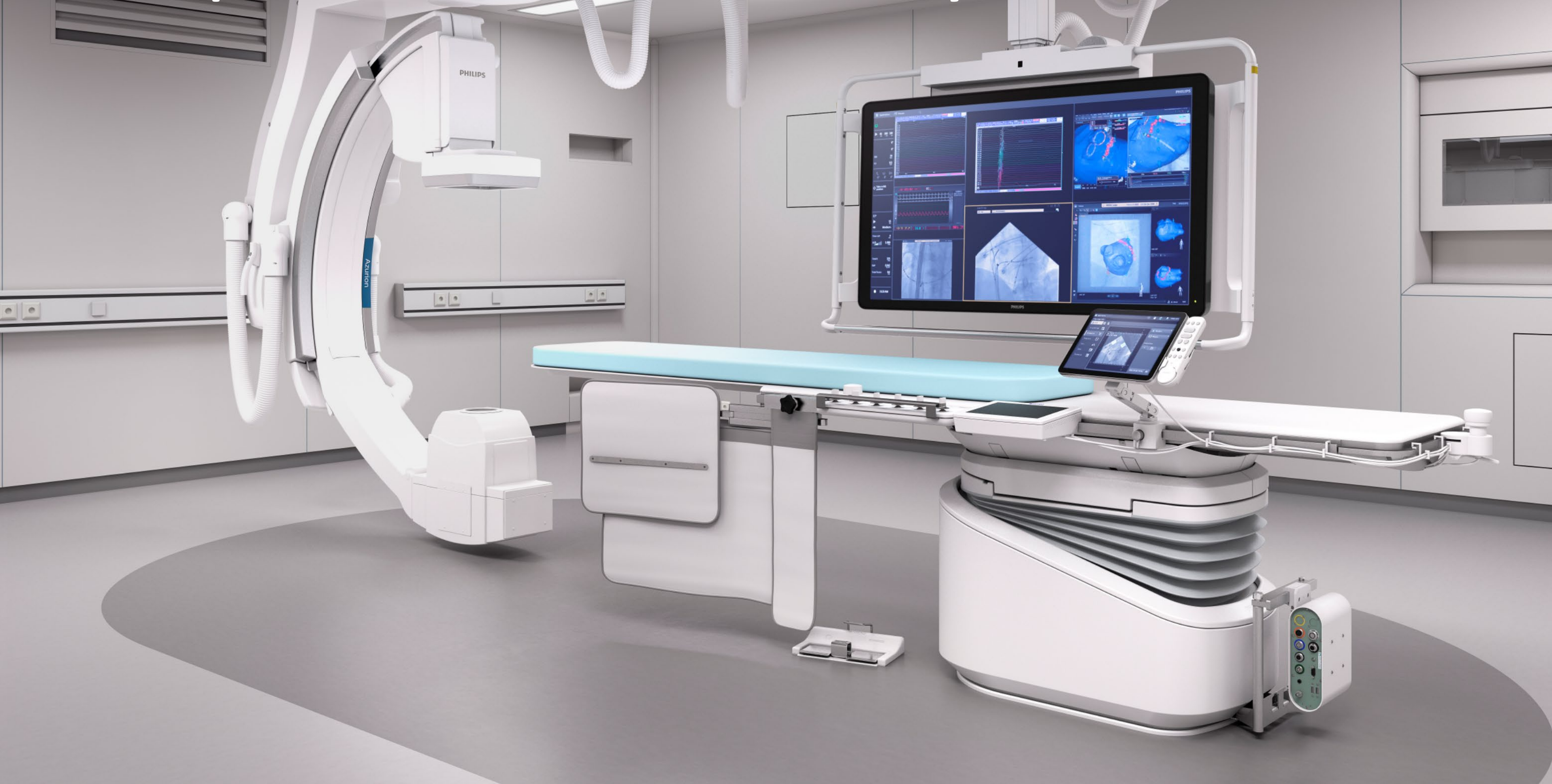
Philips Azurion Interventional X-Ray Platform



