technische universiteit eindhoven

System Scenarios
Exploiting Dynamism in Embedded System Design

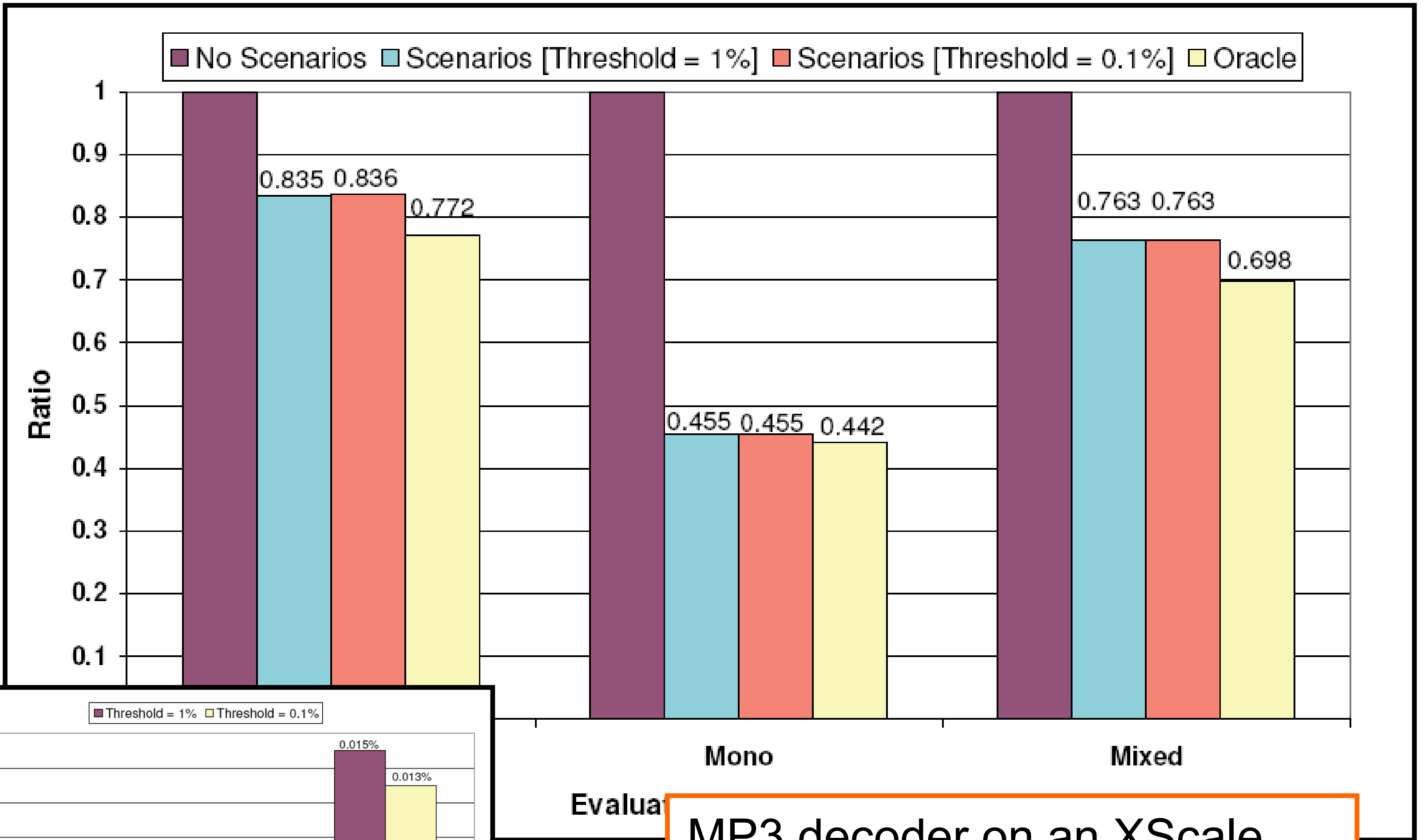
Twan Basten

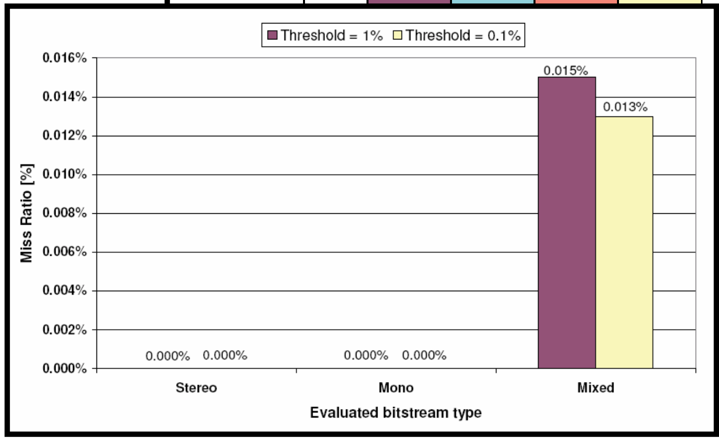
Department of Electrical Engineering
Electronic Systems

'Knowing is not understanding.'
Charles Kettering

2 The Result …

TU/e

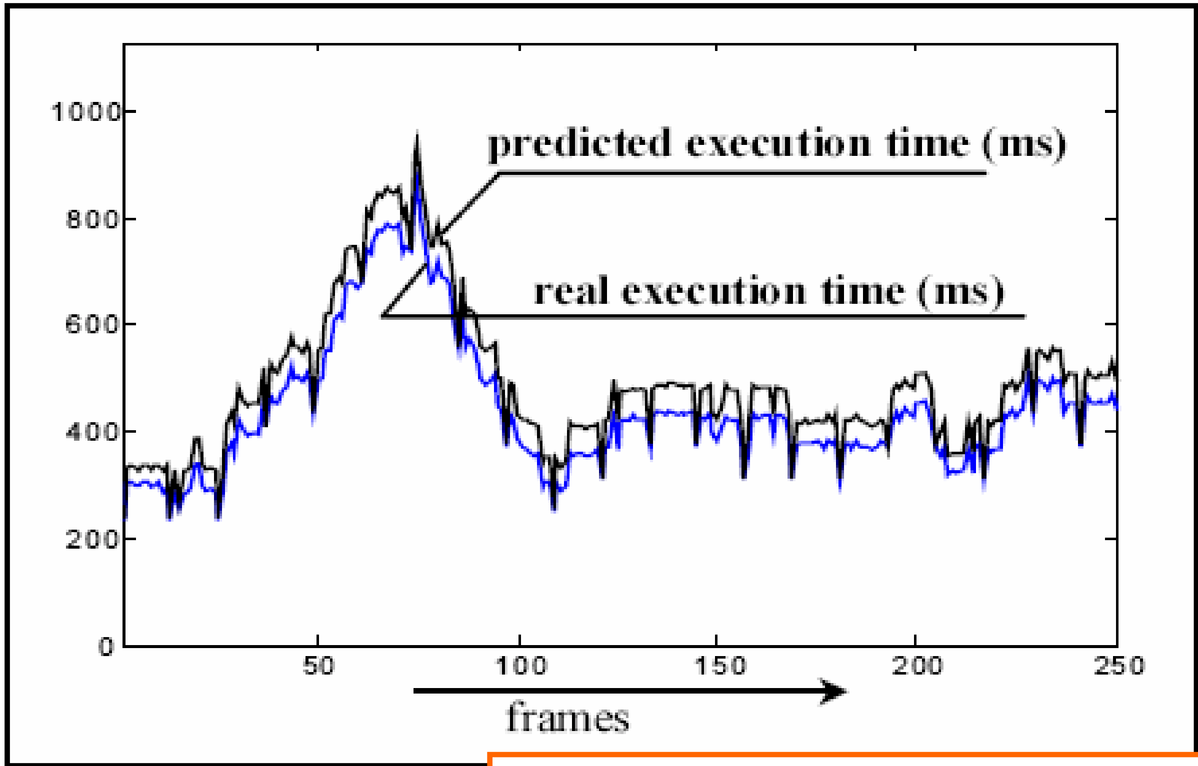




MP3 decoder on an XScale

- 24% energy reduction
- very small deadline miss ratio

3 The Result …



MPEG-4 shape decoder on an MP-SoC

- conservative (on-line) execution time prediction
- average over-estimation within 11% (vs. 60% without scenarios)

4 Joint work with

- Valentin Gheorghita,
  Henk Corporaal, Peter Poplavko

- Francky Catthoor, Stelios Mamagkakis,
  Martin Palkovic, Arnout Vandecappelle

- Koen De Bosschere, Lieven Eeckhout,
  Juan Hamers, Frederik Vandeputte

5 The Essential Idea

design time

Classify system behavior from a cost perspective

run time

Exploit this classification

What Problem does it Solve ?

the

increasingly dynamic nature

of embedded systems becomes an

optimization opportunity

instead of a

potential problem

System Scenario

executions, execution fragments

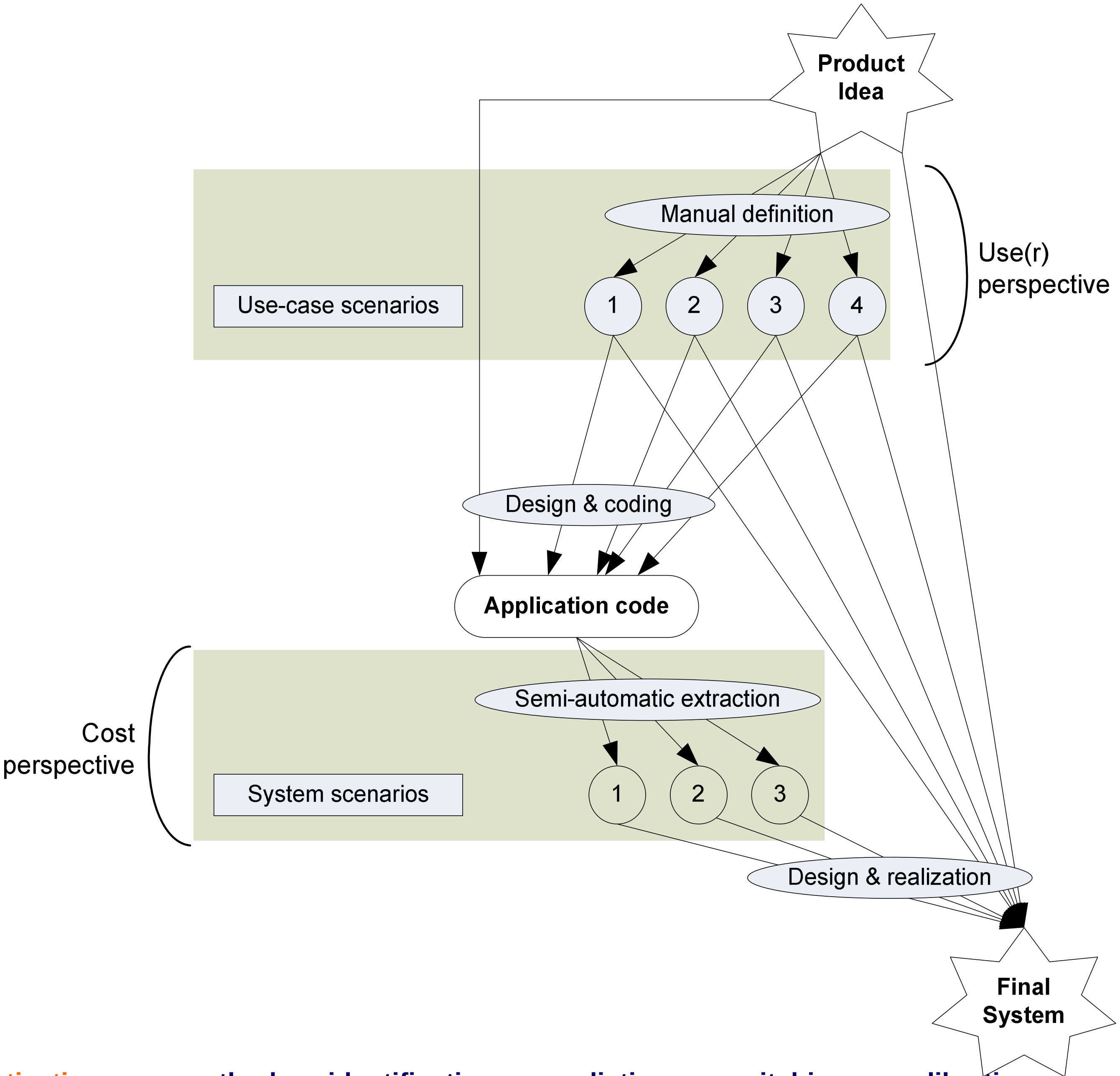
Run Time Situation (RTS)

