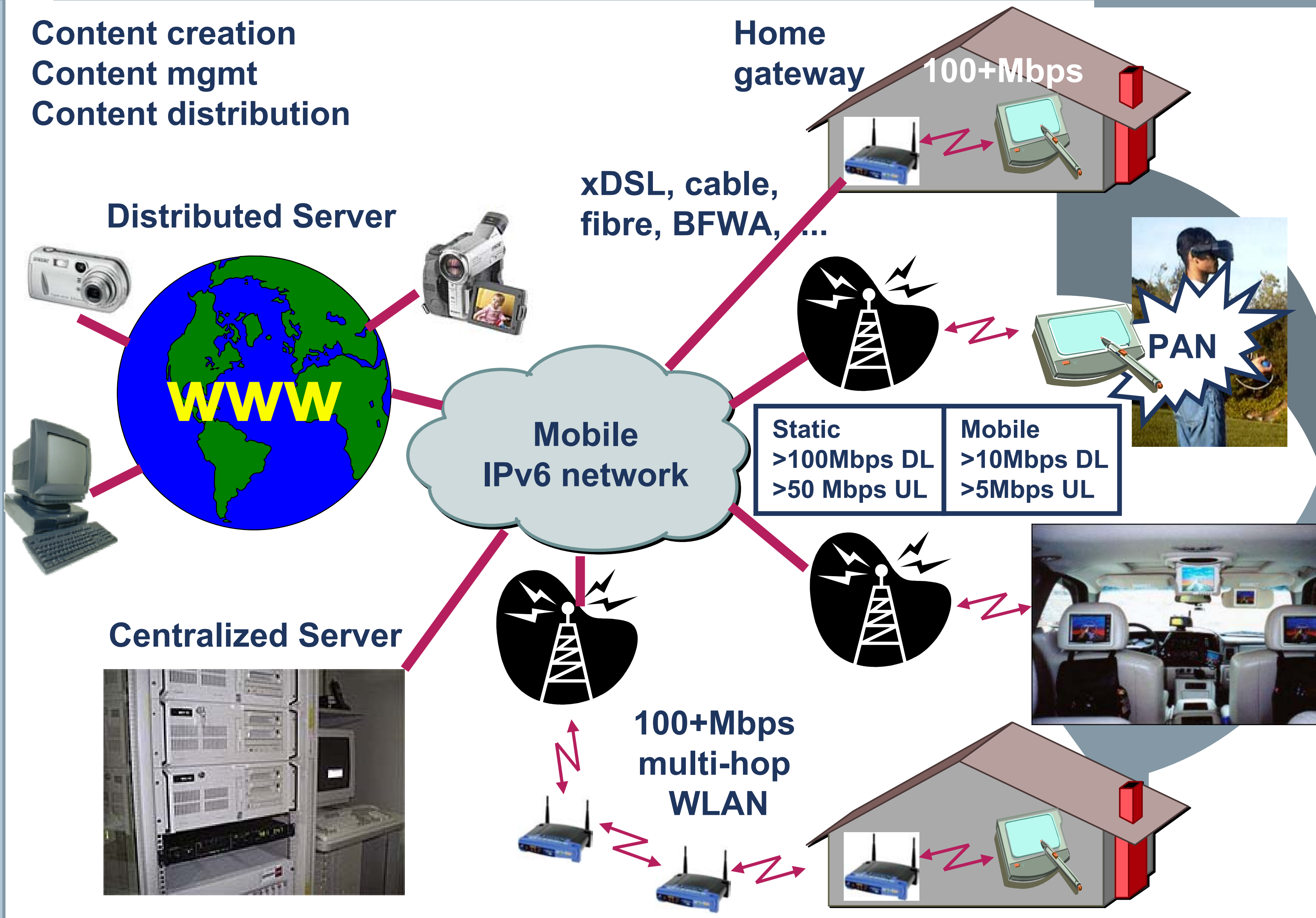
Managing dynamic real-time tasks for modern multi-media systems on multi-processor platforms

Rudy Lauwereins, IMEC & KULeuven, Belgium

(with input of MATADOR project members)

A View Into the Future…

Ubiquitous communication: fixed, mobile and nomadic convergence

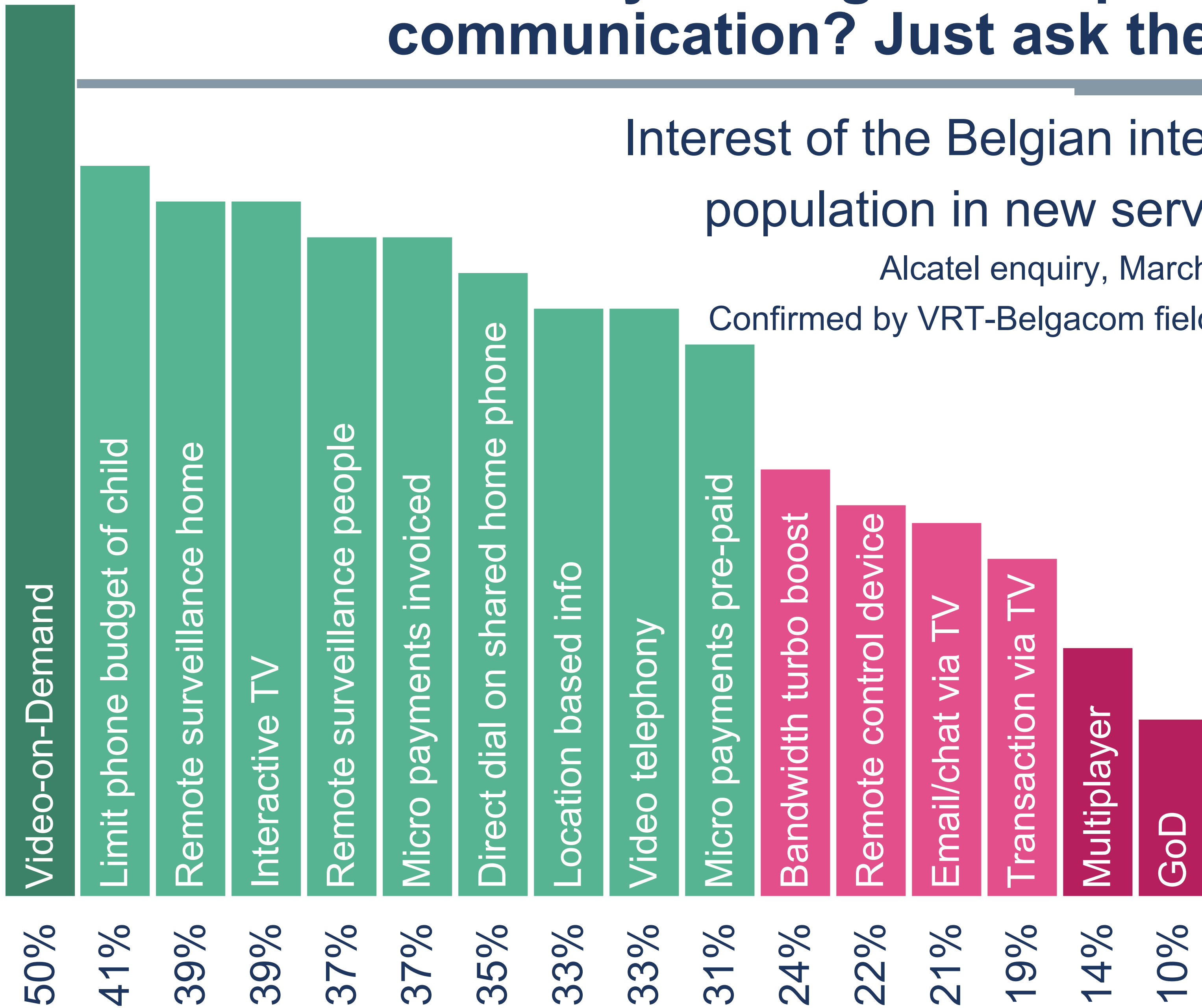


Are customers really waiting for ubiquitous communication? Just ask them!

Interest of the Belgian internet population in new services

Alcatel enquiry, March 2003

Confirmed by VRT-Belgacom field trials



What do we learn from these customer enquiries?

More **mobile** terminal, less fixed TV & phone

More **multimedia**

- Video-on-demand
- Remote surveillance of property and people
- Video telephony

More **personalized**

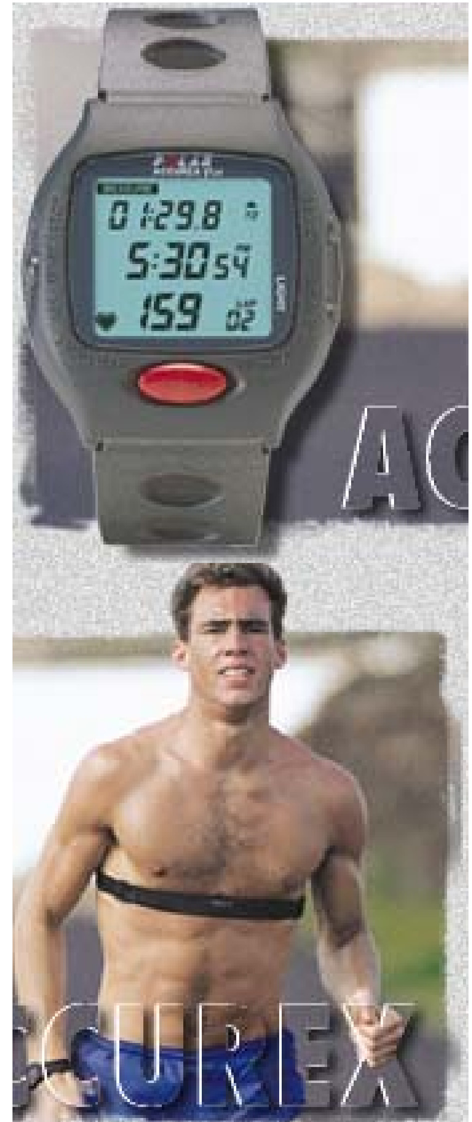
- Personalized services on shared phone: direct dial, mailbox, …
- Location based personal services
- Phone budget control

Service driven, not technology driven

Hence, focus on

- **Wireless** communication
- **Multimedia** services
- **Design technology** for low power

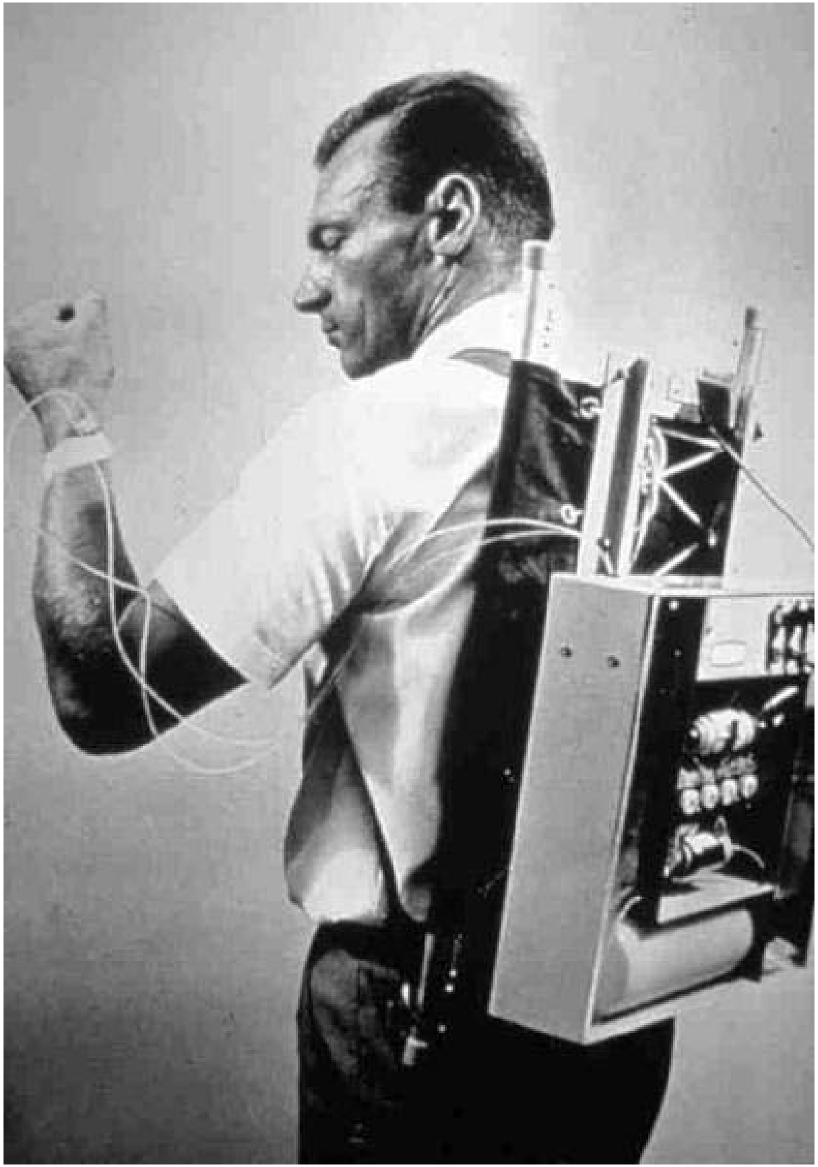
Some user scenario's: Electronic Devices Support Athletes



Now: Heart beat

Future: Blood composition

Some user scenario's: Electronic Devices for Health Care



Insuline pump in 1963…

Some miniaturization could improve comfort…

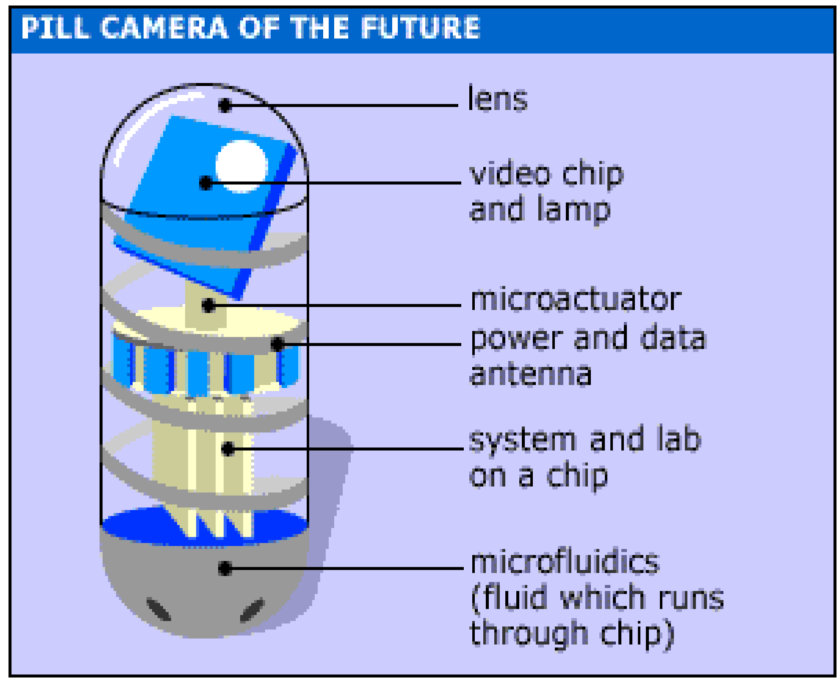
Some user scenario’s: Electronic Devices for Health Care

State-of-the-art insuline pump

Still manual control…

Wanted: Implanted glucose sensor with wireless transmission to pump

Some user scenario's:
Electronic Devices for Diagnostics and Repair



Managing Dynamic Real-Time Tasks

A view into the future

Properties of ambient multimedia platforms

The dynamic nature of ambient multimedia applications

Methodology for managing dynamic real-time tasks

Applying this methodology to an MPEG-21 QoS application

Main messages

Managing Dynamic Real-Time Tasks

A view into the future

Properties of ambient multimedia platforms

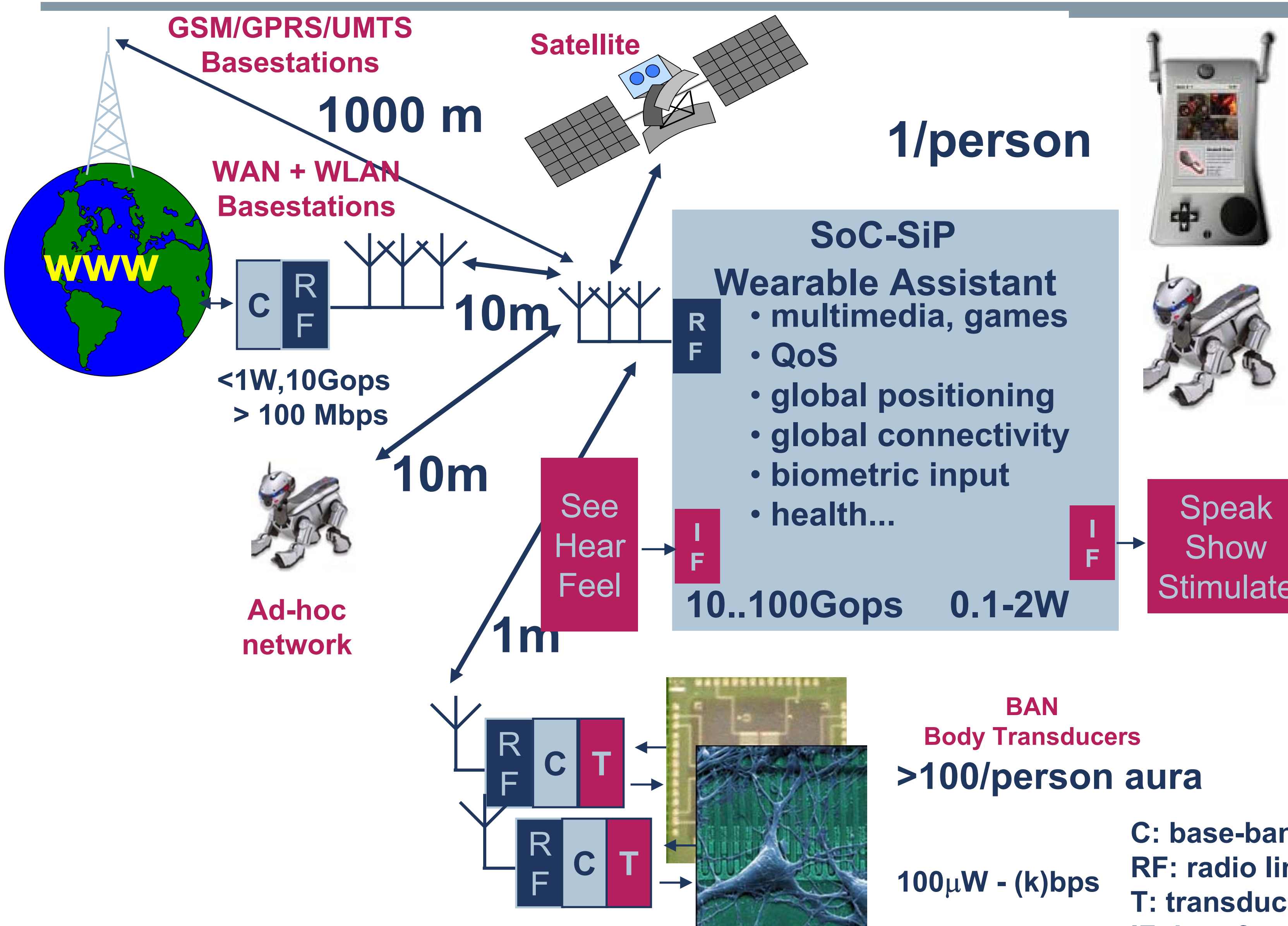
The dynamic nature of ambient multimedia applications

Methodology for managing dynamic real-time tasks

Applying this methodology to an MPEG-21 QoS application

Main messages

IMEC's vision on the end-user equipment that has to be built by your customers



What are the properties of these Ambient Intelligence architectures?