Philips Azurion Interventional X-Ray Platform

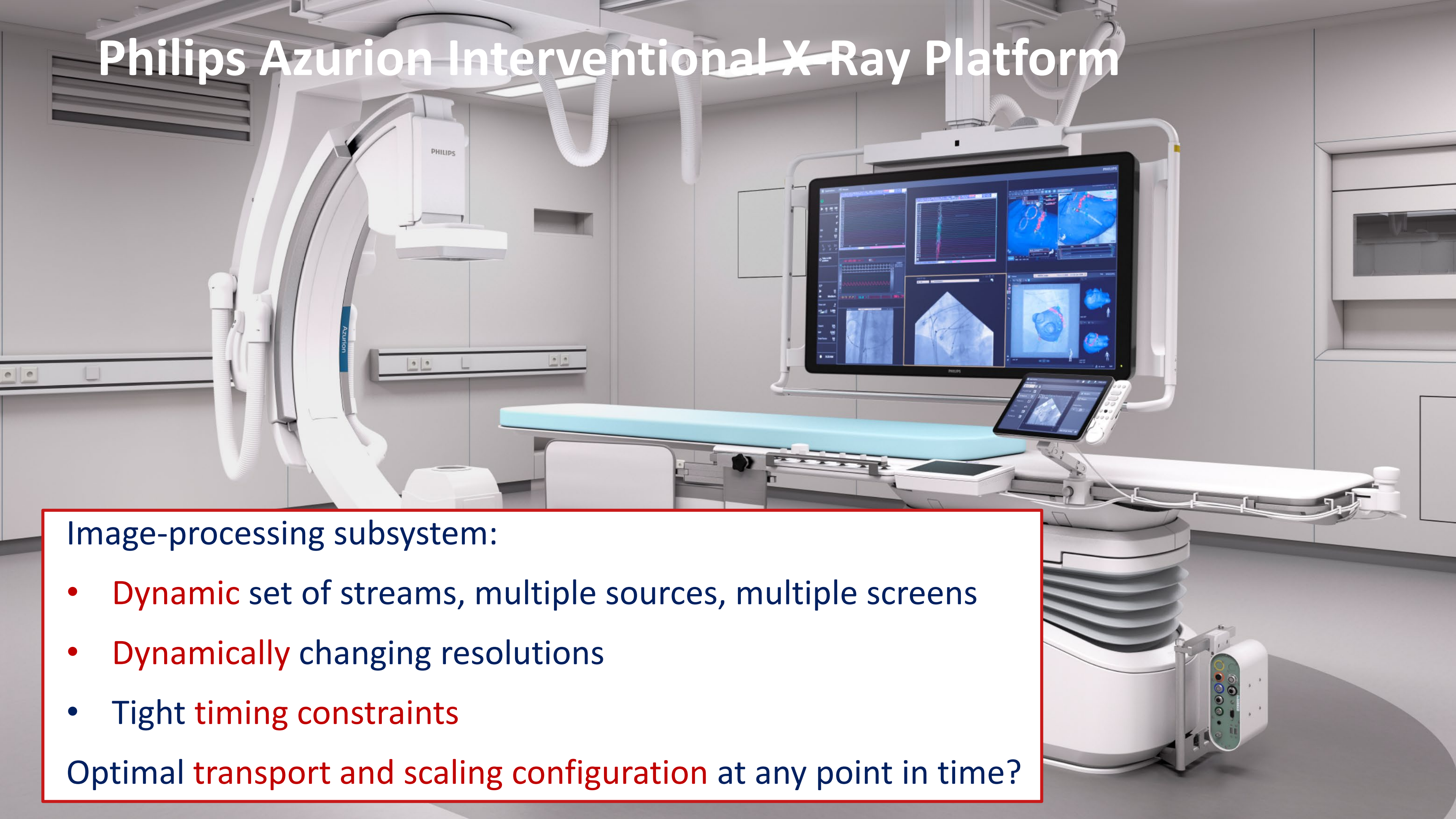
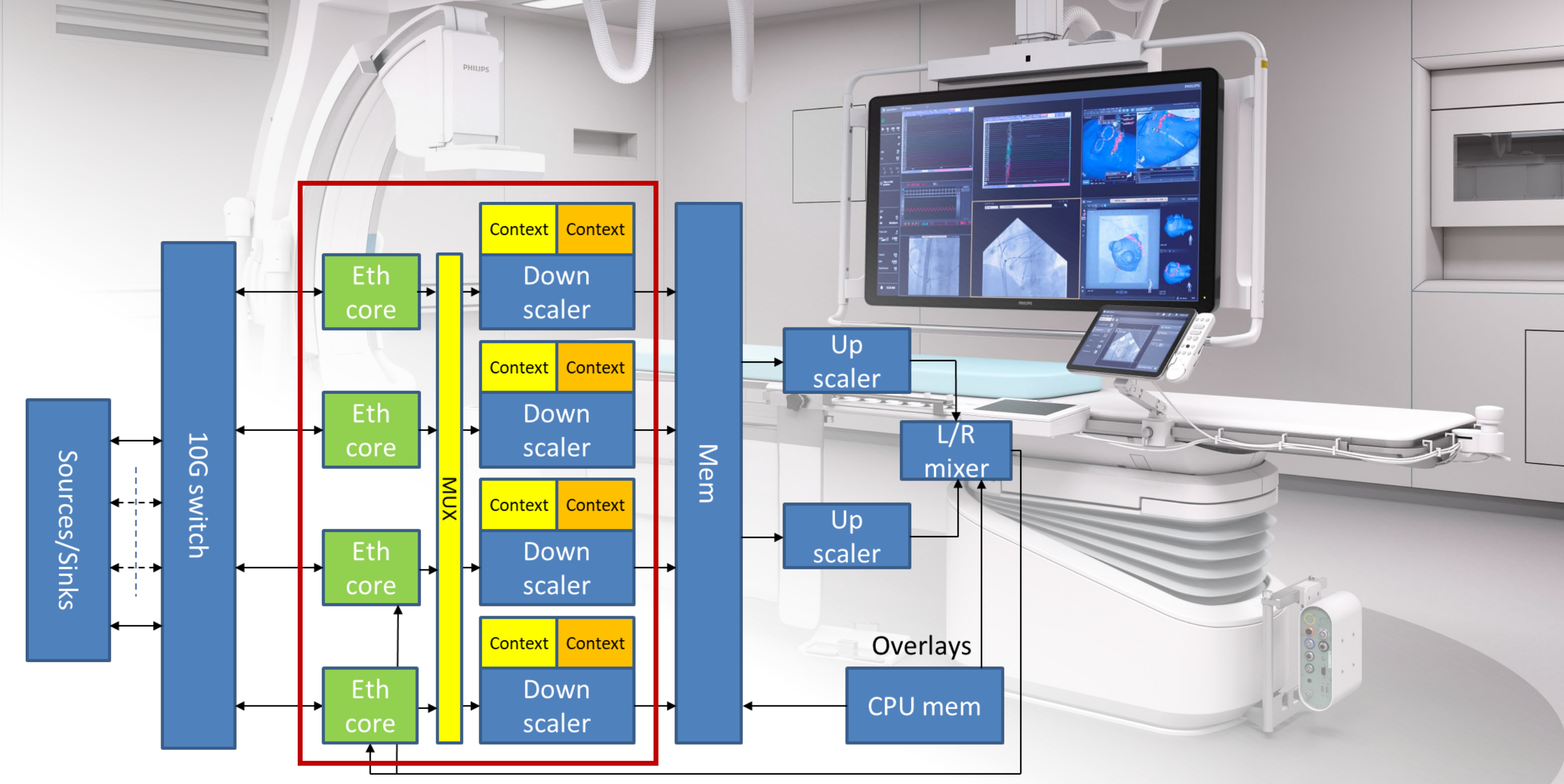