A set of system behaviors that are similar from a cost perspective

potentially multidimensional

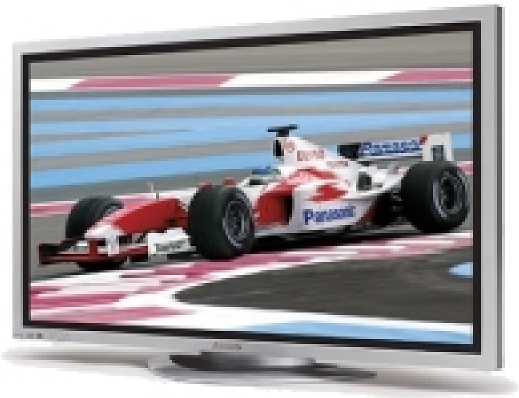
processor cycles, memory accesses, …

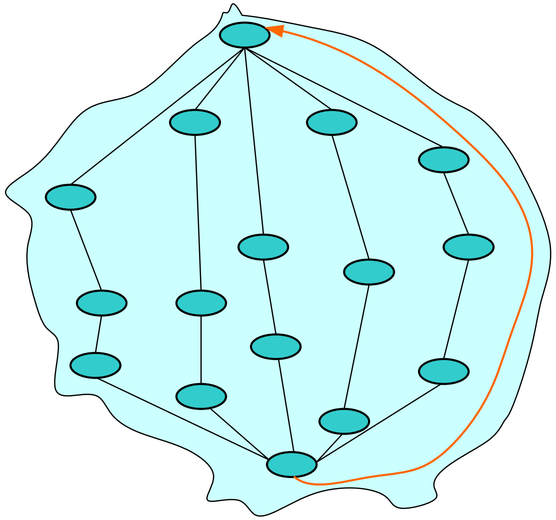
8 What about Use-Case Scenarios ?

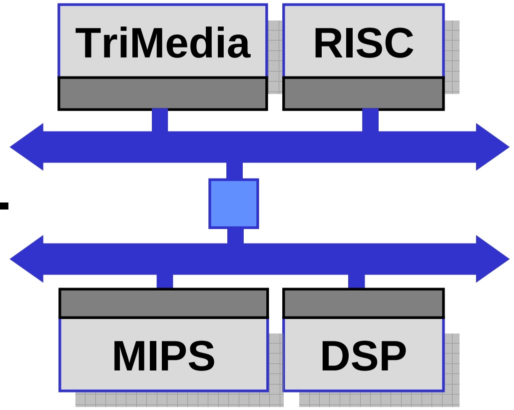


9 Embedded System Design

System = Application + Architecture





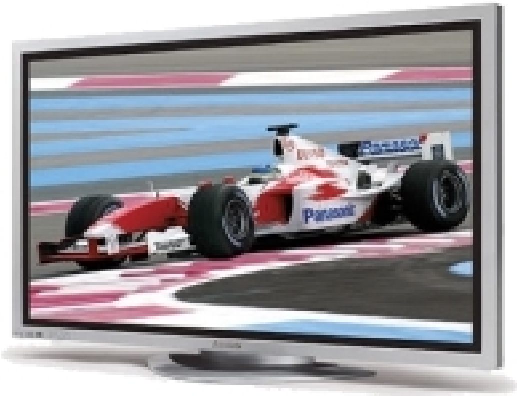


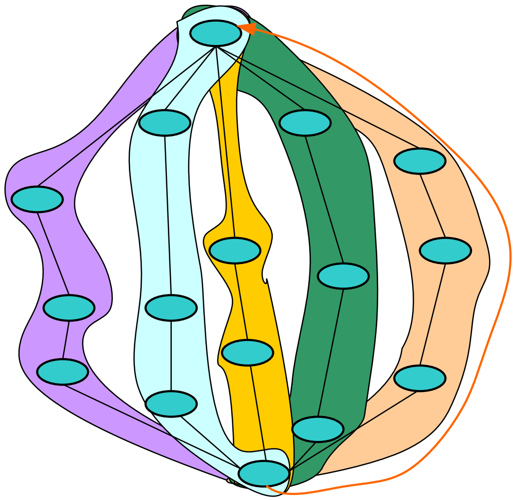
Monolithic worst-case approach

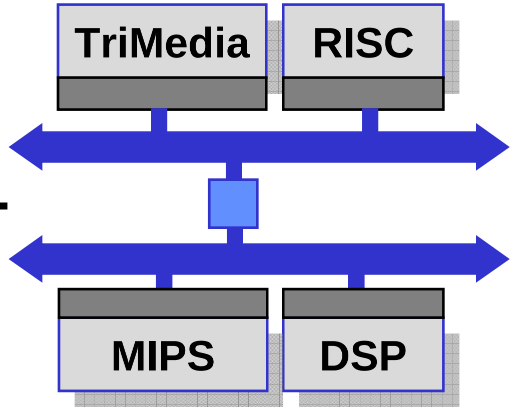
Design time

Embedded System Design

System = Application + Architecture







Fine-grain operation mode approach

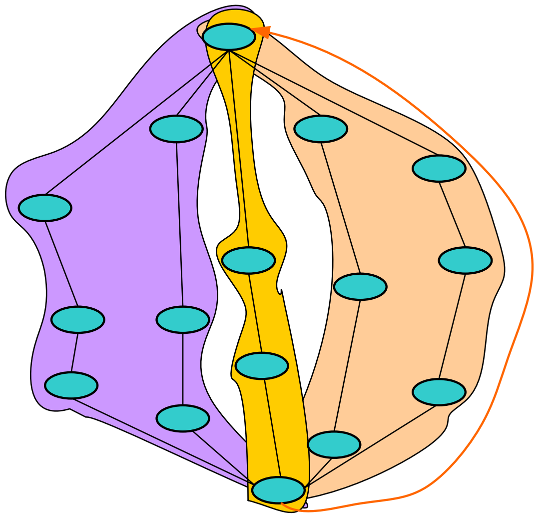
Run time

Embedded System Design

System

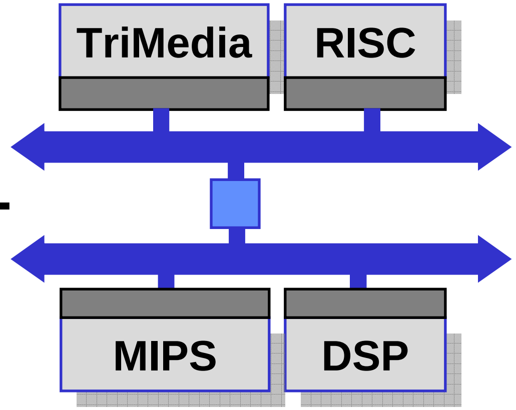
=

Application



\+

Architecture



System-scenario approach

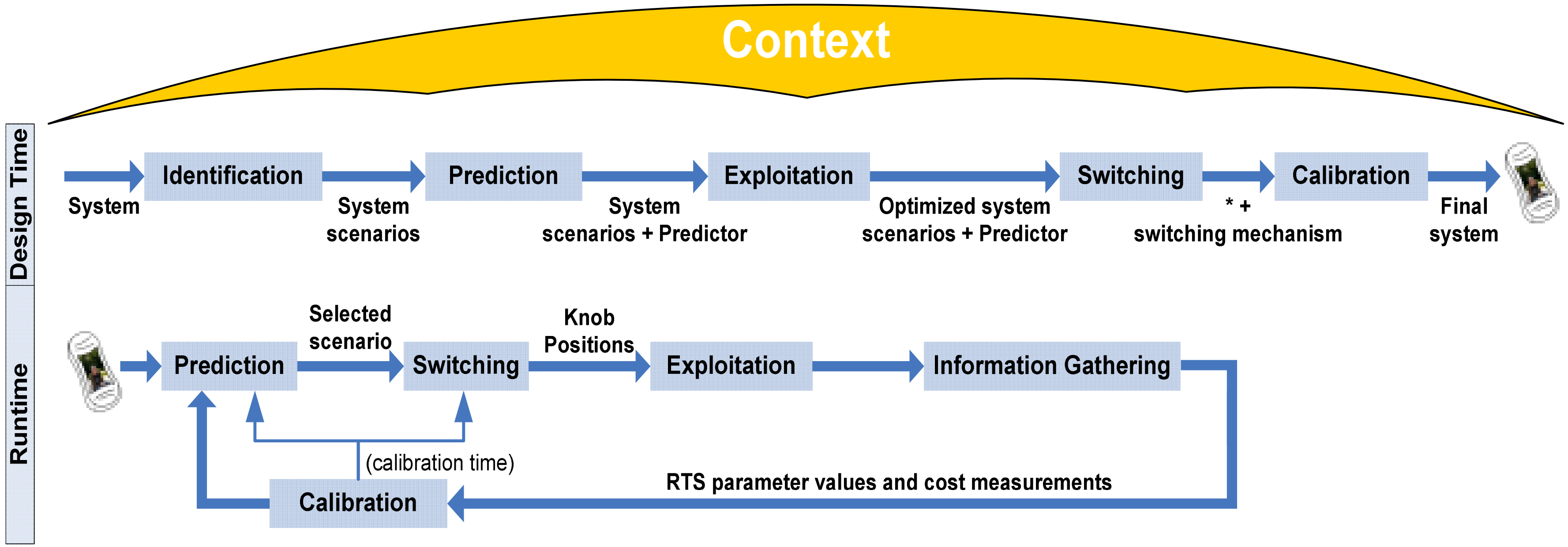
Combined design and run time

Generic Activities

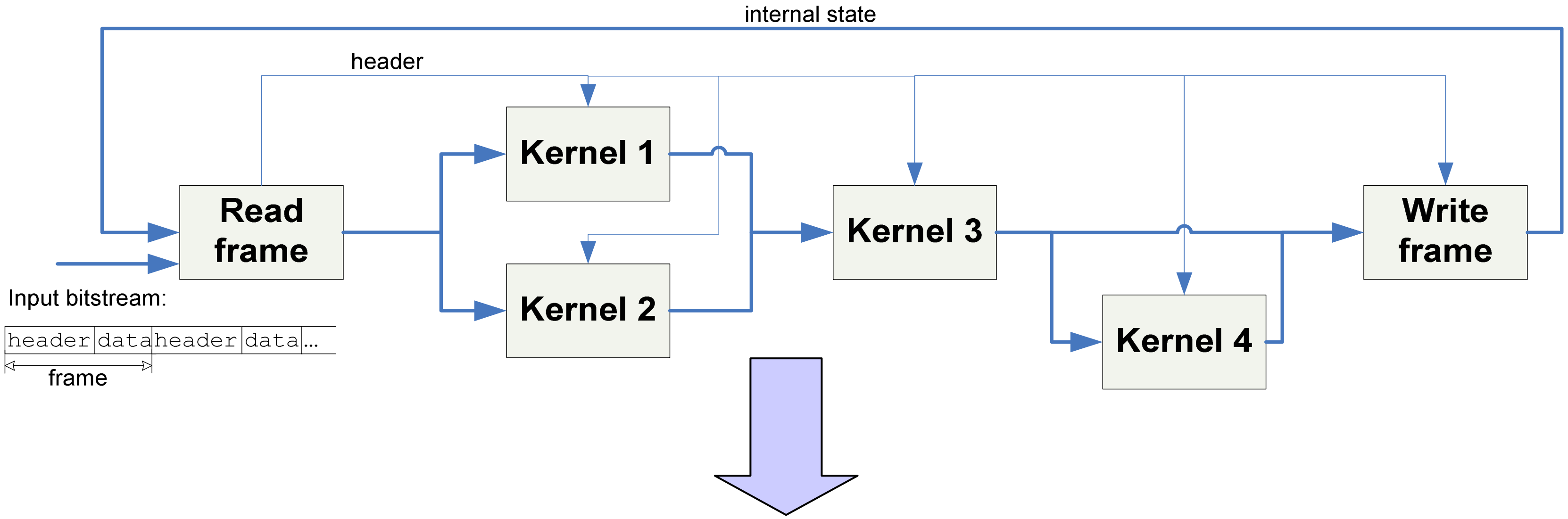
- Identification
- Prediction
- Exploitation
- Switching
- Calibration