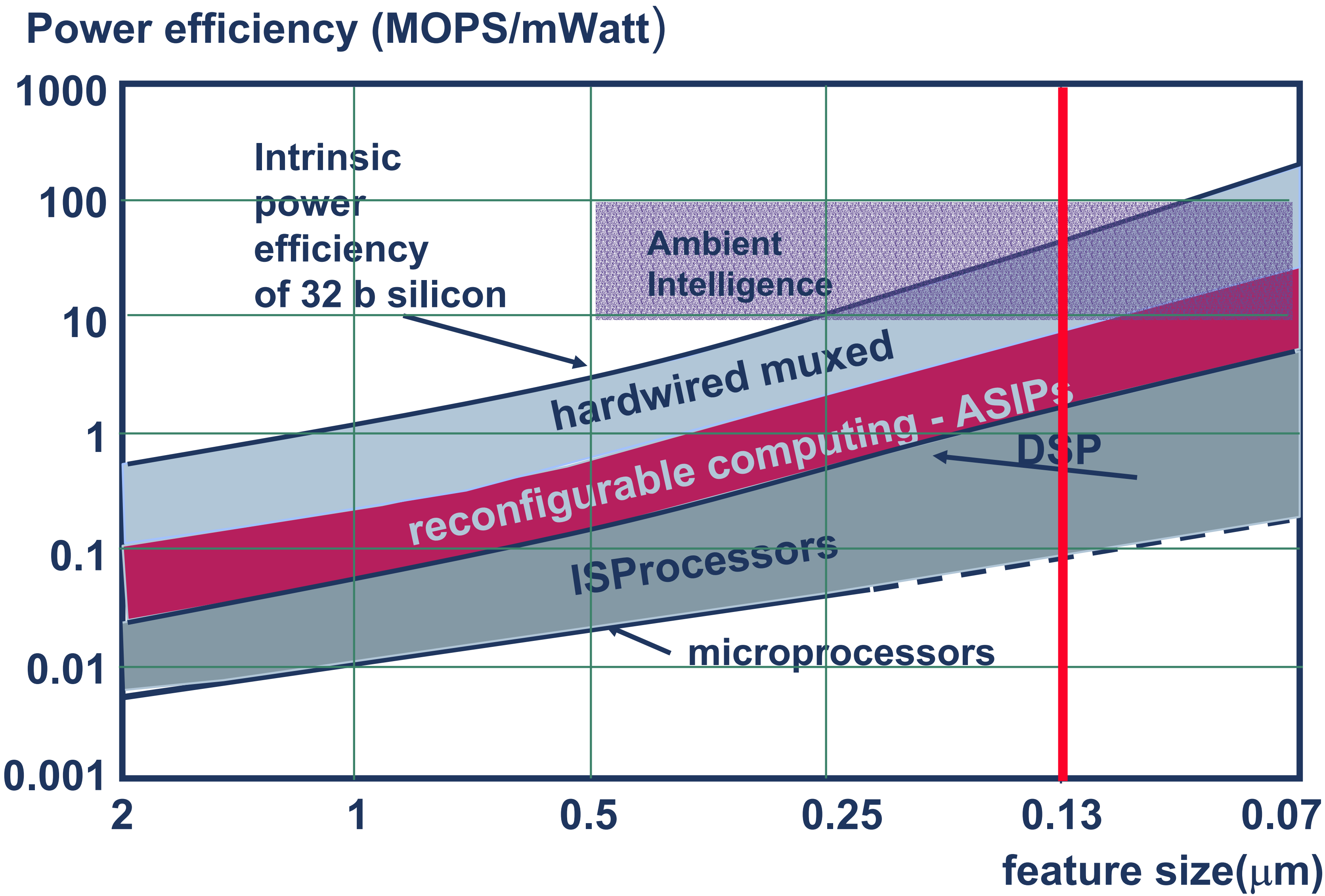
Need global system optimization
GHz RF and mixed signal everywhere

Transducer nodes	Assistant nodes/basestations
Ultra low energy **(100Mops/mW)**	Low energy **(10-50Mops/mW)**
Low flexibility	**High Flexibility**
Ultra low cost (1€)	Low cost (100 €)
1..10 Mtr (small size)	10..100 Gops, >100 Mtr
Low clock frequency	High (not GHz) frequency
Chip-package co-design	Dynamic task management

"PACKAGE in a week" **"PLATFORM"**

@100..1000 times Power efficiency of today's μP...

Energy efficiency and flexibility are conflicting requirements



After T.Claasen et al. (ISSCC99)

Goal of this course

To present a methodology to map ***dynamic*** and ***concurrent*** real-time applications on an embedded multi-processor platform

Managing Dynamic Real-Time Tasks

A view into the future

Properties of ambient multimedia platforms

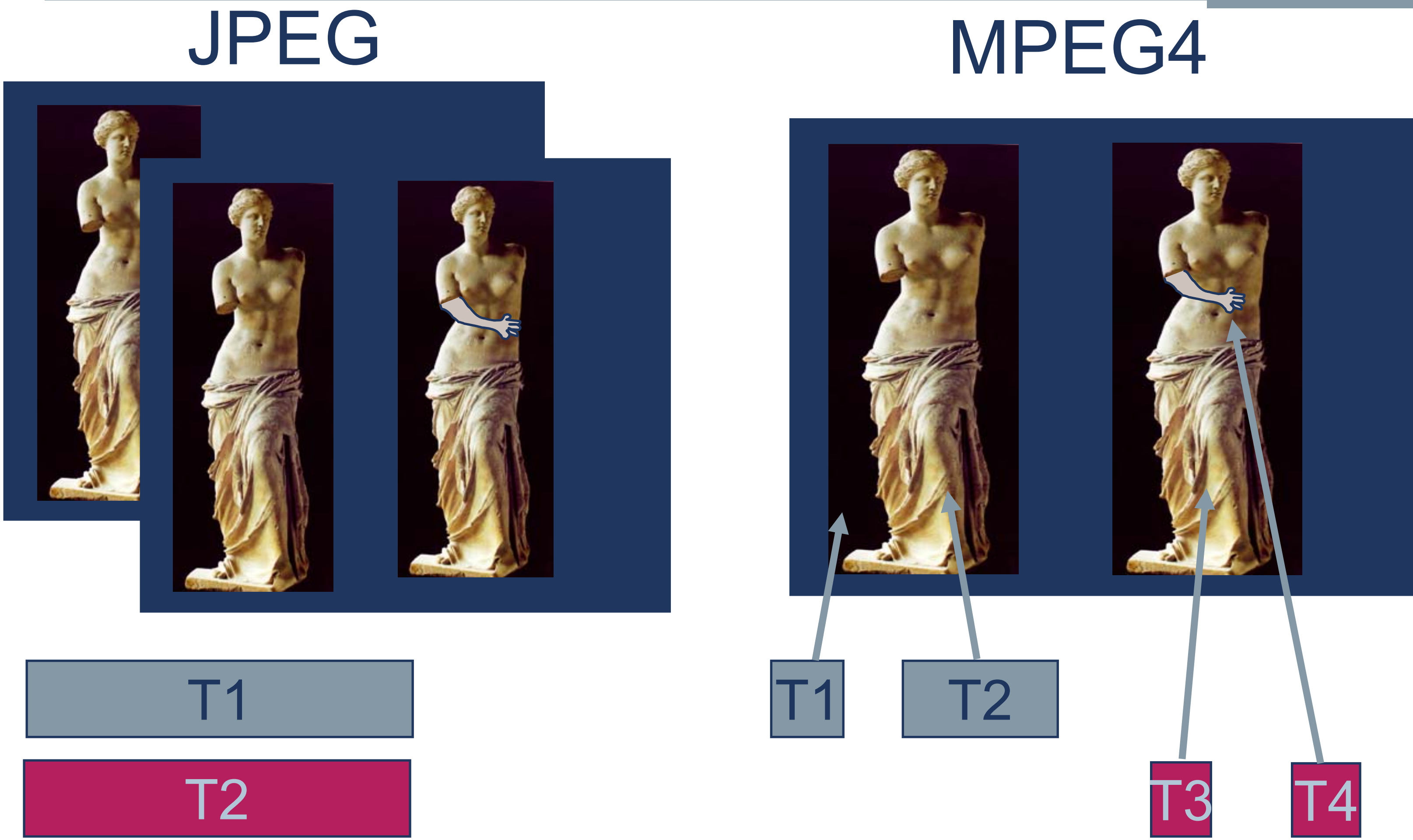
The dynamic nature of ambient multimedia applications

Methodology for managing dynamic real-time tasks

Applying this methodology to an MPEG-21 QoS application

Main messages

Why are applications becoming more dynamic and concurrent?



The workload decreases but the tasks are dynamically created and their size is data dependent

What happens if quality is not scalable and you run out of resources?

2D Texture

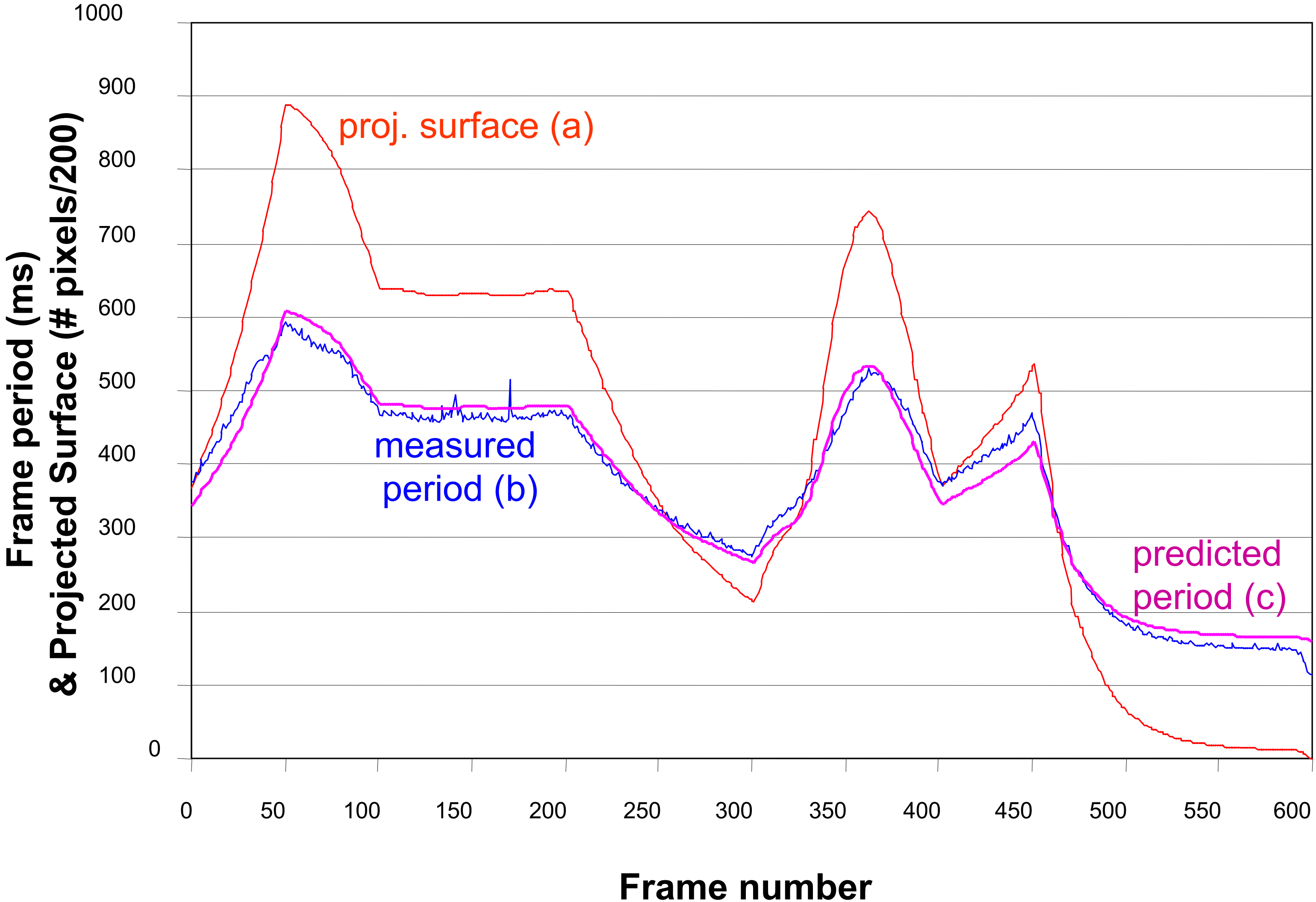
3D Object

Decoded @
unlimited resources

Decoded @
limited resources
without QoS

Decoded @
limited resources
with QoS

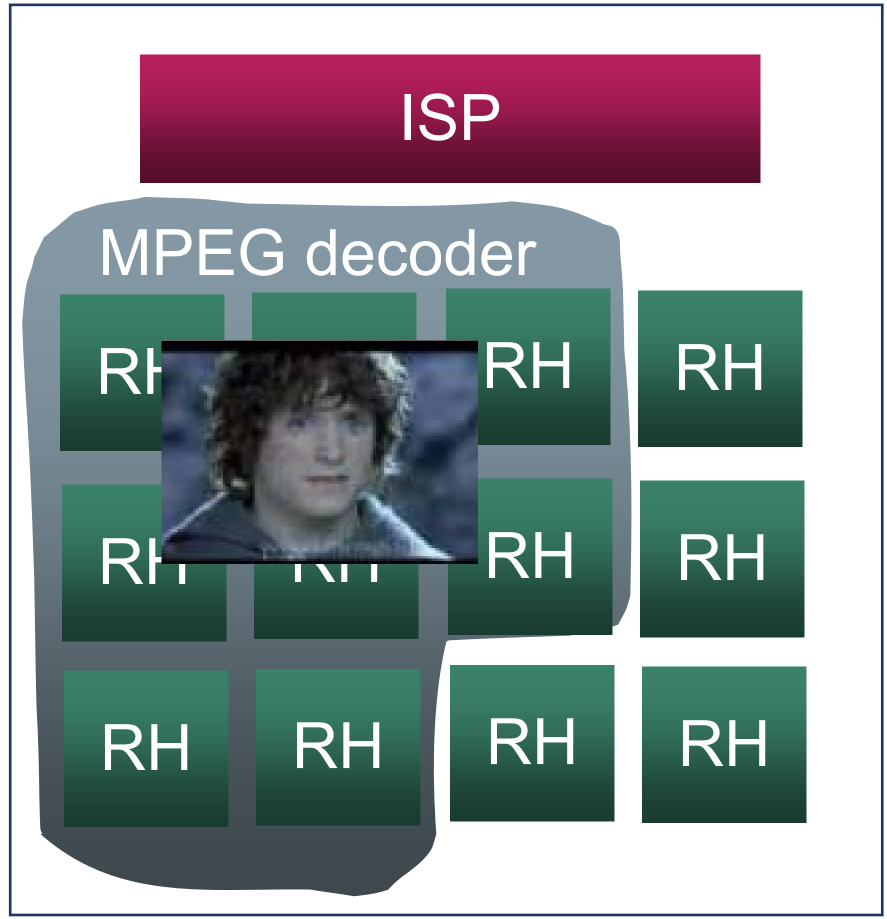
Compute power for a single 3D object is determined by projected surface



Another source of dynamism: multi-tasking and user interaction

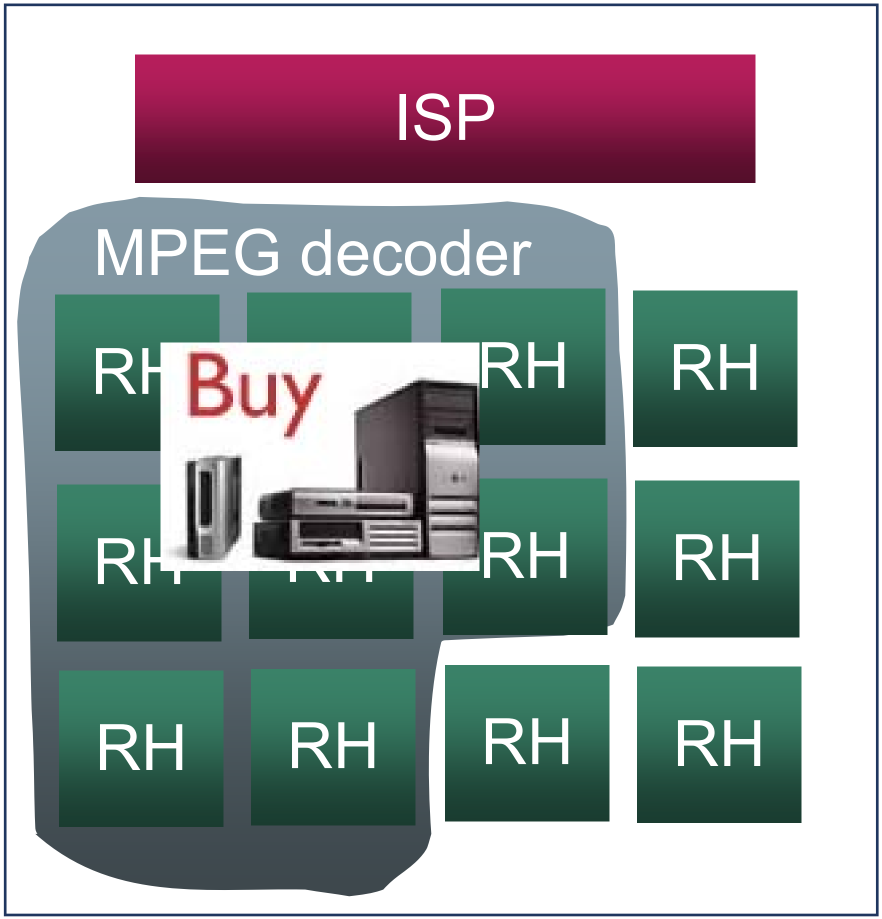
Broadcast TV:

Movie time



Another source of dynamism: multi-tasking and user interaction

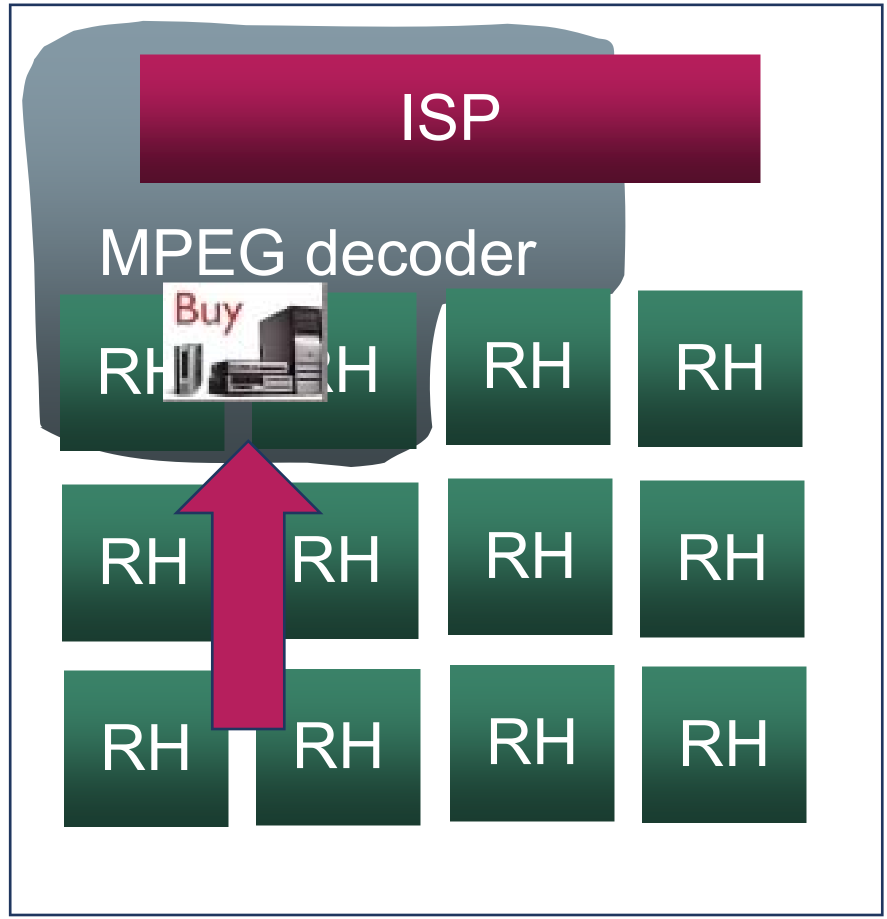
Broadcast TV:
Advertisement



Another source of dynamism: multi-tasking and user interaction

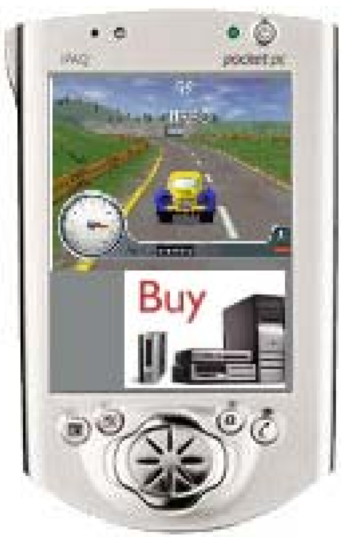
Broadcast TV:

QoS ↘

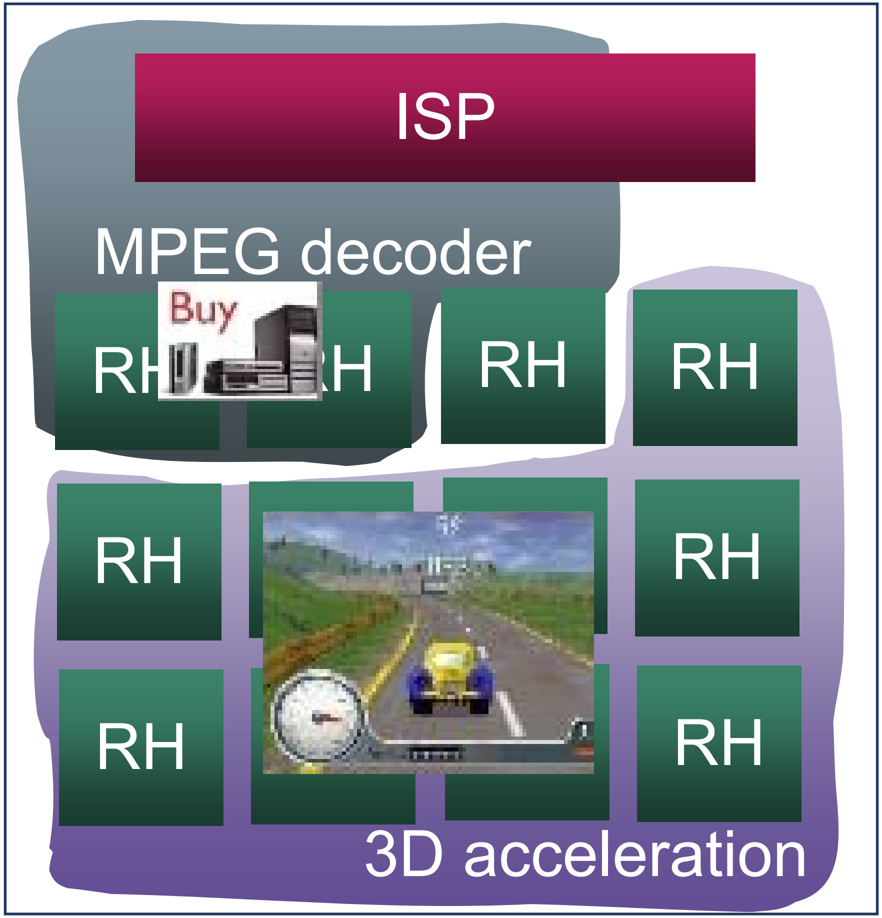


Another source of dynamism: multi-tasking and user interaction

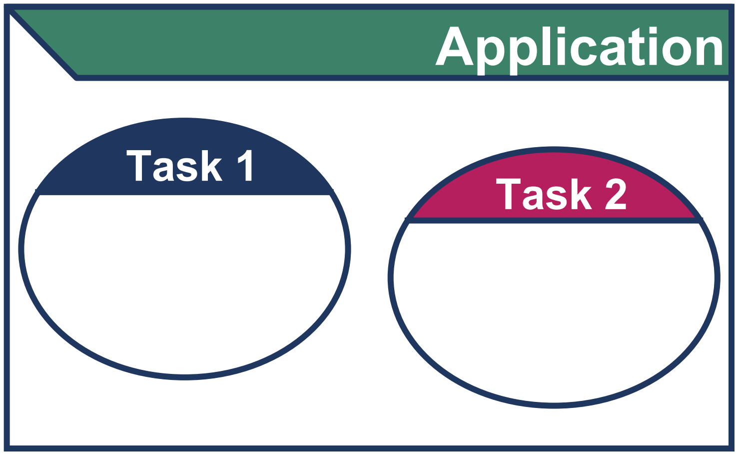
Play time

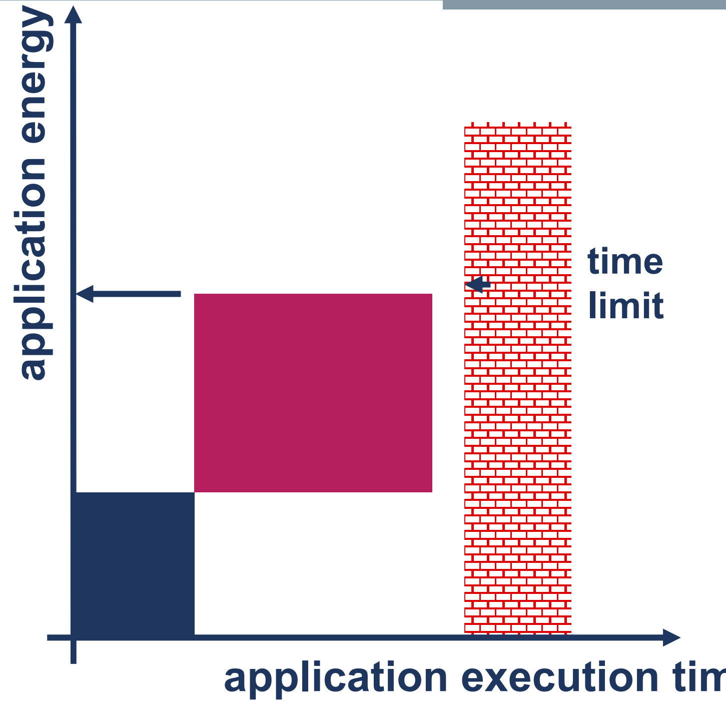


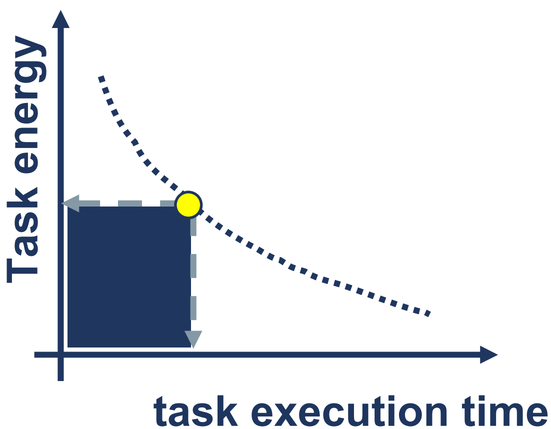
Broadcast TV +
3D game

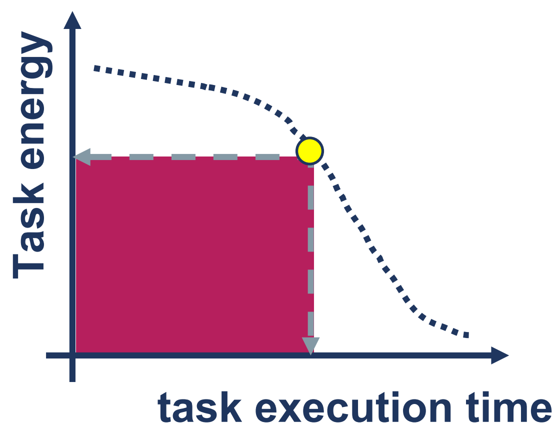


Which method can handle dynamism? Try to select optimum schedule on Pareto.

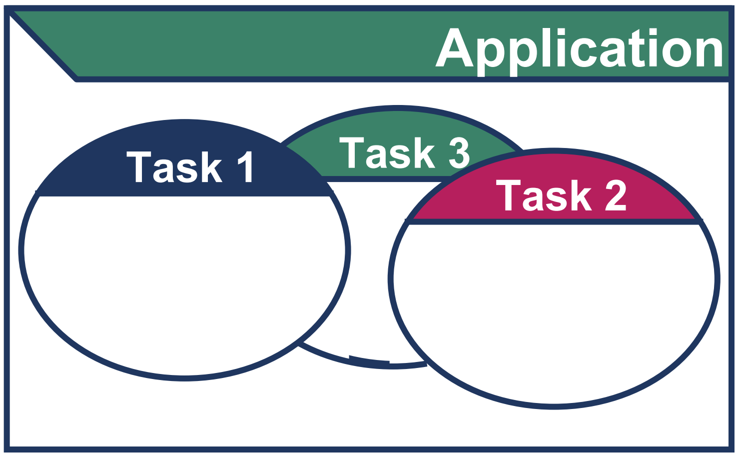


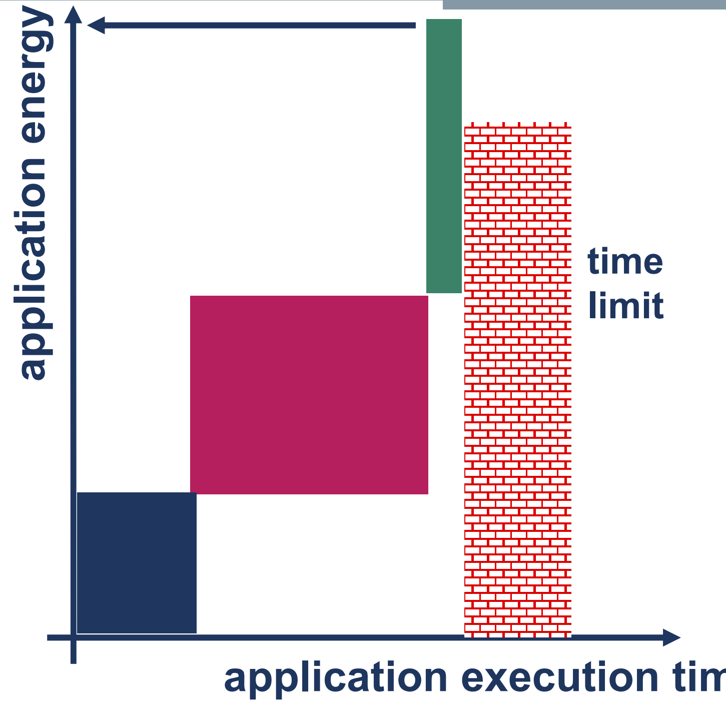


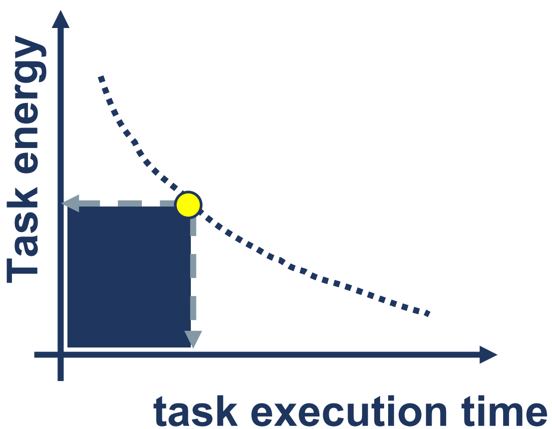


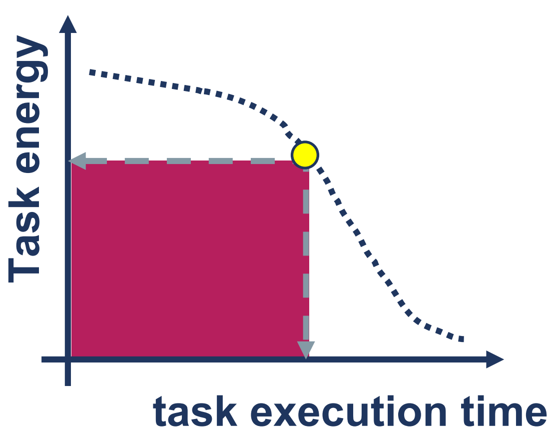


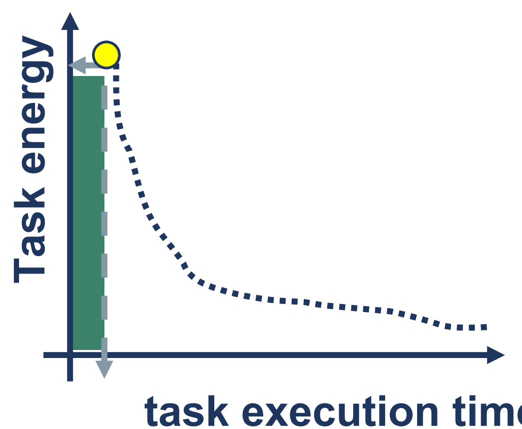
Which method can handle dynamism? But: new task works in high energy point



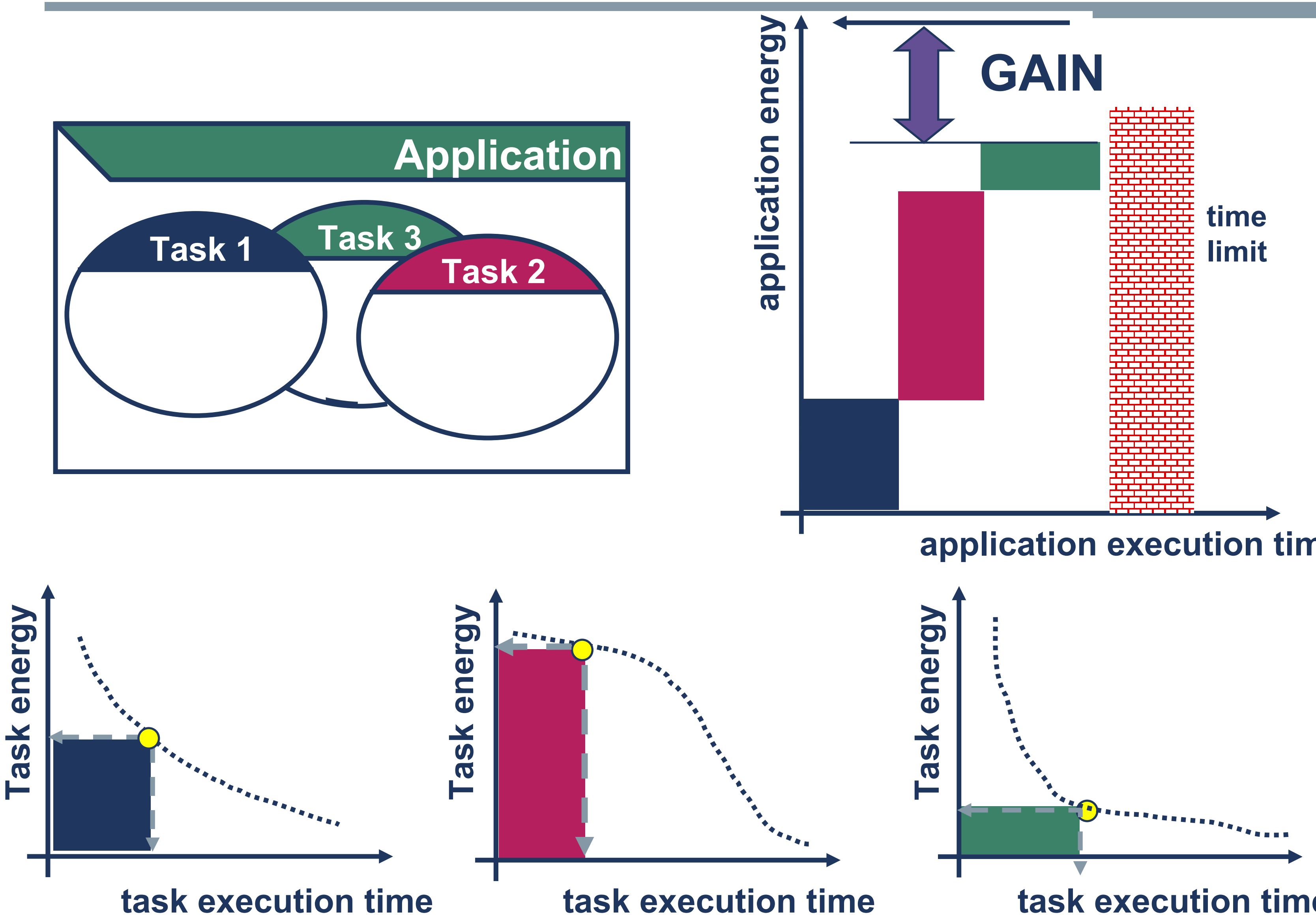




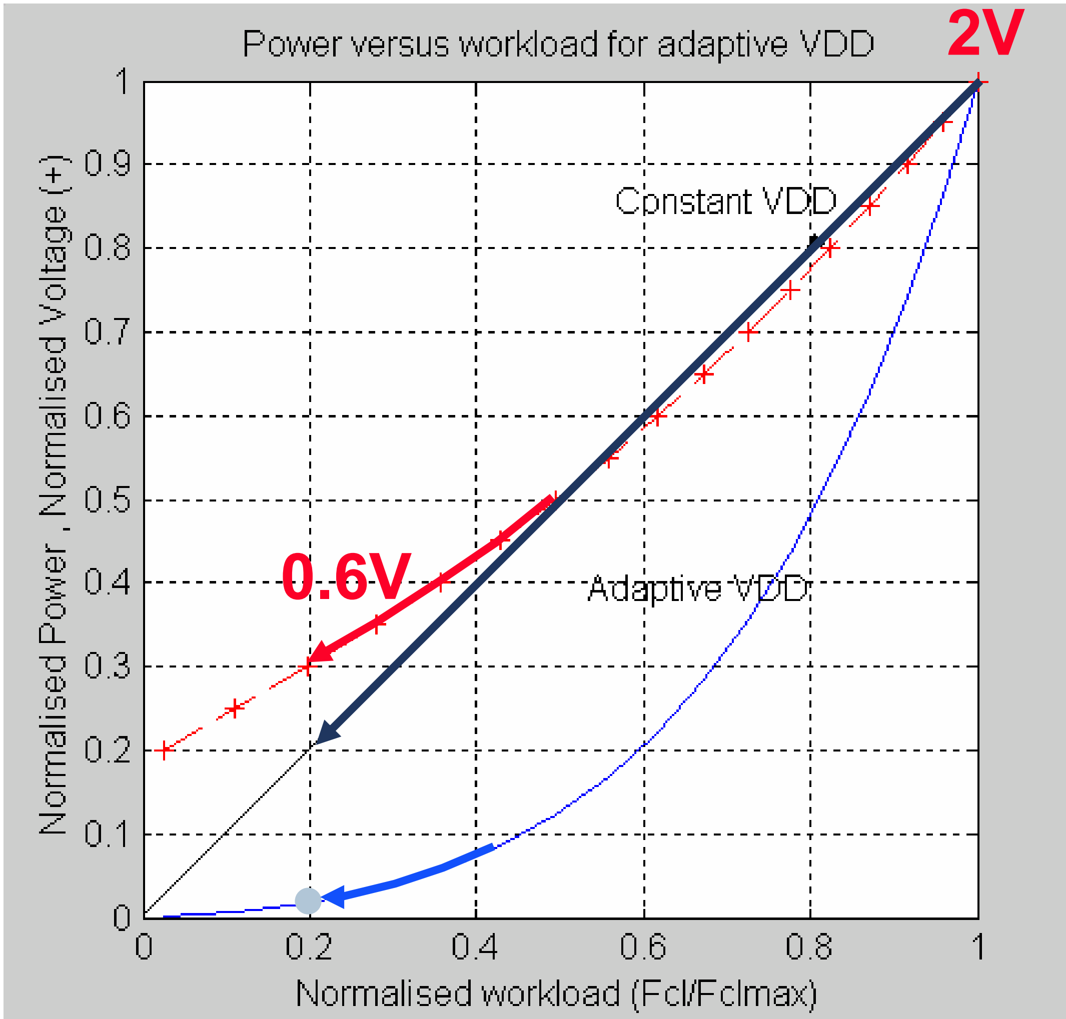




Which method can handle dynamism? Let RTOS select other schedule combination



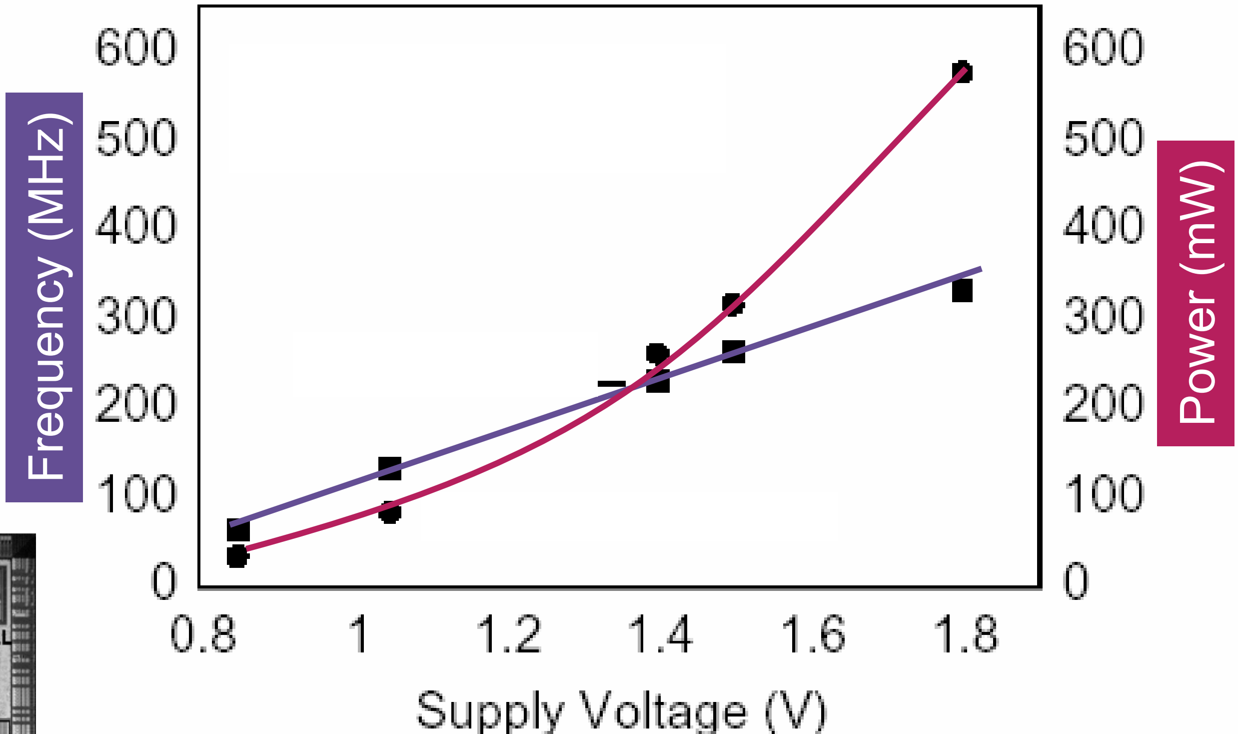
The same method can be applied to use dynamic voltage and frequency scaling

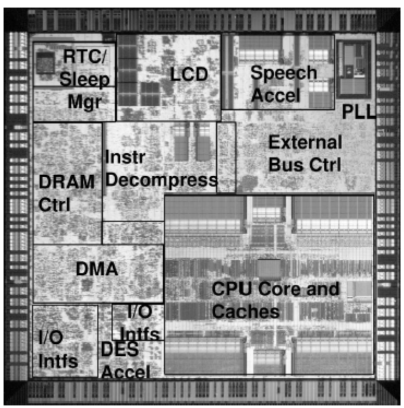


0.18 μm
V_{DDmax}=2V
V_T=0.35V

Power PC[IBM] with dynamic voltage and frequency scaling

17:1 power range over 5:1 frequency range





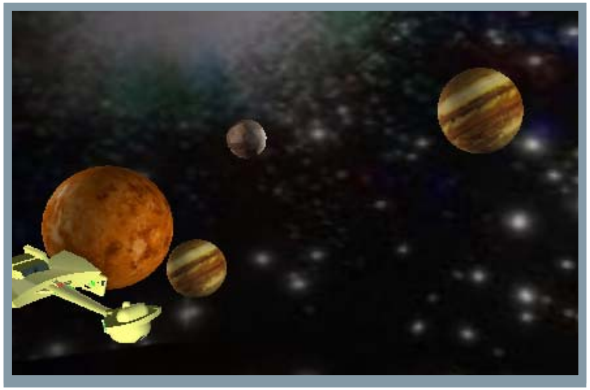
Dynamic voltage adaptation offers substantia energy reduction… if applied well

Power-efficient mapping of a dynamic application on (multi-)processor platforms with multiple Vdd’s

reference energy

state-of-the-art

TCM



TCM

memory

CPU 2.4V

memory

CPU 1.2V 2.4V

memory

CPU 1.2V

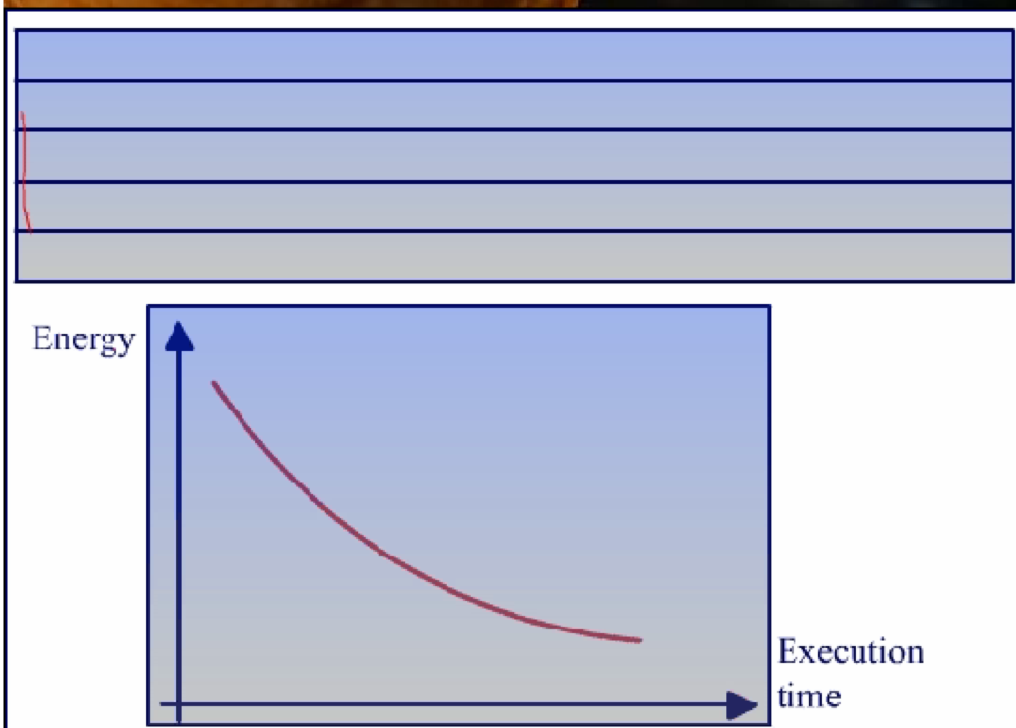
CPU 1.6V

CPU 2.0V

CPU 2.4V

How is this dynamic voltage scaling implemented?

- At design-time, generate optimal scheduling and assignment possibilities (Pareto curve) as fast as determining a single schedule
- At run-time, decide the optimal operating point (i.e. schedule) on the Pareto curves



animatie pareto.avi

What did we learn?

Determine the Pareto space
at design time

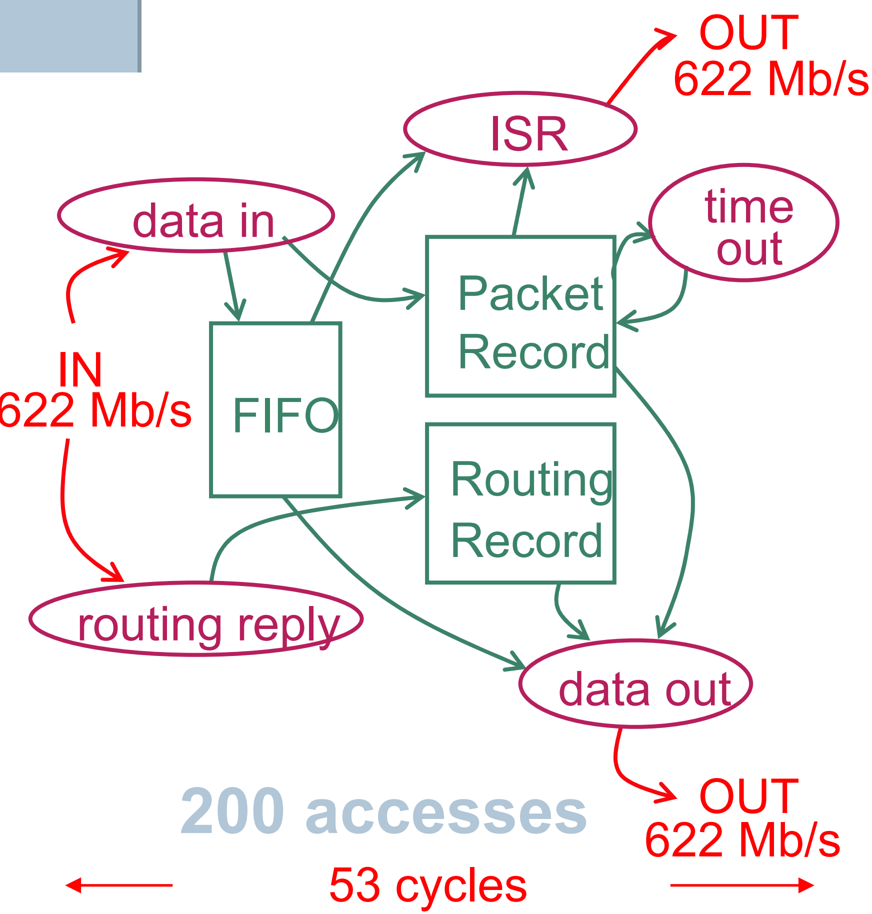
Pass the Pareto space to a
run-time kernel which
decides on the operating
point depending
on the task actually running

Next, we'll see whether this dynamism happens in real-life embedded applications…

Managing Dynamic Real-Time Tasks

- A view into the future
- Properties of ambient multimedia platforms
- The dynamic nature of ambient multimedia applications
- **Methodology for managing dynamic real-time tasks**
- Applying this methodology to an MPEG-21 QoS application
- Main messages

What are the constituents of an application specification?



Processes

- Dynamic and concurrent processes
- Global/local control
- Non-deterministic events

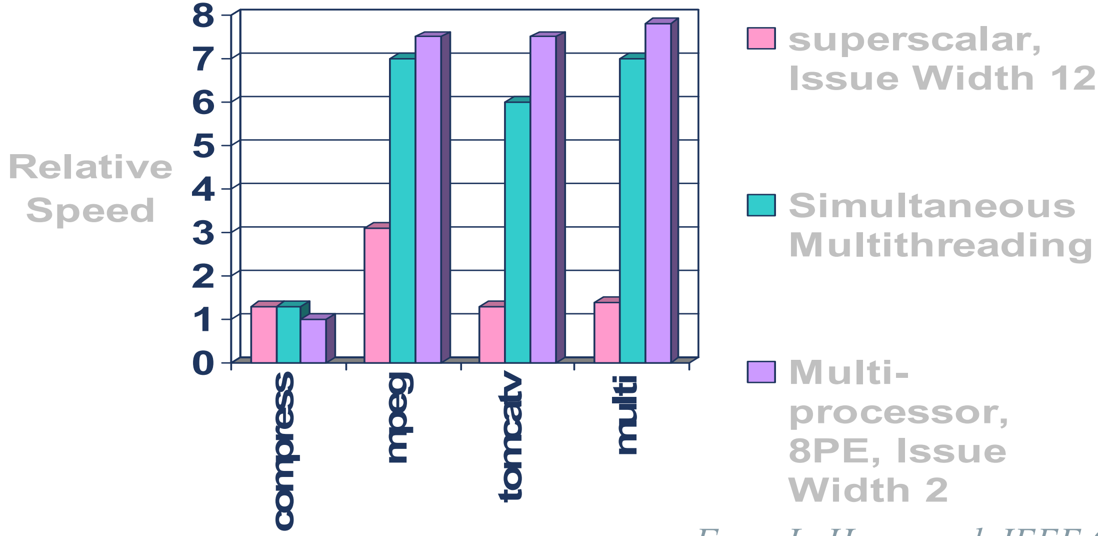
Complex data sets

- Large and irregular dynamically allocated da
- Huge memory accesses

Stringent real-time constraints

Network layer protocols (ATM, IP)
Dynamic multi-media algorithms (MPEG4/7/21)
Wireless/wired terminals (Internet, WLAN)

Multi-processors are needed, ILP alone doesn't offer enough parallelism



From L. Hammond, IEEE Computer, Sept 1997

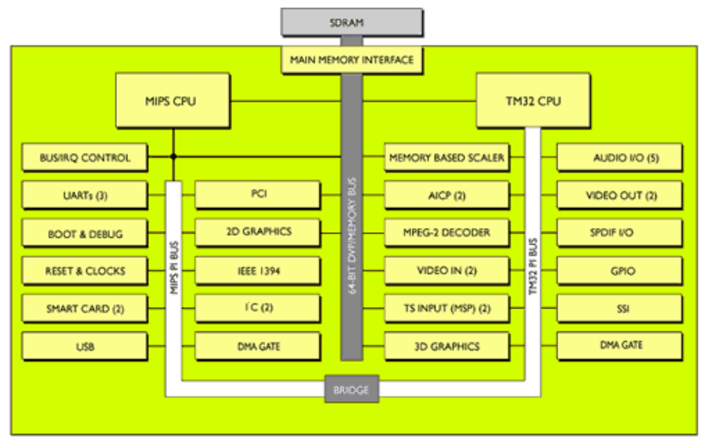
Advantages of multi-processors:

- Performance: possibility to exploit thread level parallelism combined with ILP
- Energy: low energy cost per instruction by customizing the nodes (ASIPs) + effective memory hierarchy and distributed customisable organisation
- Flexible: programmable nodes
- Scalability: memory bandwidth is scalable (if good memory hierarchy is used)

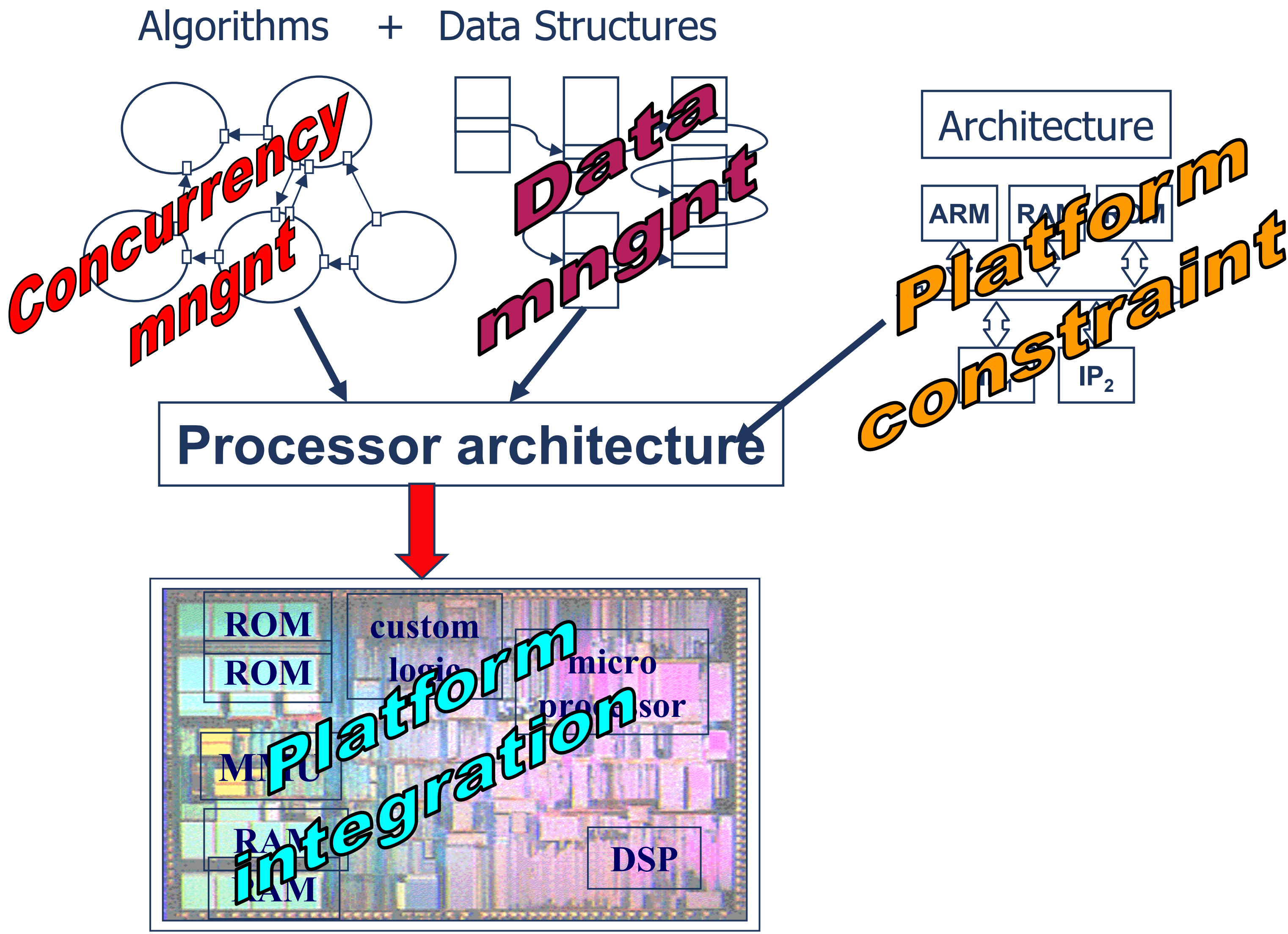
Why aren't we using them now in embedded domains?

Efficient mapping requires a very high design effort when done manually

=> need for a cost-sensitive real-time system compiler



Coming to a methodology to support multi-processors…



Presenting a method for a "SW washing machine"

Algorithmic level

↓

Task parallel level

↓

Data parallel level

↓

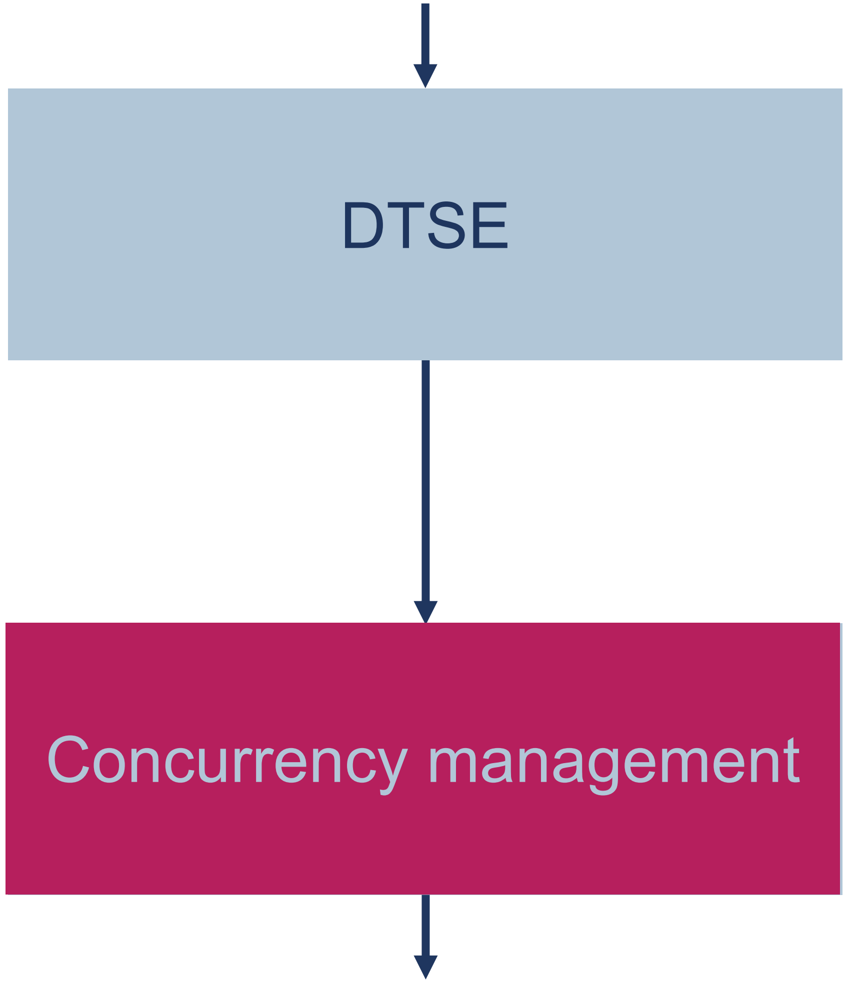
Instruction parallel level

↓

Synthesis & compilation

The dynamic behavior should be analyzed at this level

Each of the “parallel” levels consist of two steps: DTSE & Concurrency Mgmt

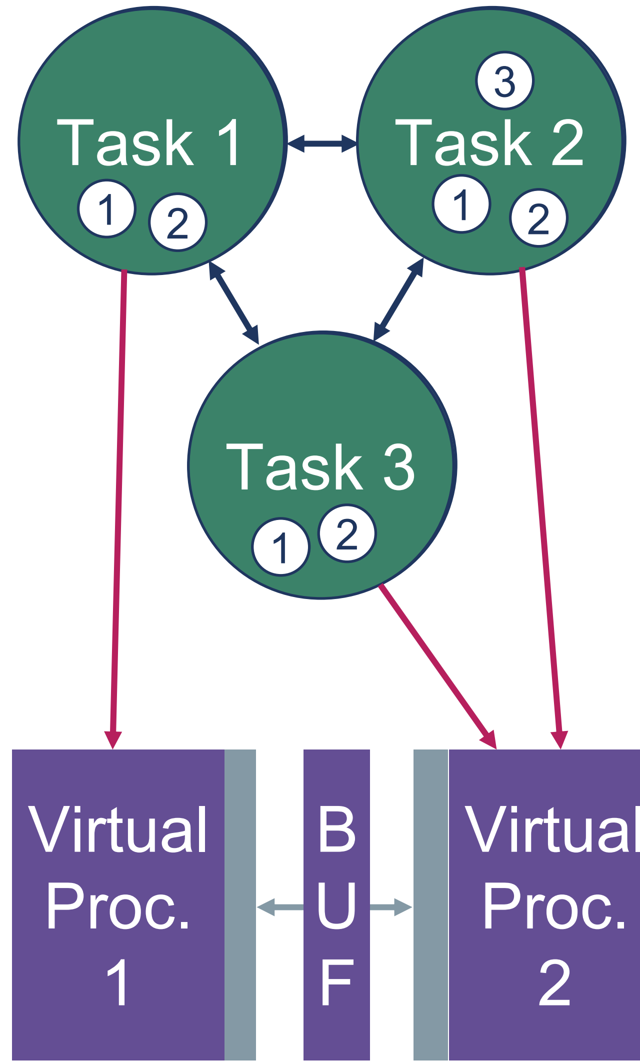


Reduce power consumptior by reducing the number of memory accesses to large memories and by reducing memory sizes

Reduce power consumptio by fixing the operating poin on the Pareto space at run-time

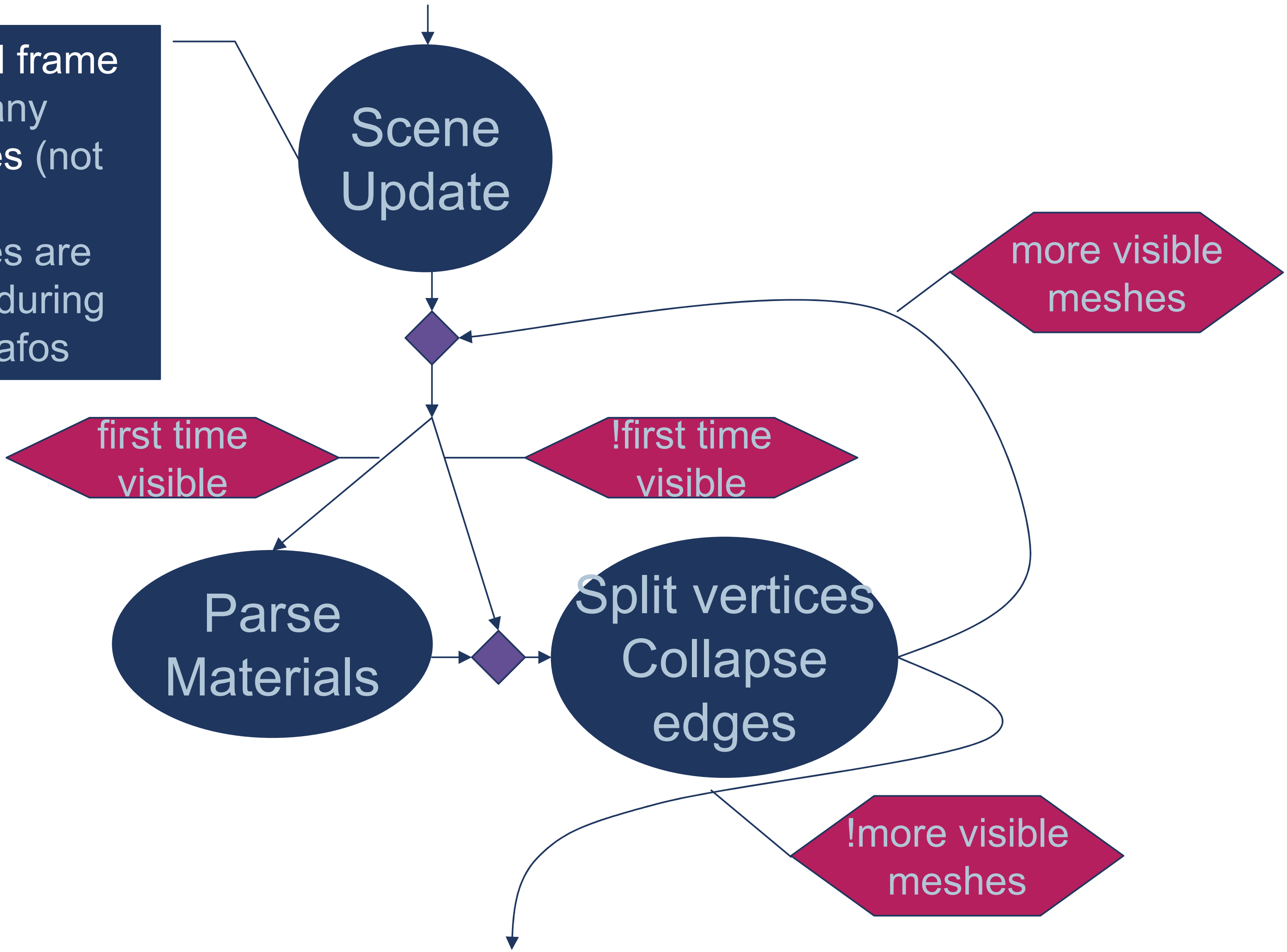
What should be done at the Task Parallel Level?

- Extract concurrent tasks
- Transform the code to improve concurrency
- Schedule threads and tasks
- Allocate virtual processor
- Assign tasks to processor
- Refine the inter-task interface

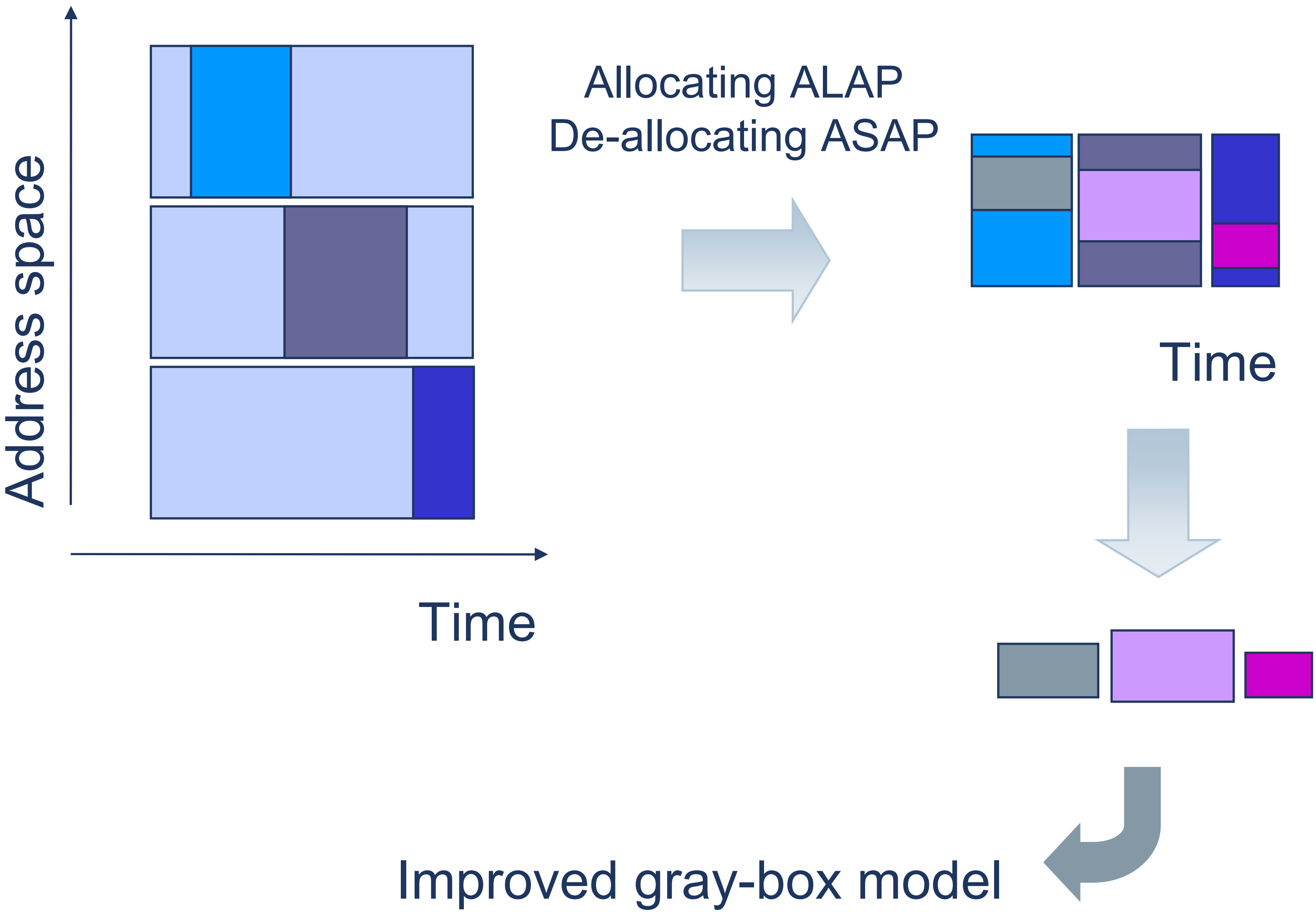


Let's go through the methodology. Build task-level gray-box model of dominant part.