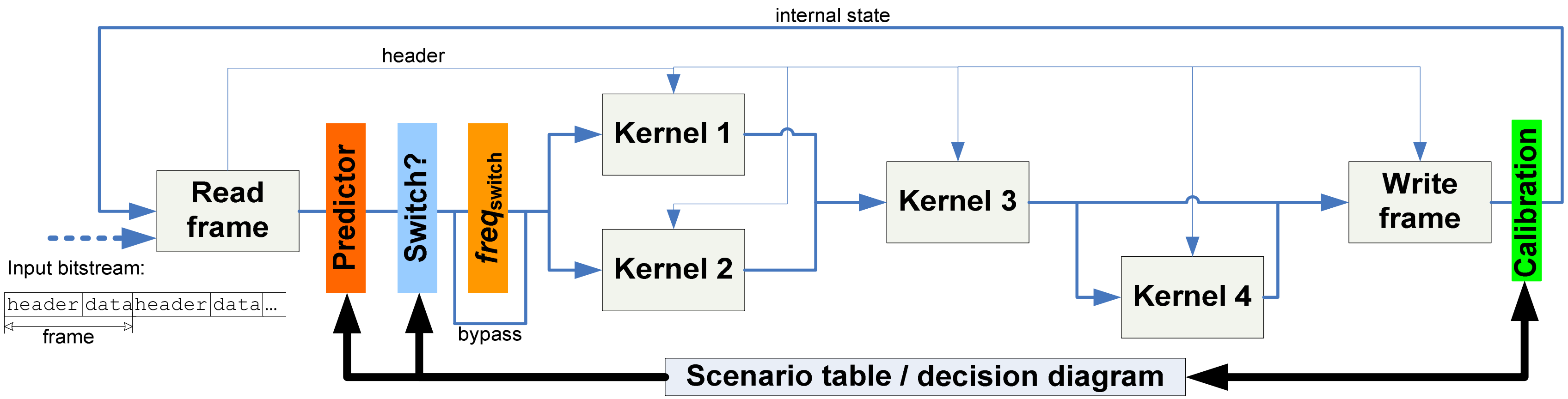
A Generic Method

TU/e



The Final System

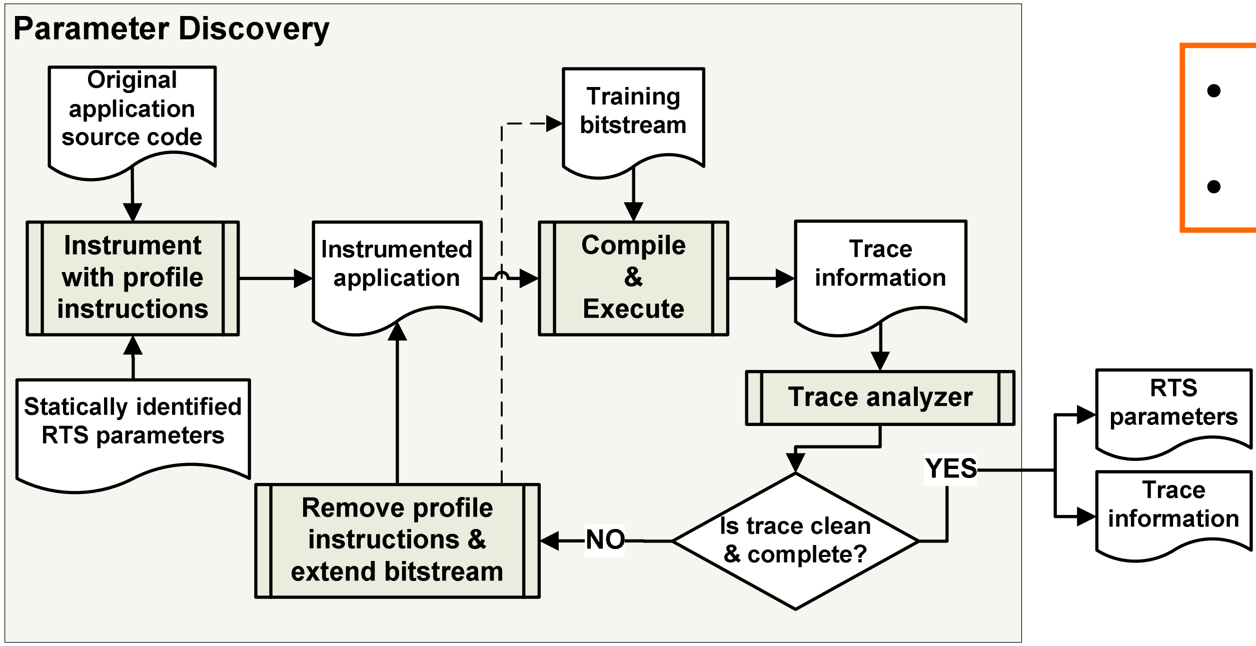




15 The Trade Off

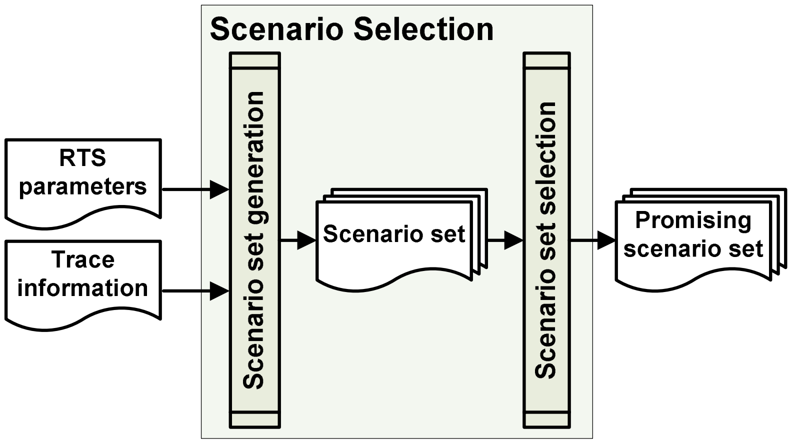
Quality vs. Cost

Identification



- Parameter discovery
- Scenario selection

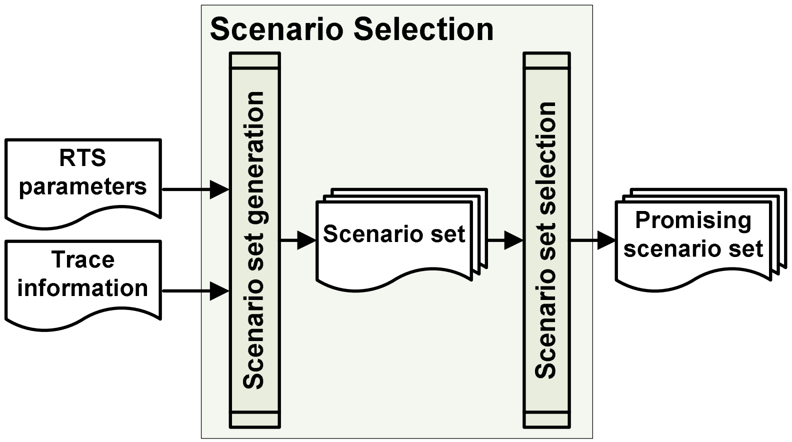
- Static analysis
- Profiling
- Hybrid



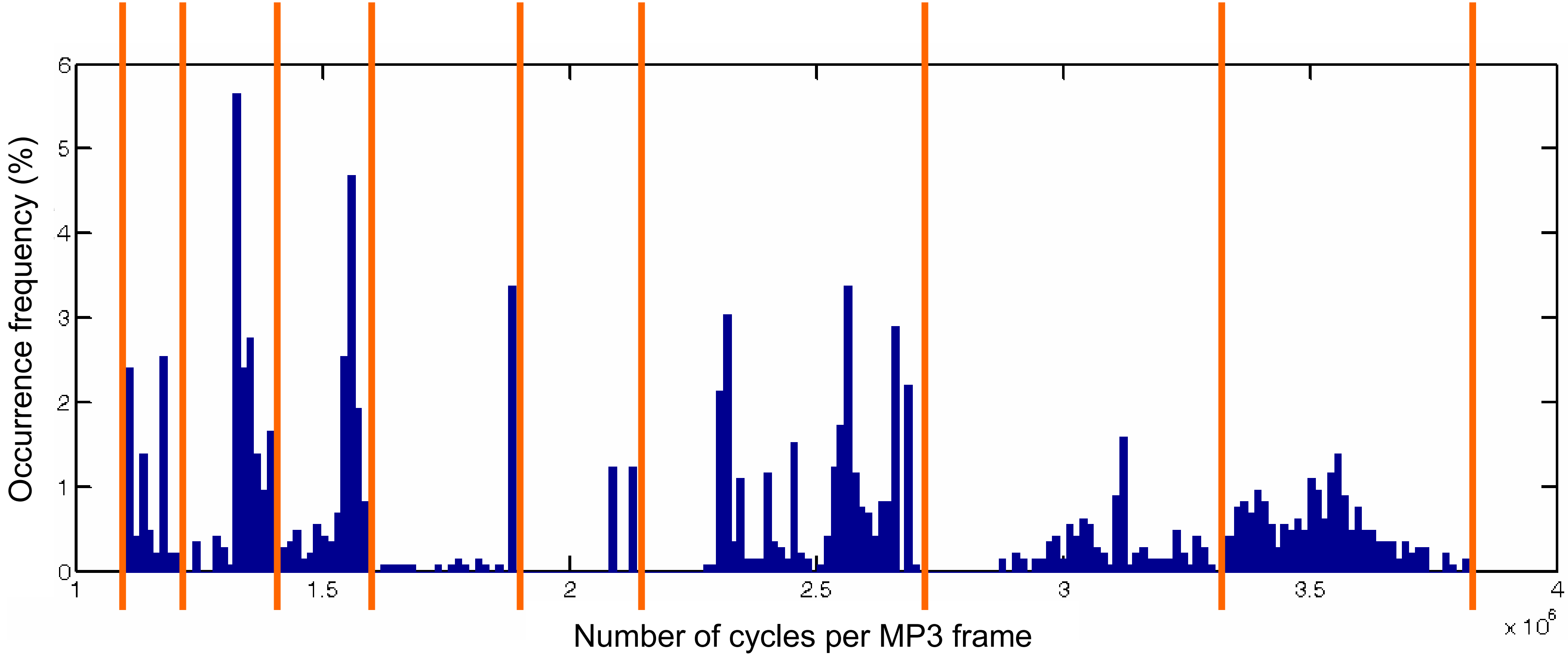
Scenario Selection

Cluster RTSs based on:

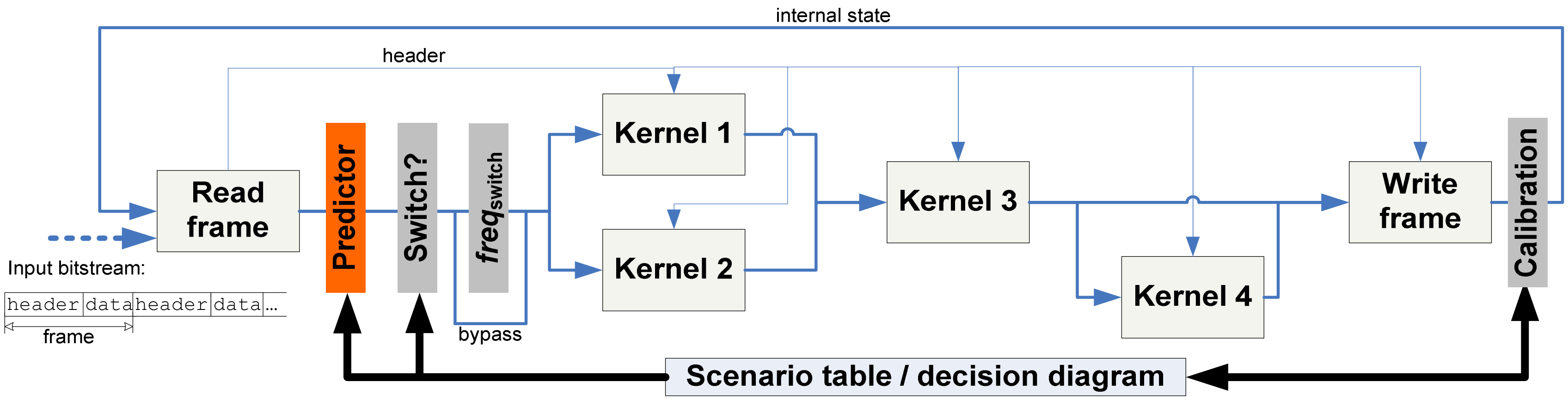
- RTS occurrence frequency
- cost overhead
- switch frequencies
- run-time overhead (prediction, storage, switching)



Scenario Selection



Prediction

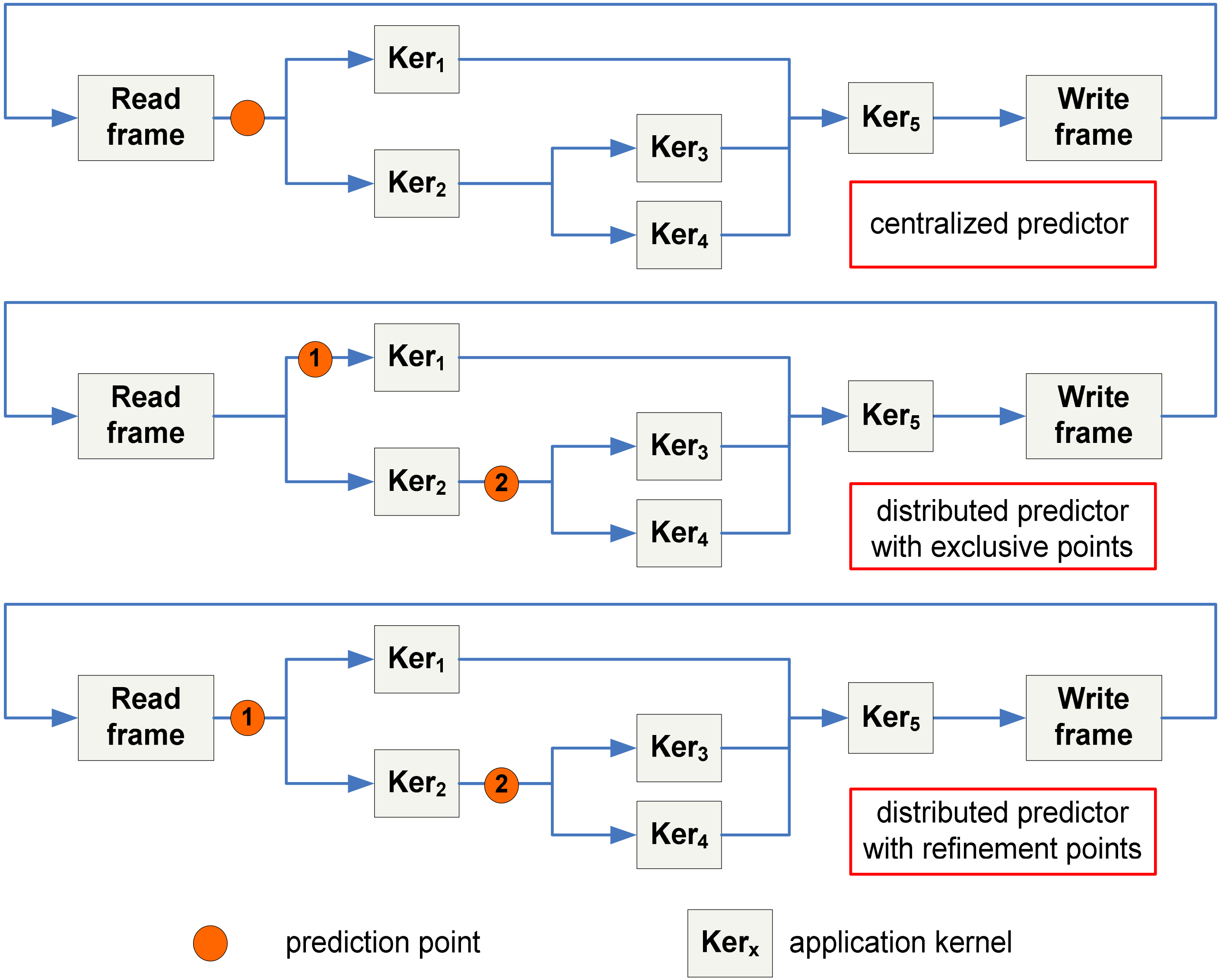

internal state
header
Read frame
Predictor
Switch?
$freq_{switch}$
bypass
Kernel 1
Kernel 2
Kernel 3
Kernel 4
Write frame
Calibration
Input bitstream:
header data header data ...
frame
Scenario table / decision diagram

Prediction

A function that given RTS parameter values returns a scenario

- All parameter values known: Accurate prediction, but possibly **late:**
  - Adapt parameters and scenarios
  - Distributed predictor
  - Probabilistic prediction
- Miss prediction
- Backup scenario(s)
- Prediction overhead