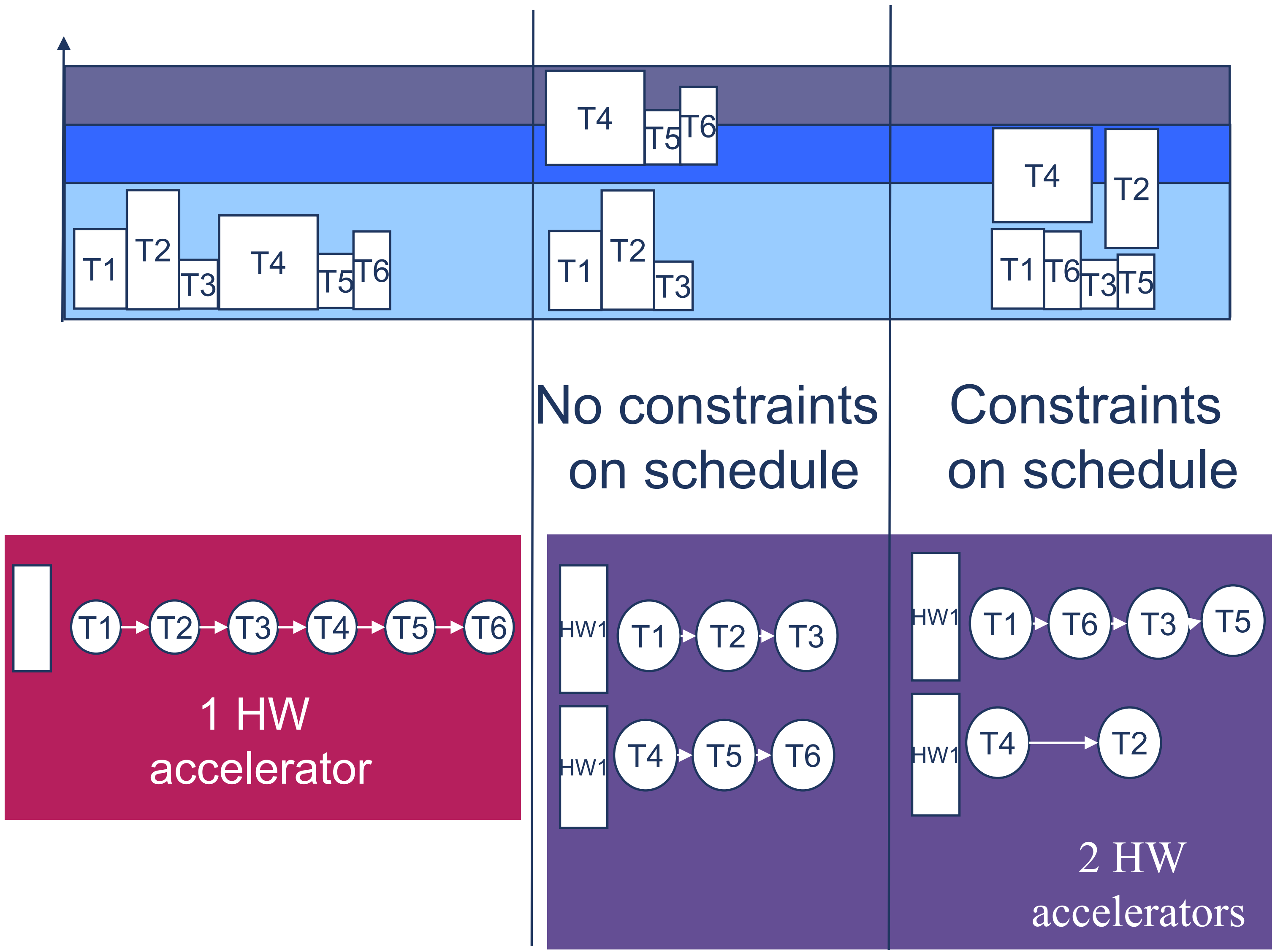
Each thread frame contains many thread nodes (not shown); thread nodes are not splitted during task-level trafos



Task-level DTSE: Platform Independent Code Transformations

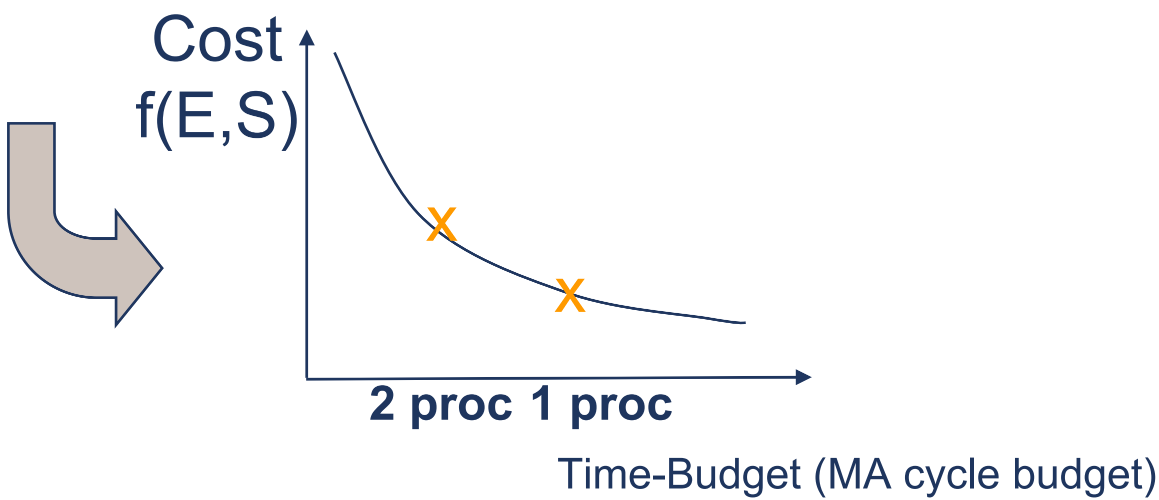


Access ordering and generation of the task (partial) precedence constraints

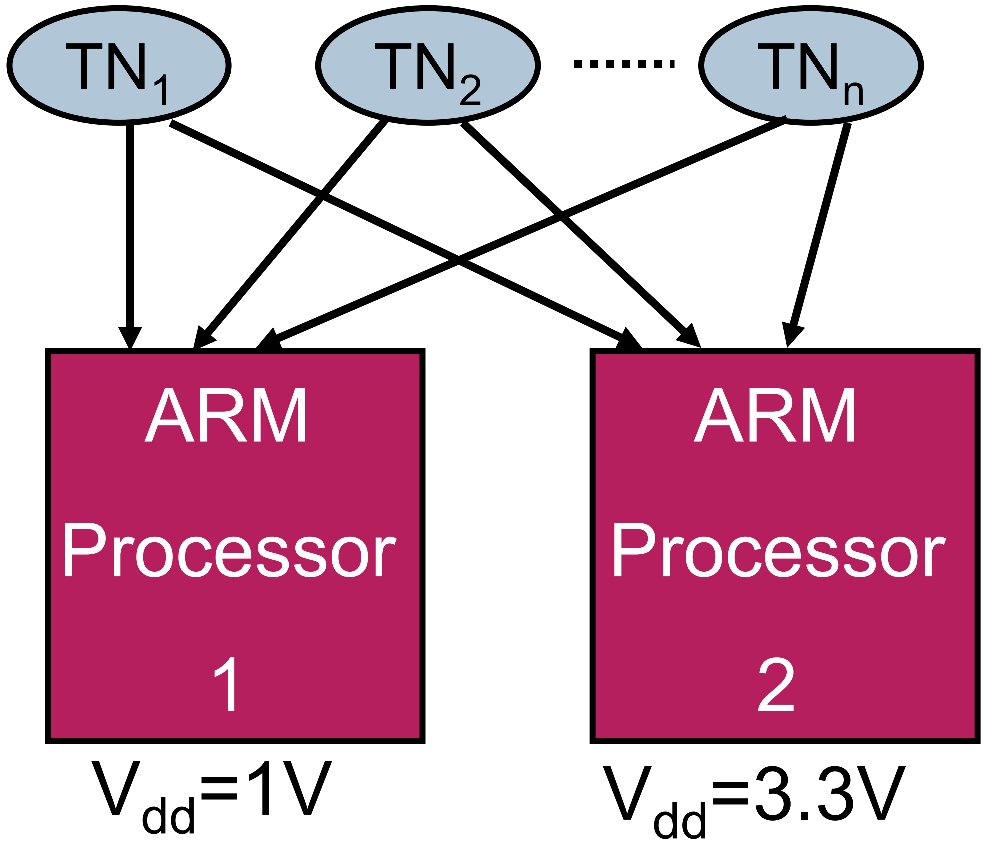


Task-level DTSE reduces power consumption drastically for IM1 player

	Memory Size Pre	*Memory Size Post*	*Memory Energy Pre*	*Memory Energy Post*
1Proc	86.9kB	14 .8kB	0.78mJ	0.16mJ
2Proc	193kB	19.41kB	1.54mJ	0.19mJ

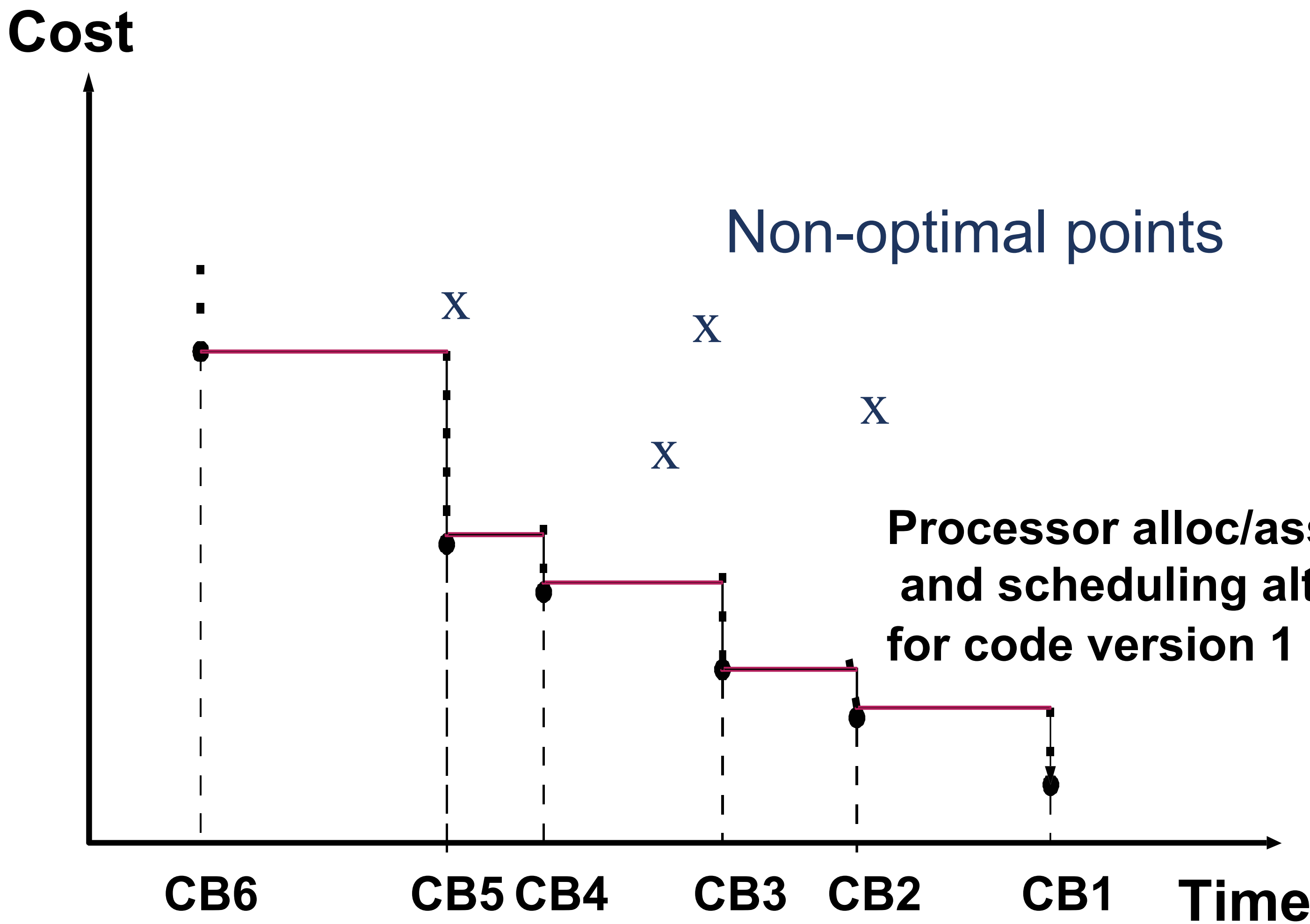


What is the best scheduling of multiple tasks on 2 processors with different Vdd?

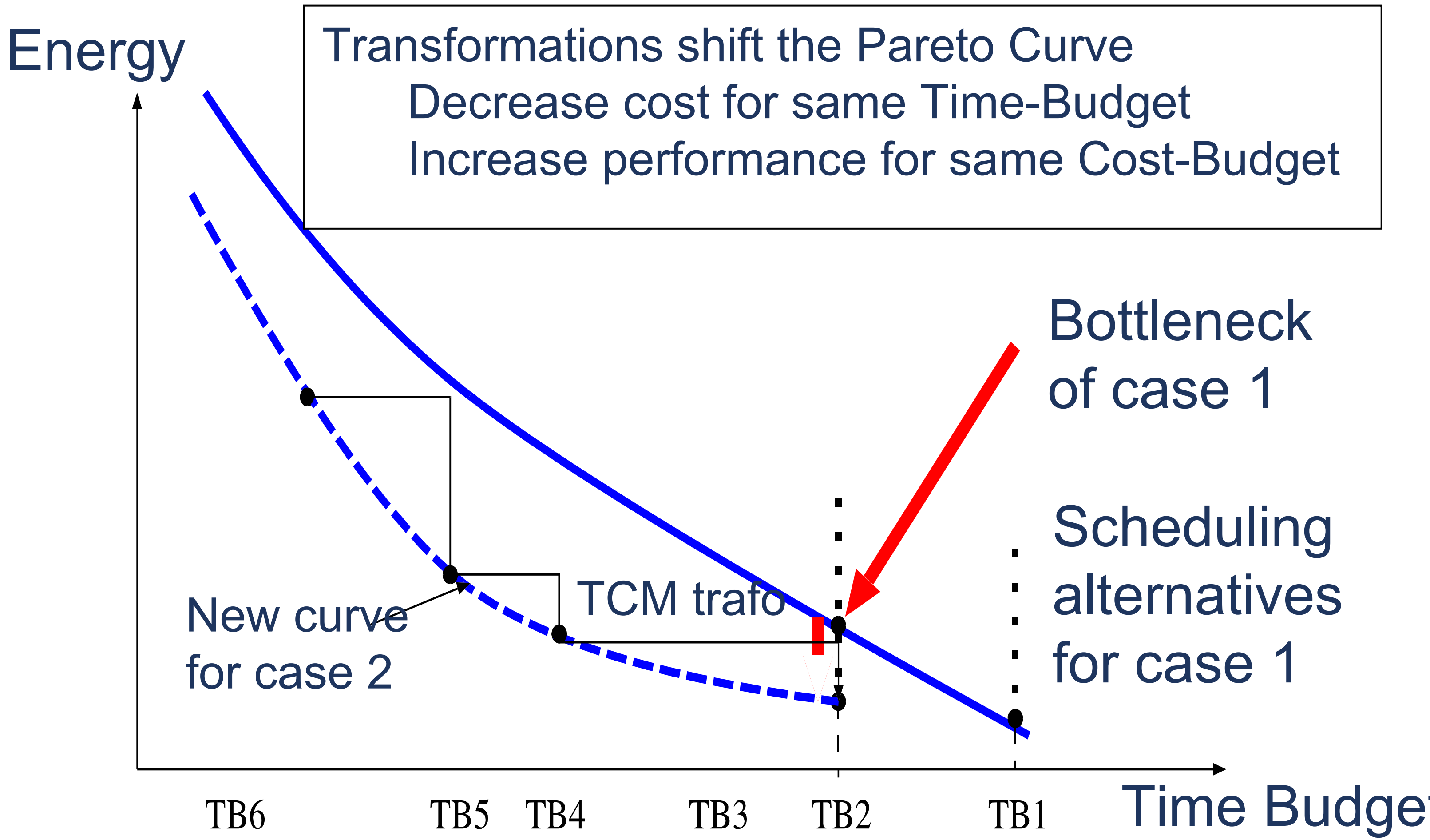


Codes'01, System Design Automation book Verlag'01

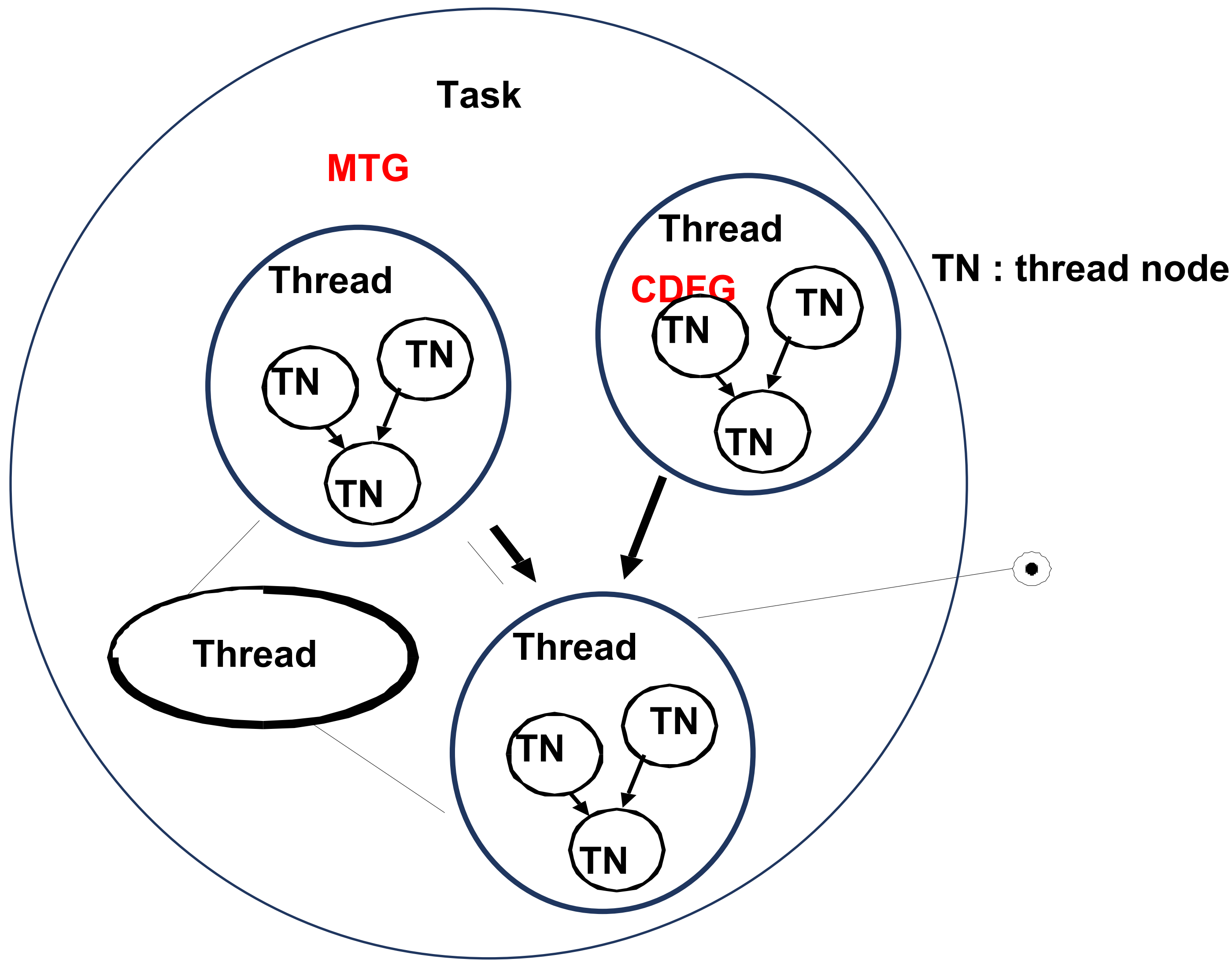
There is no single optimum, but a complete Pareto space of optima