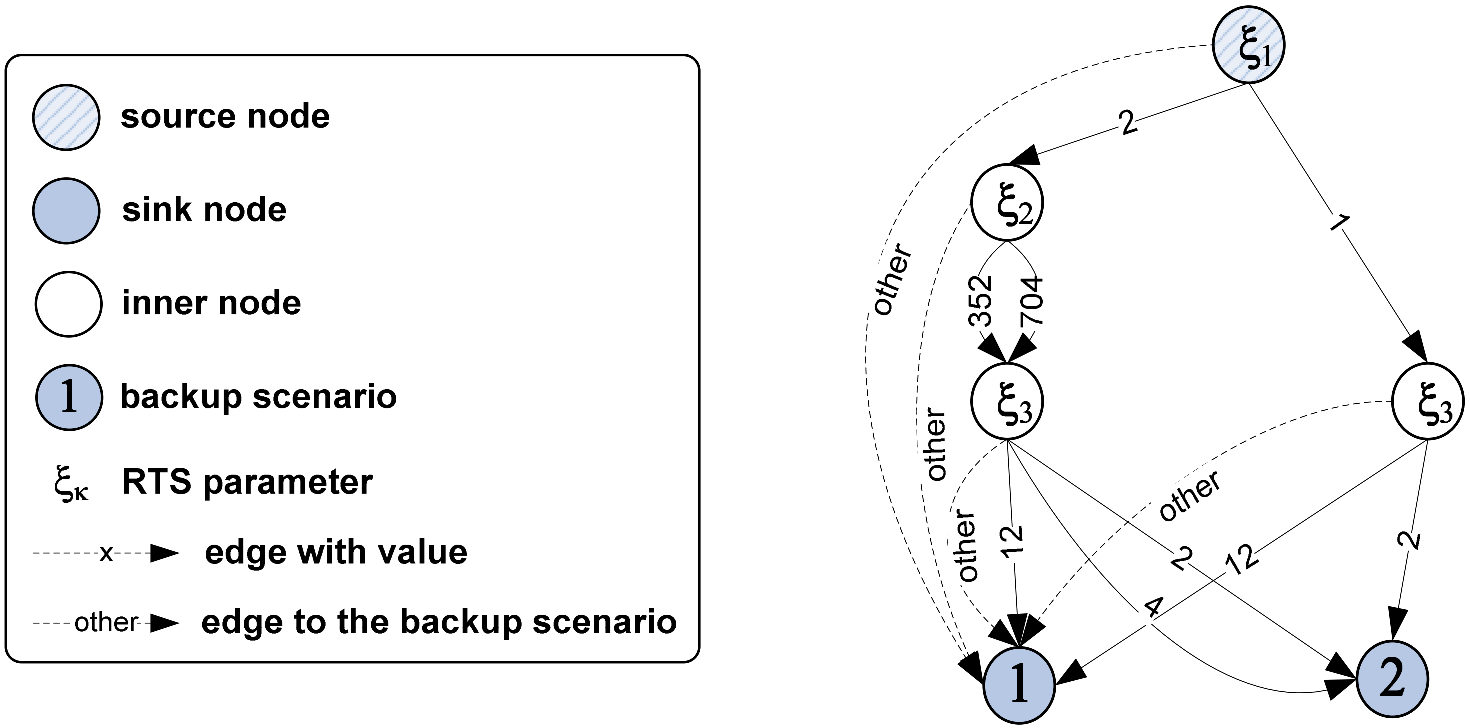
Centralized vs. Distributed Prediction



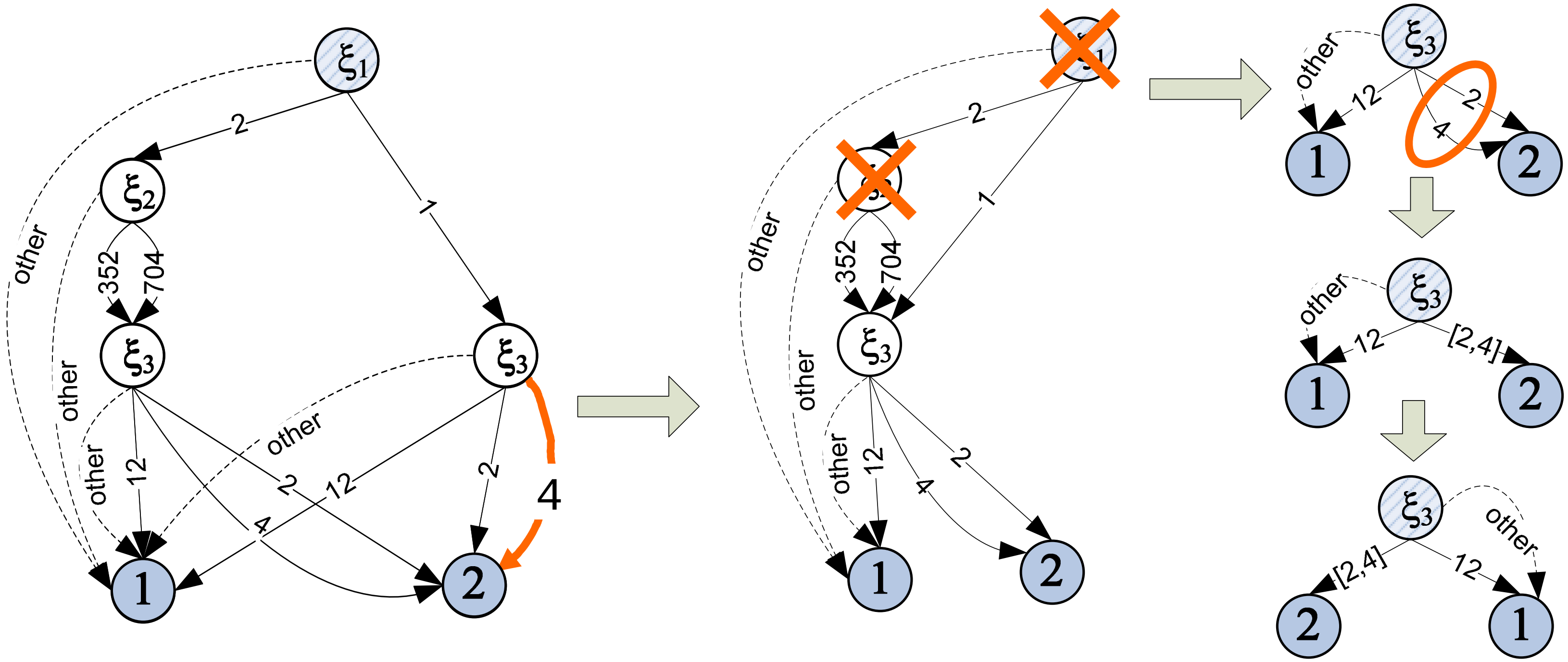
Prediction Function Implementation

Multi-valued multi-terminal decision diagram

- Code size & average decision time are important
- Trade-off with accuracy



Quality - Cost Trade Off

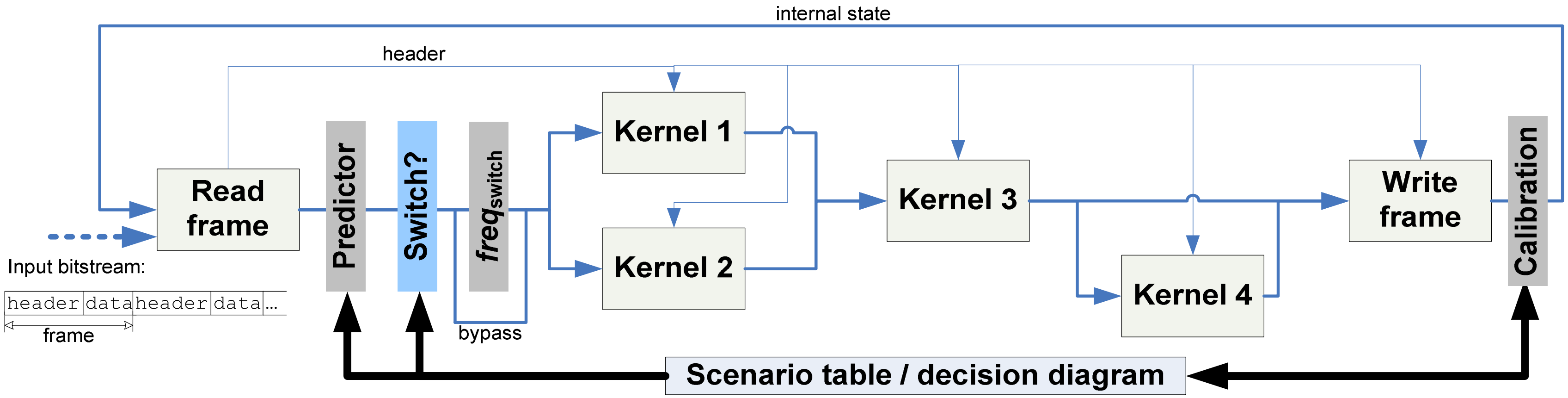


- Node merging
- Node removal

- Interval edges
- Edge reordering

Exploitation

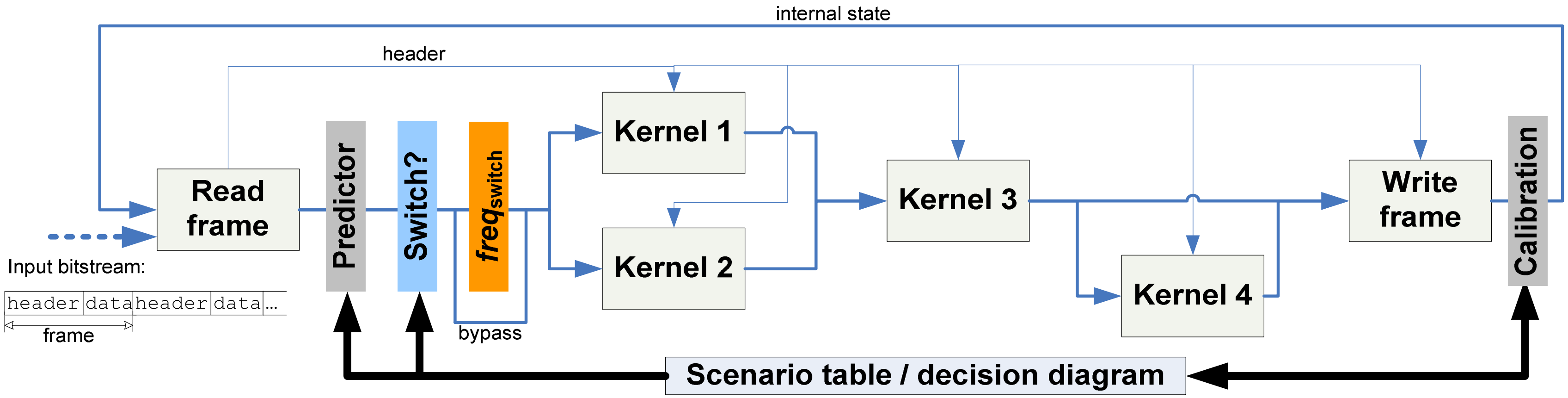
Very context dependent



Continue with

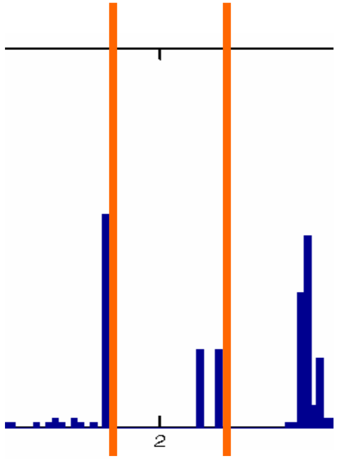
- a new frequency
- an optimized code
- …

Switching


internal state
header
Read frame
Predictor
Switch?
$freq_{switch}$
Kernel 1
Kernel 2
Kernel 3
Kernel 4
Write frame
Calibration
bypass
Input bitstream:
header data header data ...
frame
Scenario table / decision diagram

Switching Overhead

- Depends on the from-to scenarios



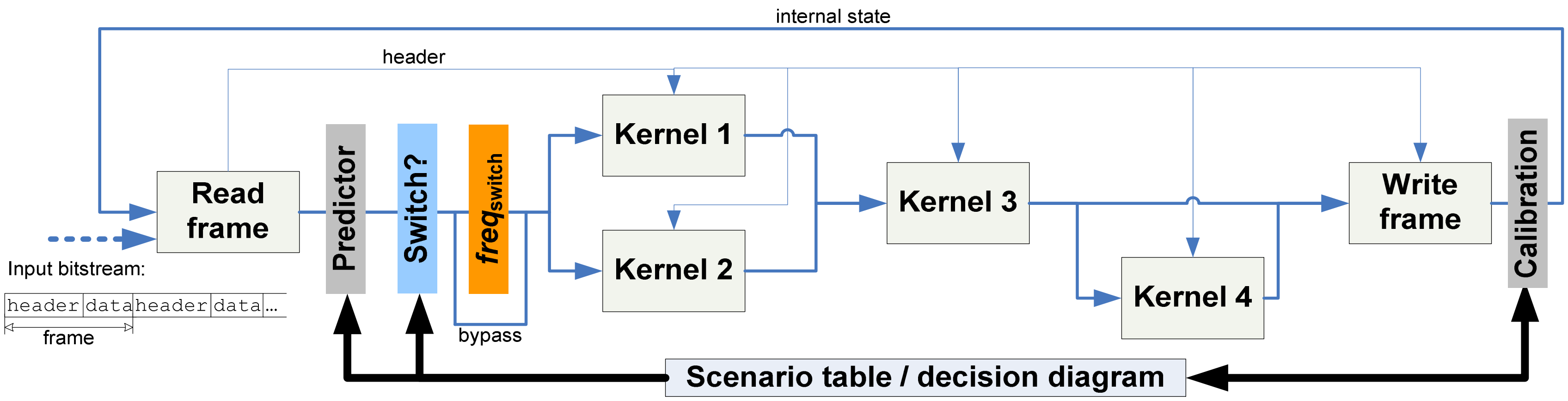
- Trade off with scenario selection
  - Switching overhead
  - Scenario cost overhead
- Trade off with exploitation overhead
  - e.g. scenarios with their own source code
  - Compress code of rarely occurring scenarios:
    - less storage
    - larger switching overhead

To Switch or Not to Switch ?

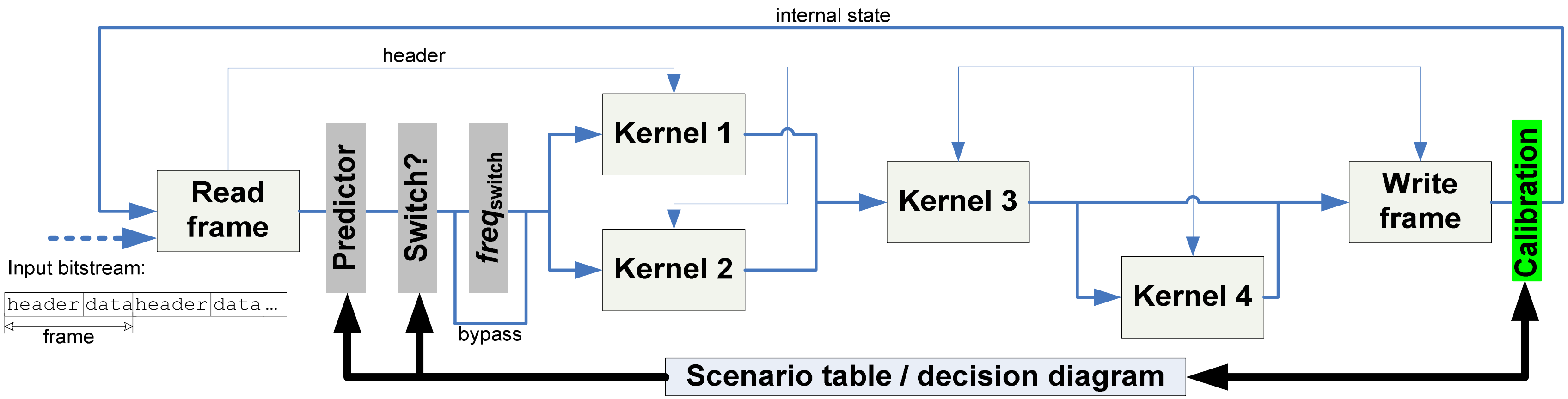
- Depends on the from-to scenarios
- Decision depends on
  - Switching cost
  - Potential cost reduction
  - Potential quality loss

200MHz → 195MHz

200MHz → 205MHz



Calibration

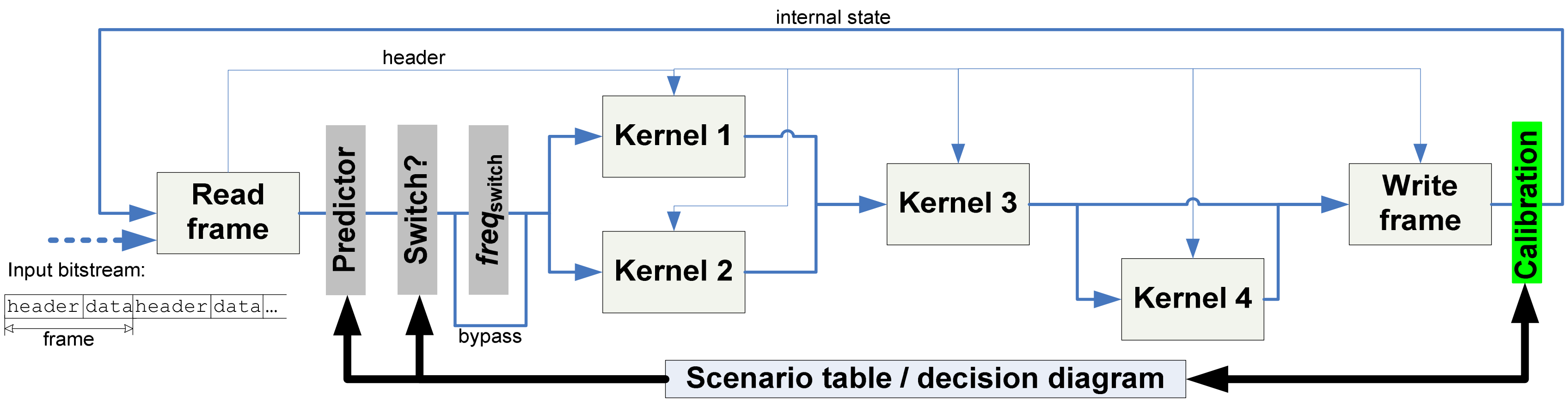


Why Calibrate ?

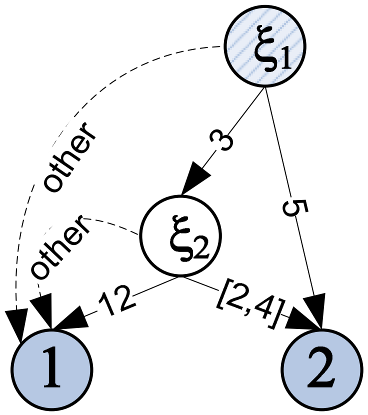
- Quality preservation
- Cost reduction

What can be Calibrated?

- Scenarios and their costs
- Predictor: decision diagram
- Switching mechanism



Decision Diagram



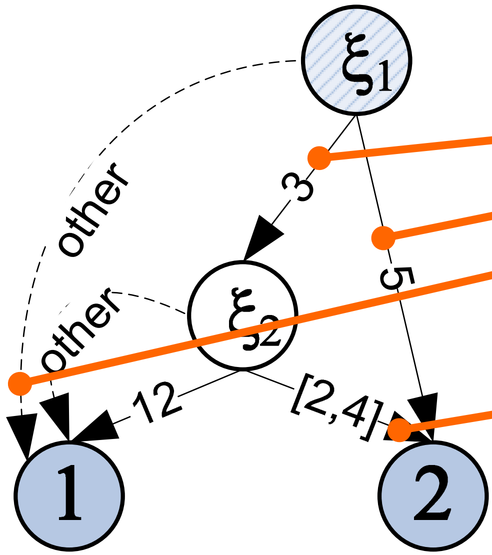
- Implemented as a program

Op	Variable id	Value	Data
JEQ	<var>	<val>	<address>
JL	<var>	<val>	<address>
JMP			<address>
SEQ	<var>	<val>	<scenario>
SLE	<var>	<val>	<scenario>
SBK			<scenario>

- Executed by an execution engine (10 line C program)

Decision Diagram

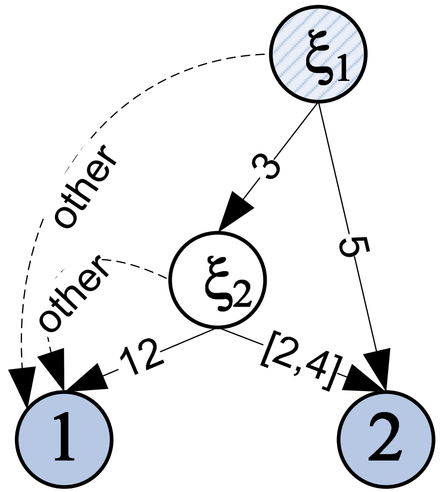
- Edge implementations



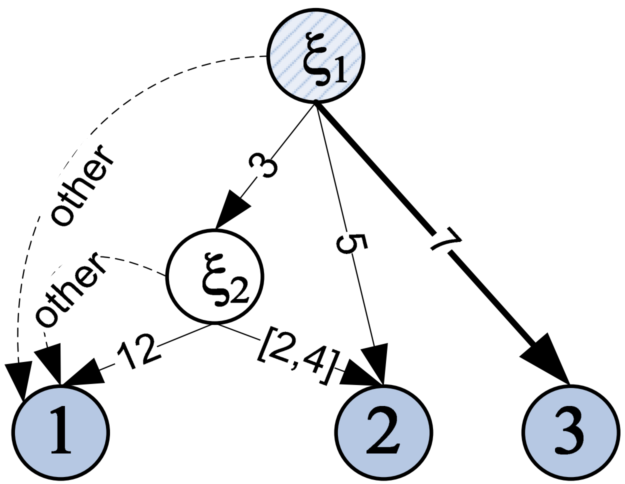
1: JEQ 1, 3, 4
2: SEQ 1, 5, 2
3: SBK 1
4: SEQ 2, 12, 1
5: JL 2, 2, 7
6: SLE 2, 4, 2
7: SBK 1

- JMP is not used at design time

Adding a new Scenario



1: JEQ 1, 3, 4
2: SEQ 1, 5, 2
3: SBK 1
4: SEQ 2, 12, 1
5: JL 2, 2, 7
6: SLE 2, 4, 2
7: SBK 1

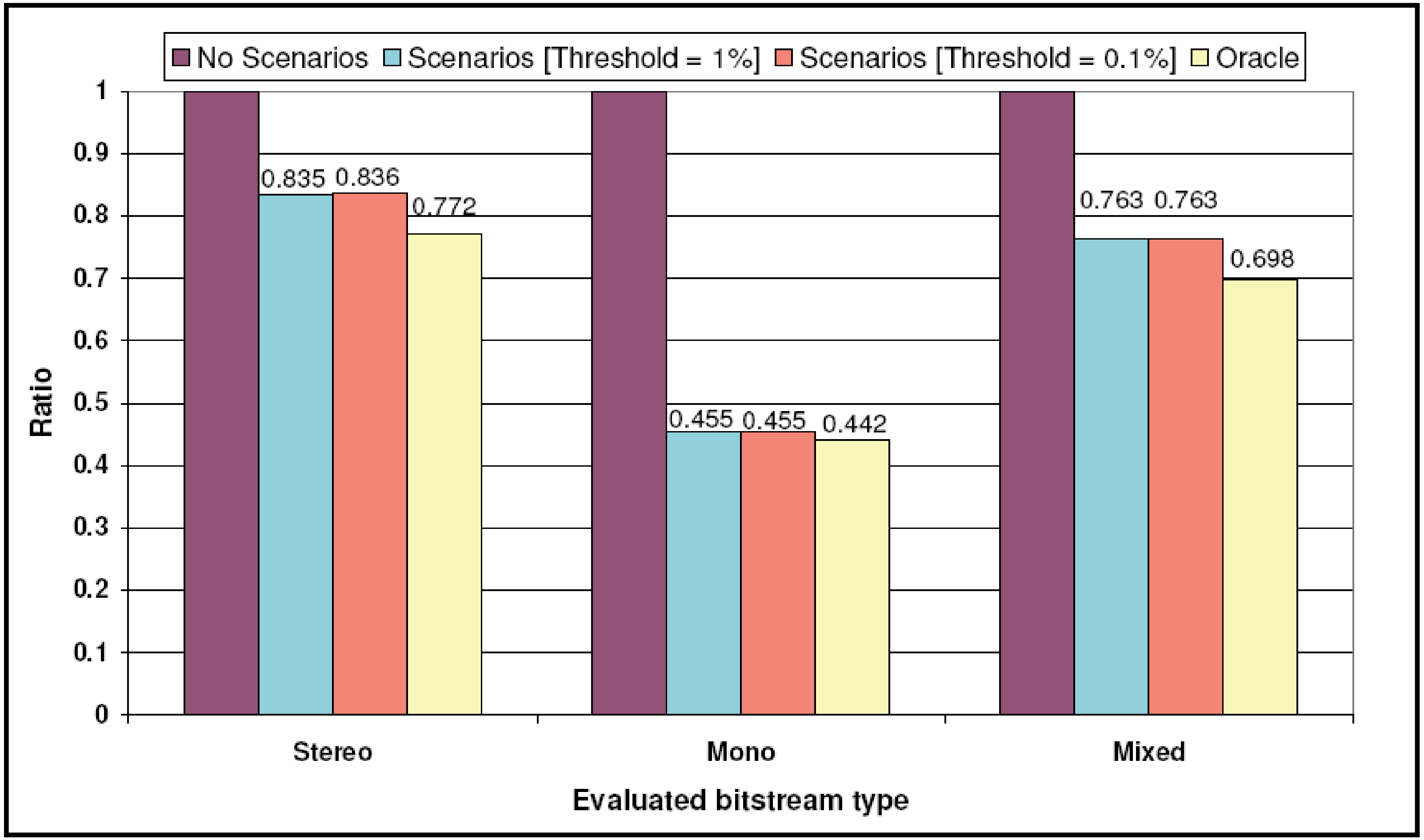


1: JEQ 1, 3, 4
2: SEQ 1, 5, 2
3: **JMP** 8
4: SEQ 2, 12, 1
5: JL 2, 2, 7
6: SLE 2, 4, 2
7: SBK 1
8: **SEQ** 1, 7, 3
9: **SBK** 1

Dynamic Voltage & Frequency Scaling

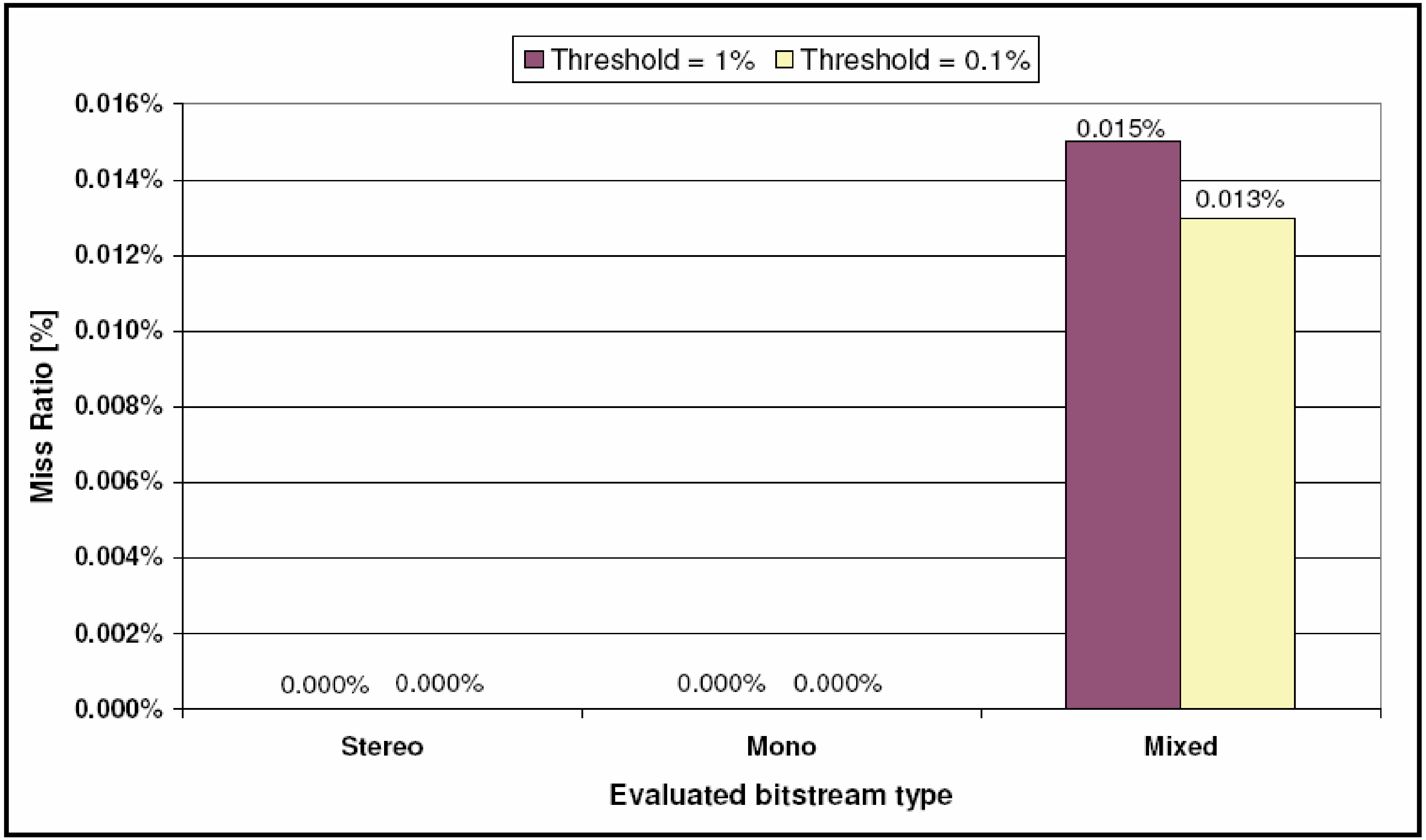
- for a dynamic streaming application
- on a single processor