Concurrency improving transformations shift the Pareto curve

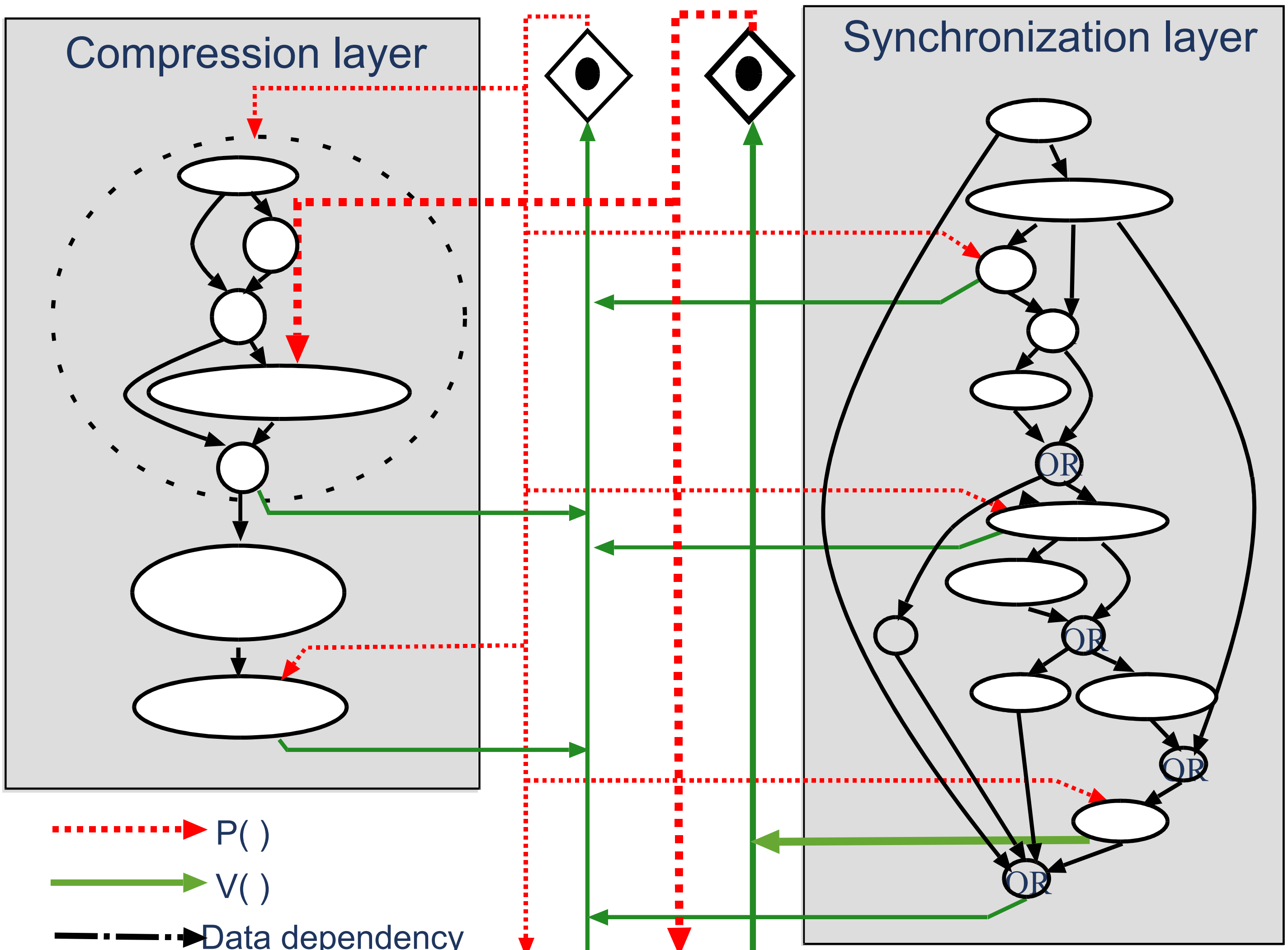


Gray box model: a combination of MTG*(Petri Net) and CDFG model

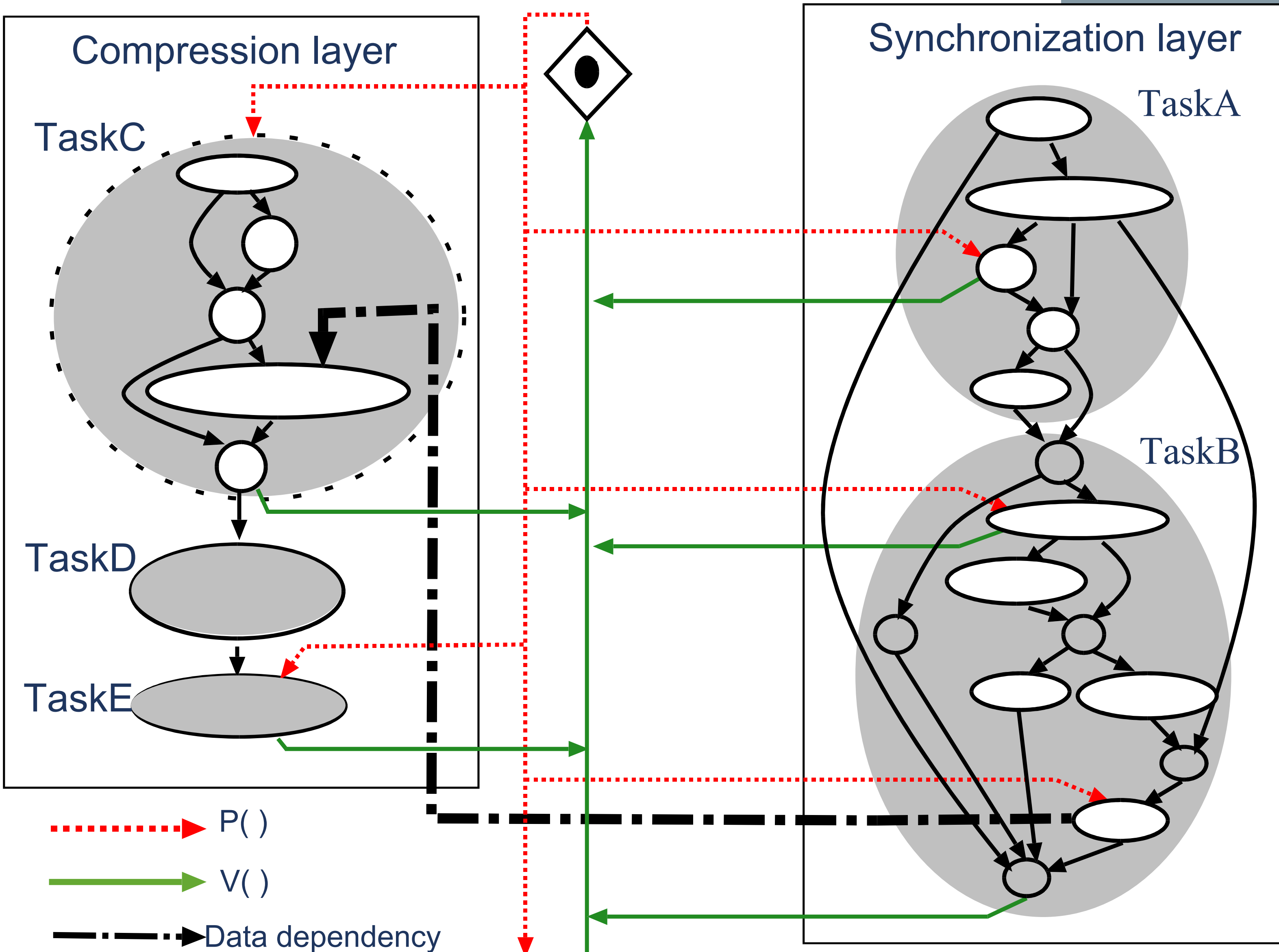


Example of concurrency improving transformations using the gray box of the IM1 player

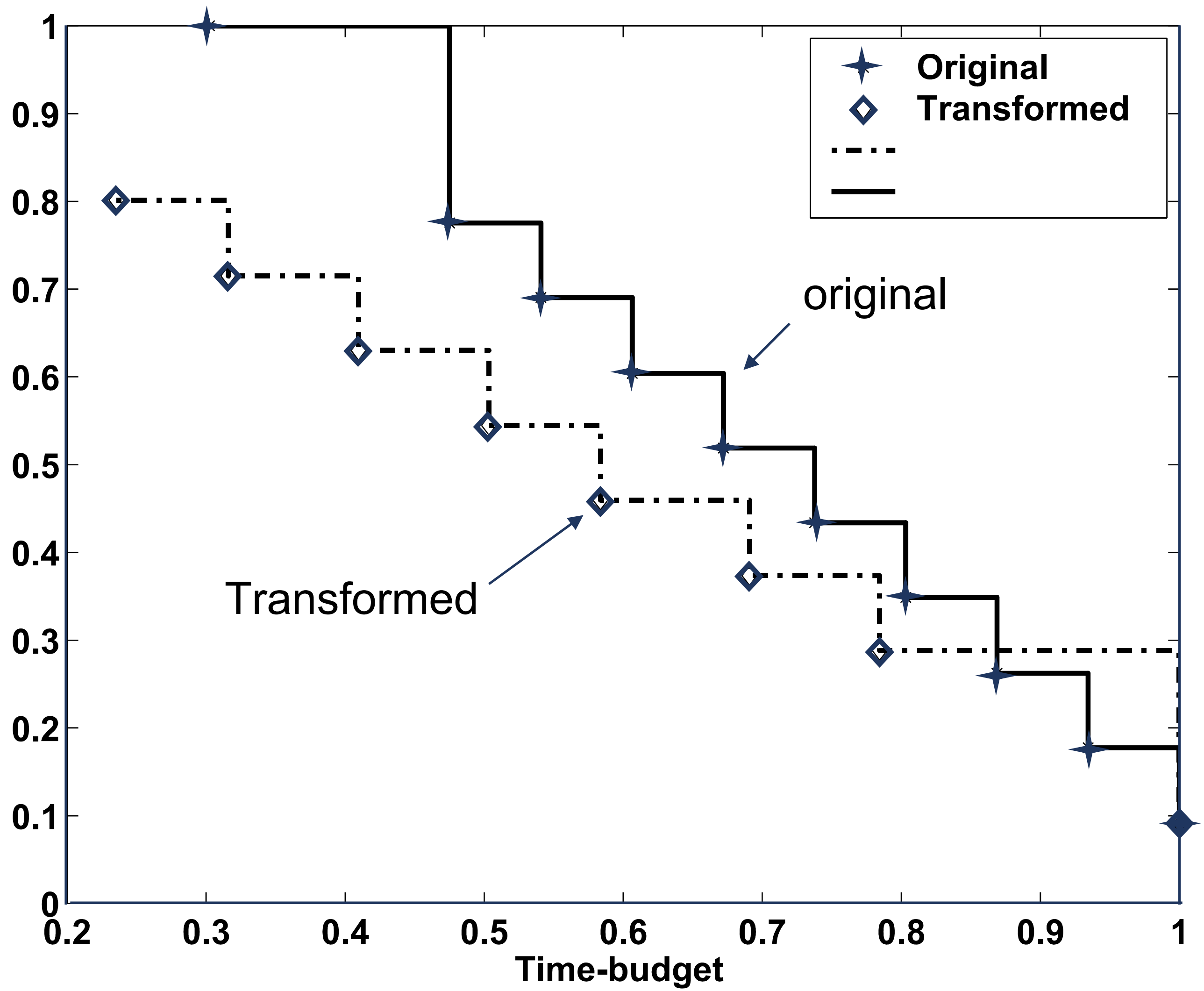
Original Gray-box model:



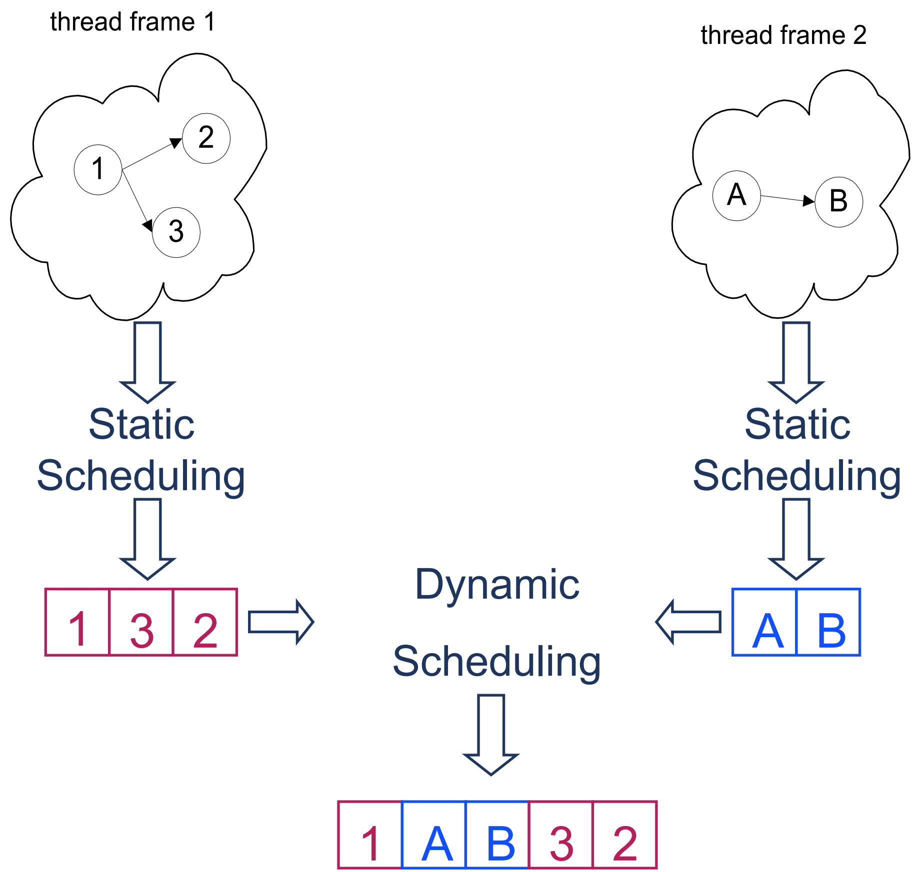
Transformed Gray-box model has more schedul freedom due to removed semaphor



Did we gain anything due to this concurrency improving transformation?



Combine design-time Pareto curves of schedules with run-time schedulers



Design-time scheduling: at compile time, exploring all the optimization possibility

Run-time scheduling: at run time, providing flexibility and dynamic control at low cost

Cases'00, Design&Test- Sep.'01

Managing Dynamic Real-Time Tasks

A view into the future

Properties of ambient multimedia platforms

The dynamic nature of ambient multimedia applications

Methodology for managing dynamic real-time tasks

Applying this methodology to an MPEG-21 QoS application

Main messages

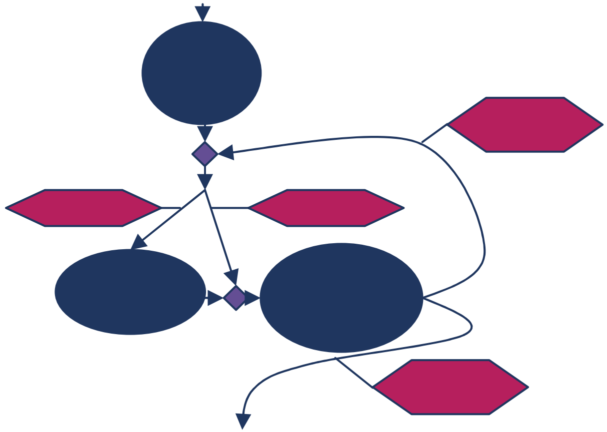
What are the sources of dynamic behavior in the MPEG-21 QoS application?

inter thread frame:

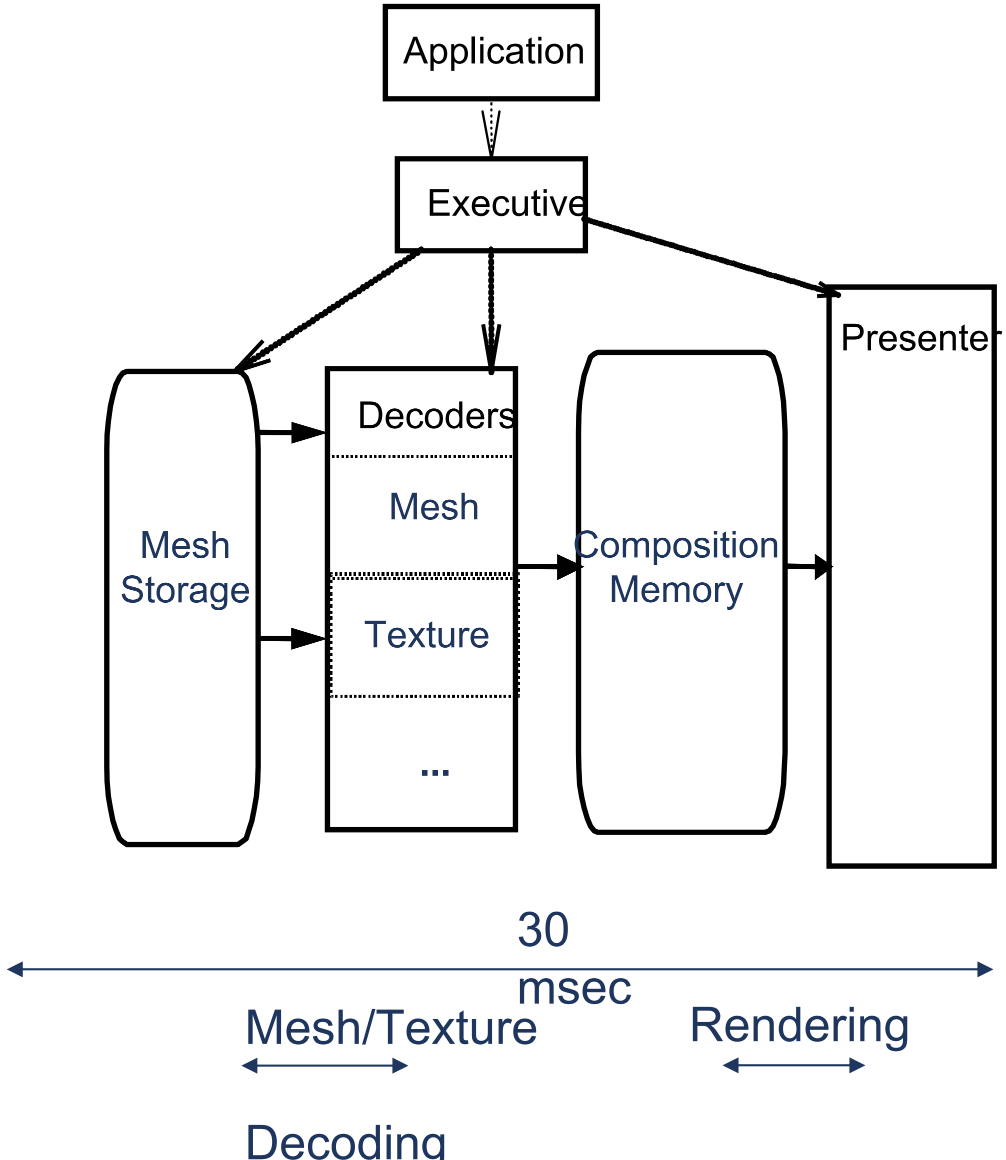
- non-deterministic user event
  - key-press: to load a new scene or to stop the system
- first time a mesh becomes visible
  - extra functionality is activated
- data dependent loop boundaries
  - e.g. number of visible meshes in animated scene

intra thread frame:

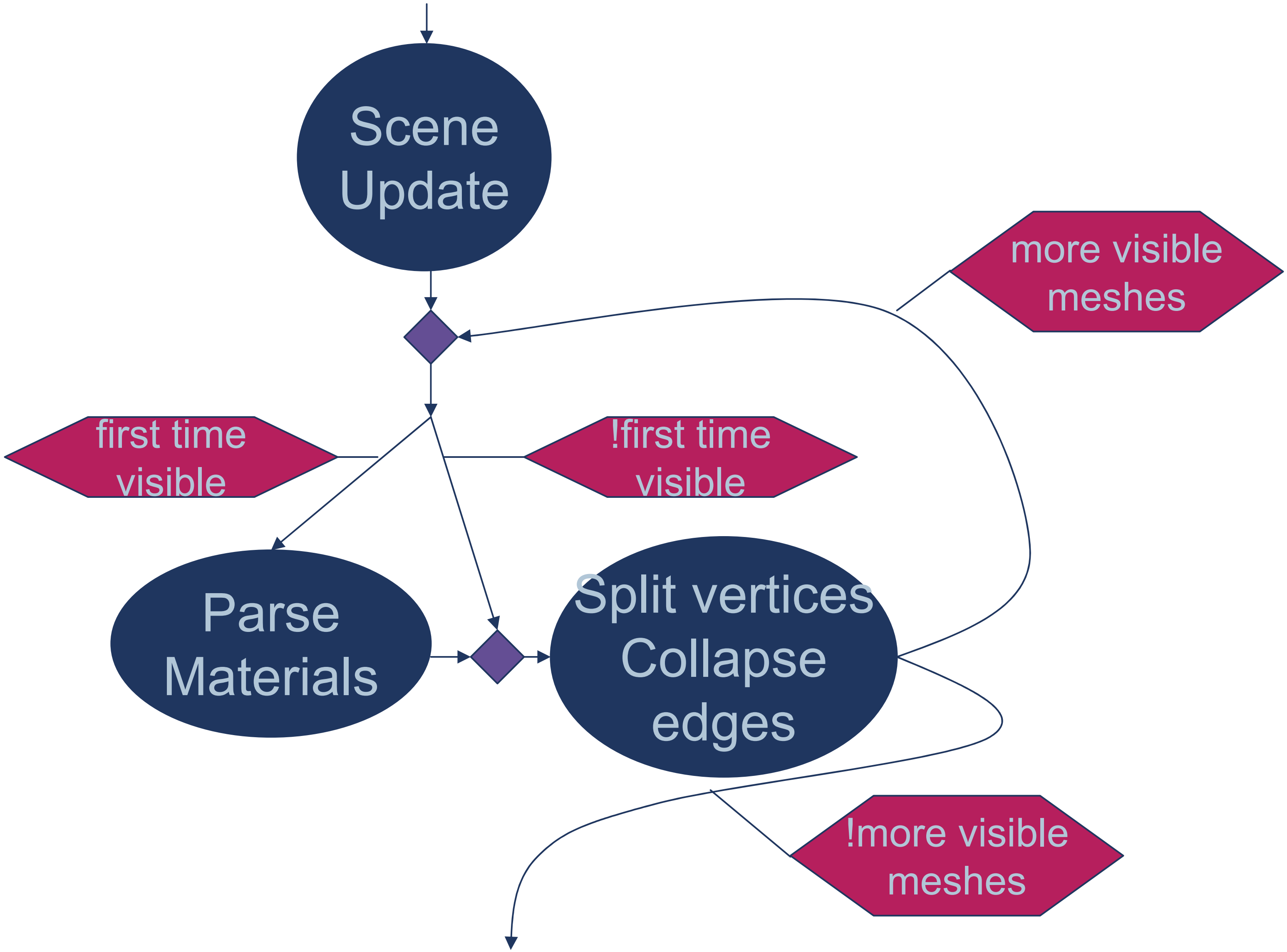
- data dependent loop boundaries
  - e.g. amount of mesh quality change



MPEG-21 high level structure and timing requirements



Task-level gray-box model before concurrency improving transformations.



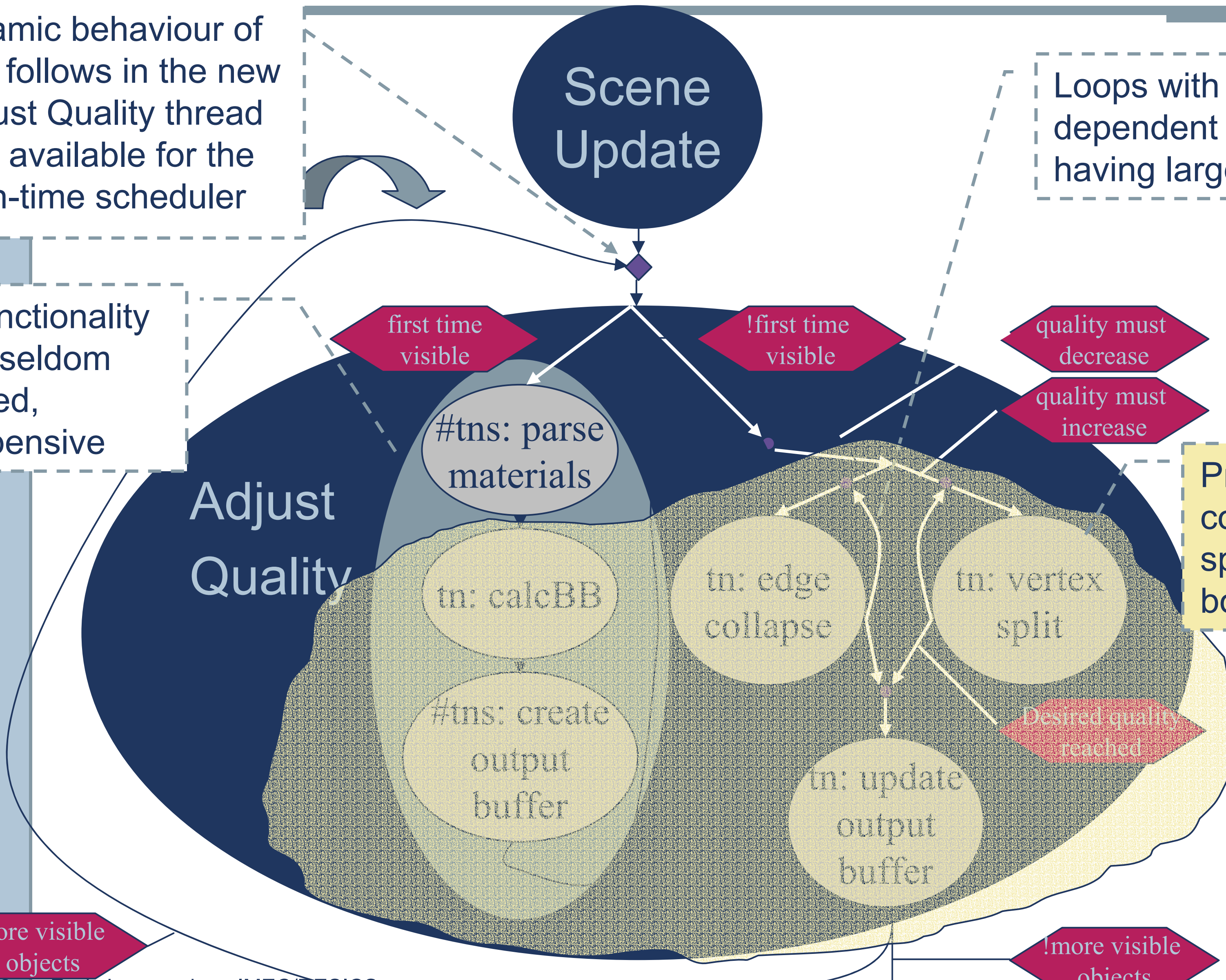
Task-level gray-box model after concurrency improving transformations.

At this point, parameters that describe dynamic behaviour of what follows in the new Adjust Quality thread are available for the run-time scheduler

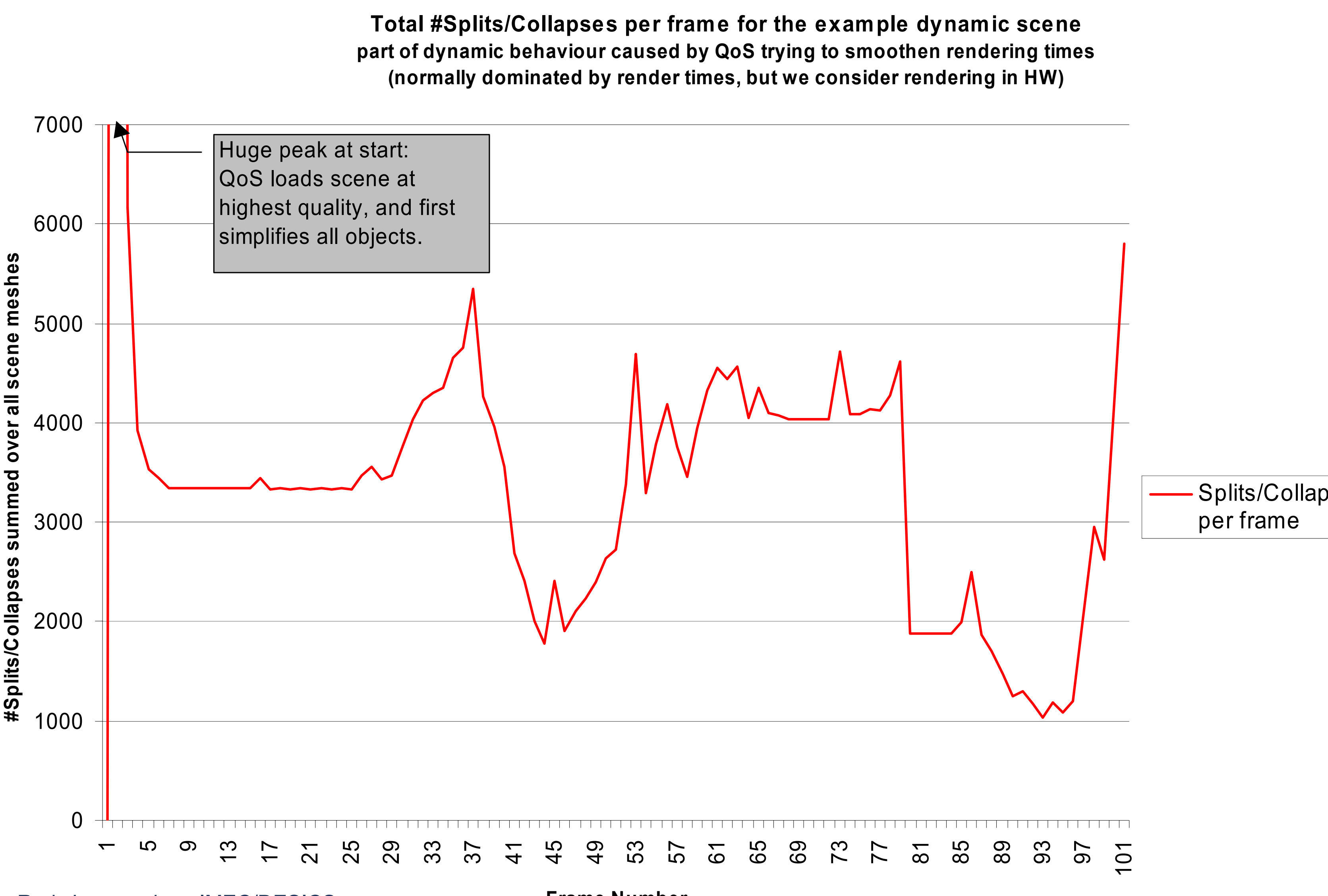
his functionality only seldom ctivated, ut expensive

Loops with data dependent boundaries having large ranges

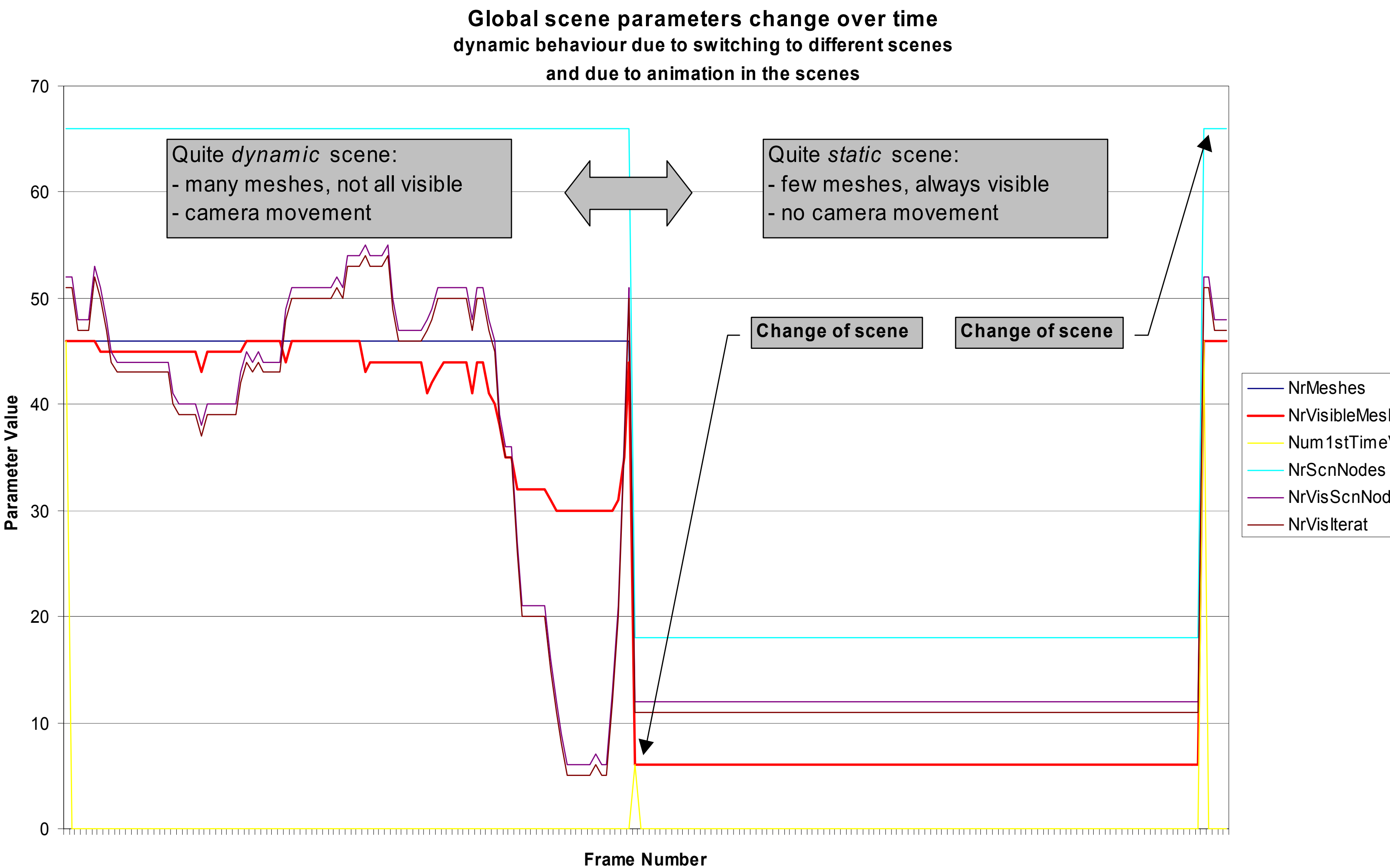
Pre-trafo collapse/ split boundary



More than 7-fold variation in computation due to intra-thread frame dynamic behavior



One order of magnitude variation due to inter-thread dynamic behavior



Managing Dynamic Real-Time Tasks

A view into the future

Properties of ambient multimedia platforms

The dynamic nature of ambient multimedia applications

Methodology for managing dynamic real-time tasks

Applying this methodology to an MPEG-21 QoS application

Main messages

Main messages

Embedded multi-media applications are becoming very dynamic and concurrent in nature
=> RTOS essential

Task Concurrency Management approach provides the flexibility and optimization possibility while limiting the run time computation complexity

A multiprocessor platform with different working voltages potentially provides an energy saving solution

Application-specific run-time scheduling technique combined with design-time scheduling to provide cost-performance Pareto-curve essential for effective solution

SEEDS FOR
TOMORROW'S
WORLD

Rudy.Lauwereins@imec.be

IMECNOLOGY

www.imec.be

Worldwide collaboration with more than 450 companies and institutes.

IMEC – Kapeldreef 75 – B-3001 Leuven – Belgium – Tel. +32 16 281211 – Fax +32 16 229400 – www.imec.be