MP3 Results



- No calibration: 24% → 16%

MP3 Results

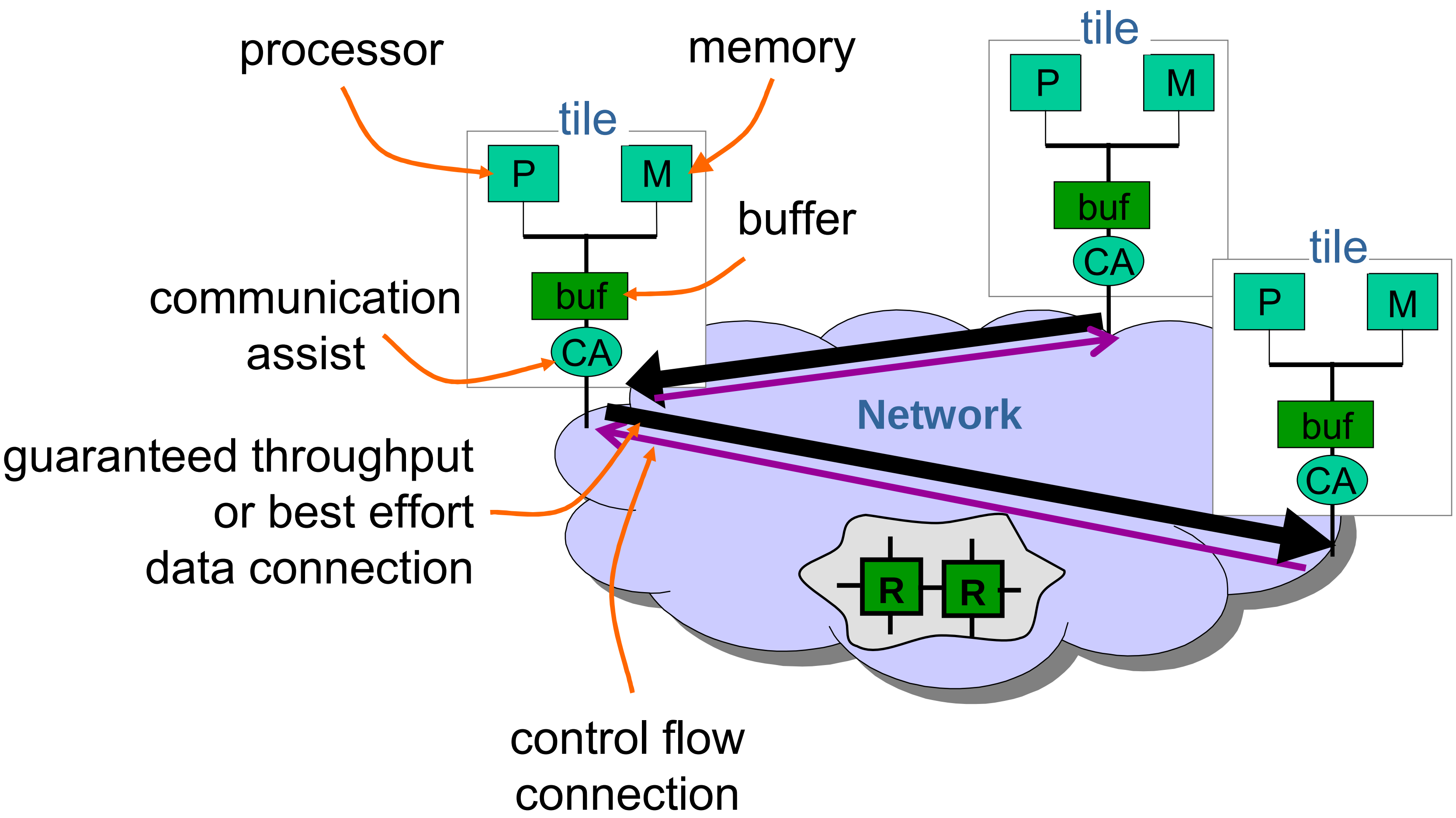


- No calibration: 0.013% → 0.1%

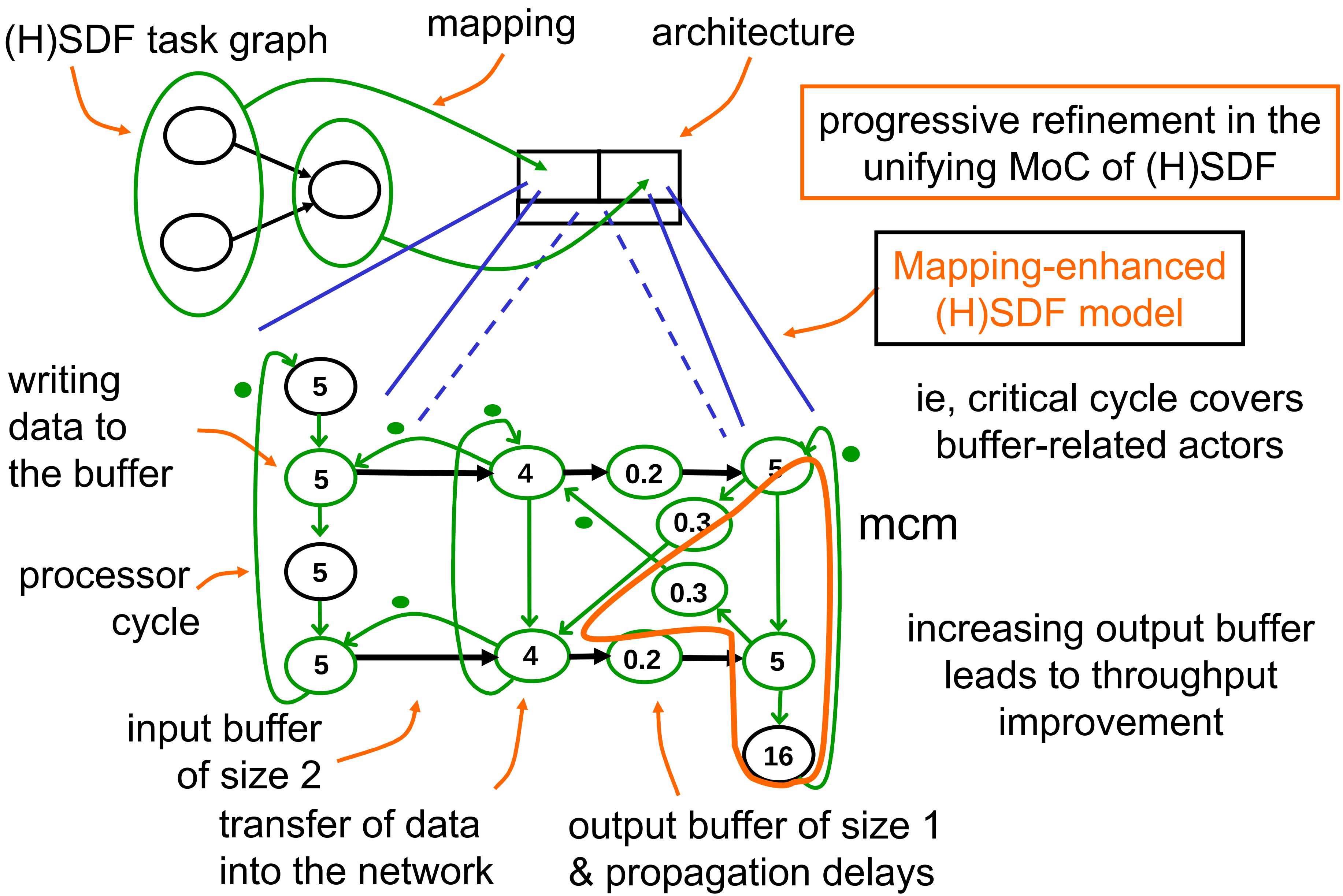
Run-time execution time prediction

- of a dynamic streaming application
- on a NoC-based MPSoC
- for QoS management

A NoC-based SoC Platform

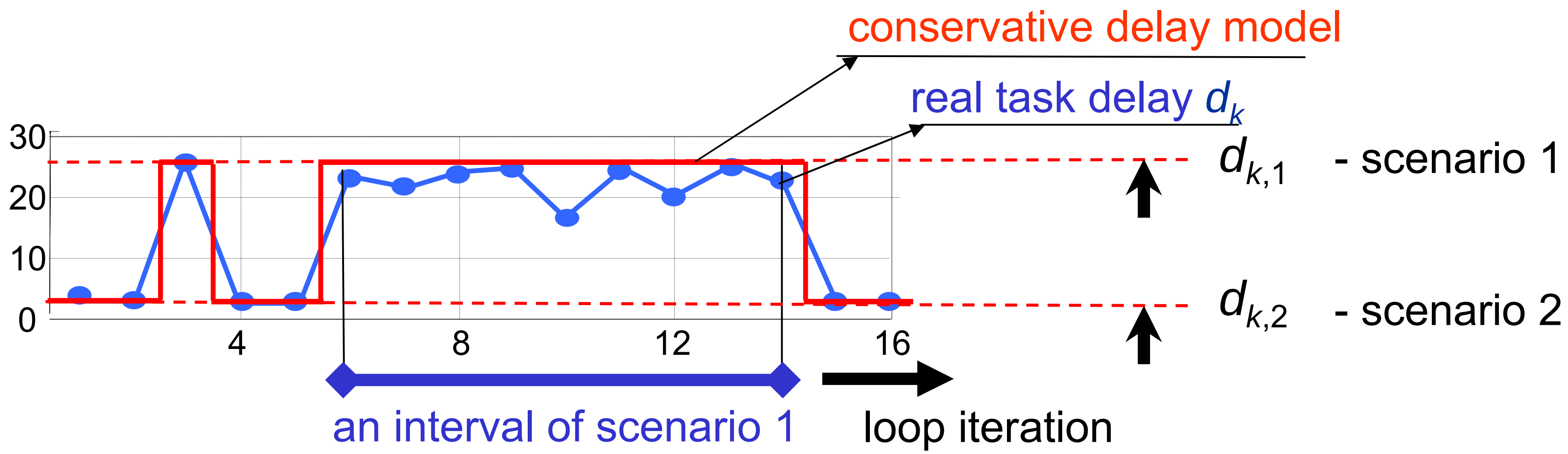


Philosophy

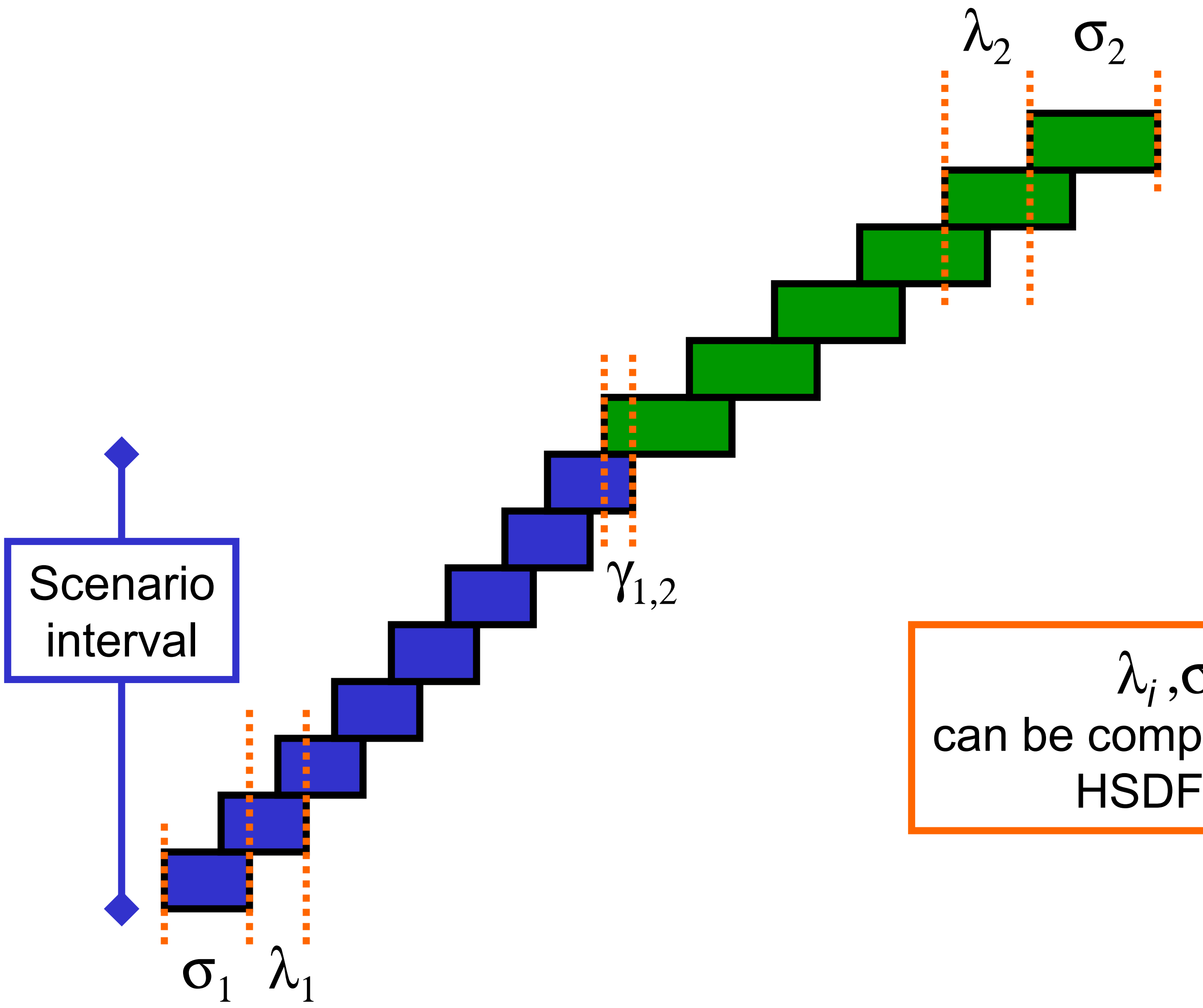


(H)SDF task graph
mapping
architecture
progressive refinement in the unifying MoC of (H)SDF
Mapping-enhanced (H)SDF model
writing data to the buffer
processor cycle
input buffer of size 2
transfer of data into the network
output buffer of size 1 & propagation delays
ie, critical cycle covers buffer-related actors
mcm
increasing output buffer leads to throughput improvement
5
5
5
5
4
4
0.2
0.2
5
5
0.3
0.3
16

Data-dependent Task Execution Times and Scenarios



The Essence: Scenario Intervals and Transitions



$\lambda_i, \sigma_i, \gamma_{i,j}$
can be computed from the HSDF graph

MPEG-4 Case Study

frame header

$\hat{\xi}_{i,s}$ →

$J_s, L_s, K_{s,t}$ →

run-time estimation

$\hat{d}_{k,s}$

↓

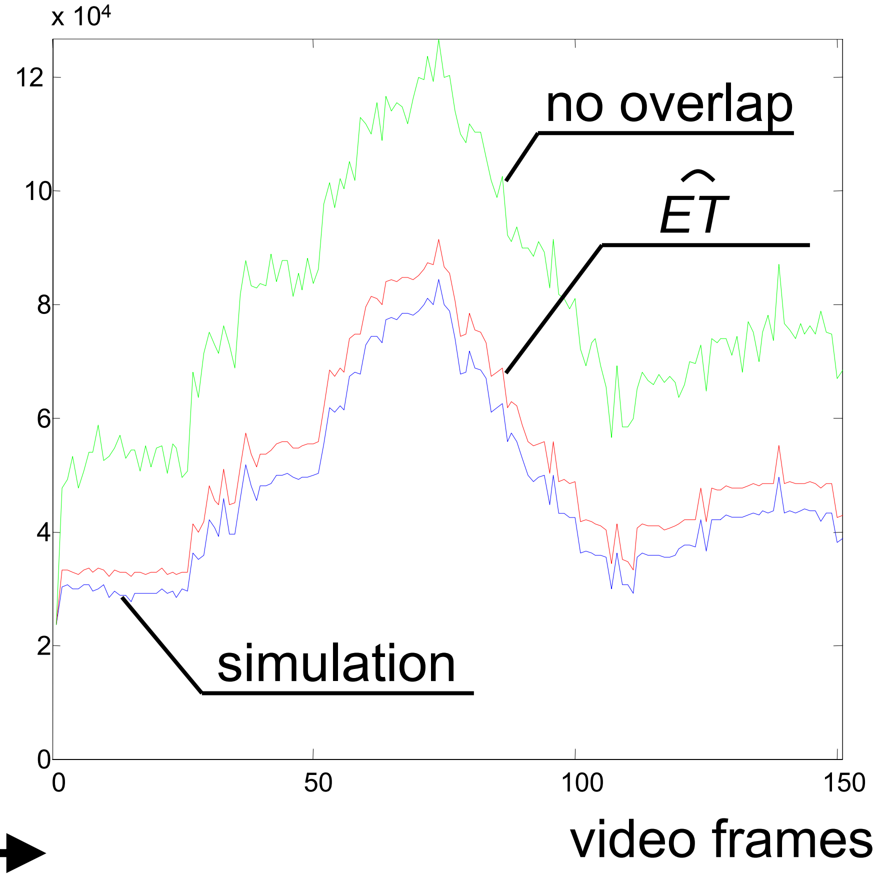
$\lambda_s, \sigma_s, \gamma_{s,t}$

↓

$\widehat{ET}$ →

results

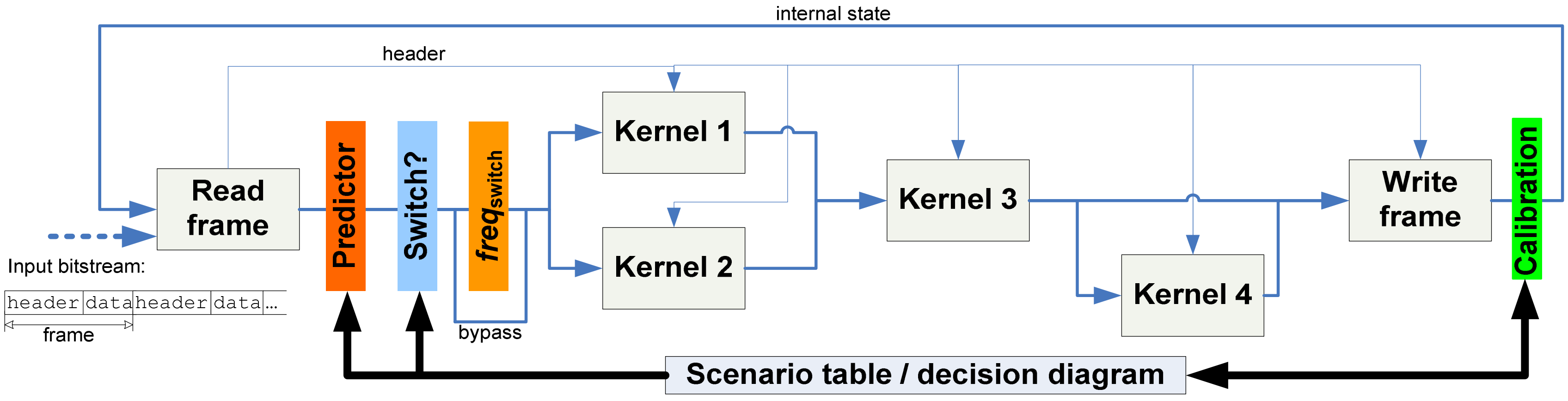
loop execution times (ARM7 kcycles)



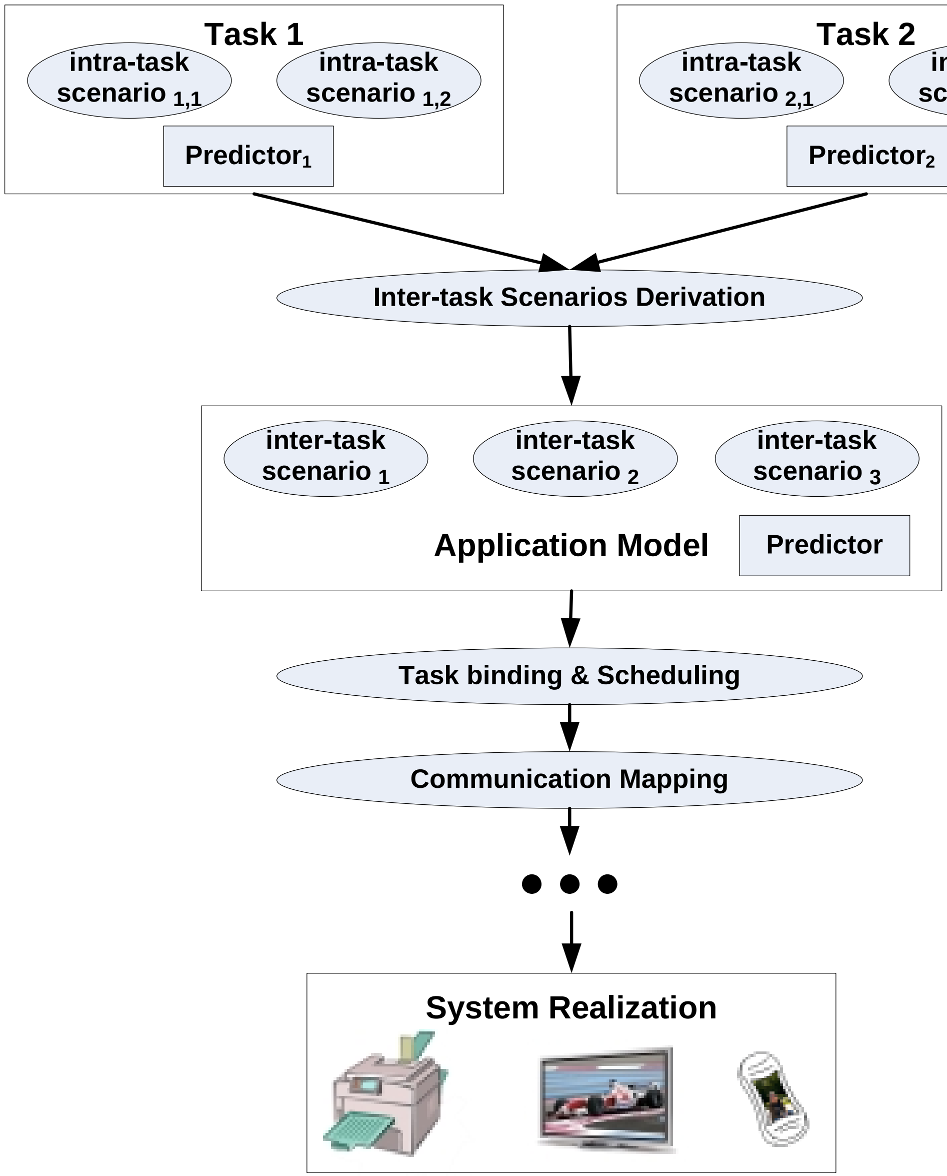
accuracy within 11%

Conclusions

System scenarios

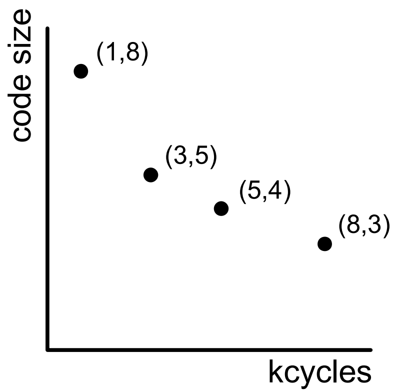


What Next ?



Beyond single-processor single-task systems

… for multiple cost dimensions



+ data memory size
+ memory accesses
+ throughput
+ bandwidth
+ area
+ energy
+ ...

Thank you !

Questions ?

More info:

www.es.ele.tue.nl/~tbasten/
www.es.ele.tue.nl/scenarios/

research = understanding

invention = problem solving

Rutger van Santen, former Rector Magnificus TU/e
In a discussion on the value of patents for the TU/e
Cursor 46(27)