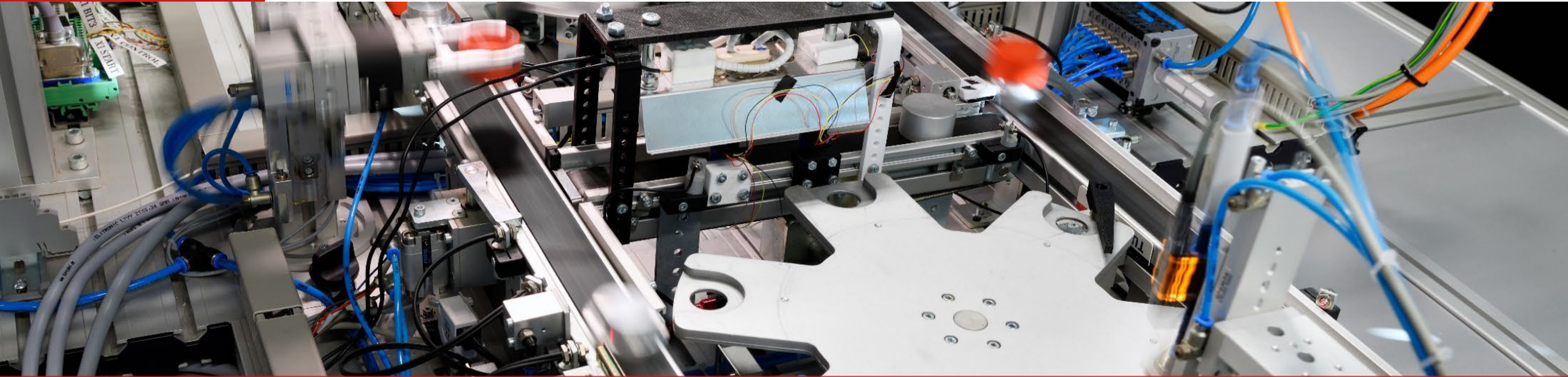
Image-processing subsystem:

- Dynamic set of streams, multiple sources, multiple screens
- Dynamically changing resolutions
- Tight timing constraints

Optimal transport and scaling configuration at any point in time?

Philips Azurion Interventional X-Ray Platform





Model-driven Quality and Resource Management for CPS

Twan Basten, Marc Geilen

in collaboration with Martijn Hendriks, Kees Goossens, and others from the FitOptiVis & TRANSACT teams

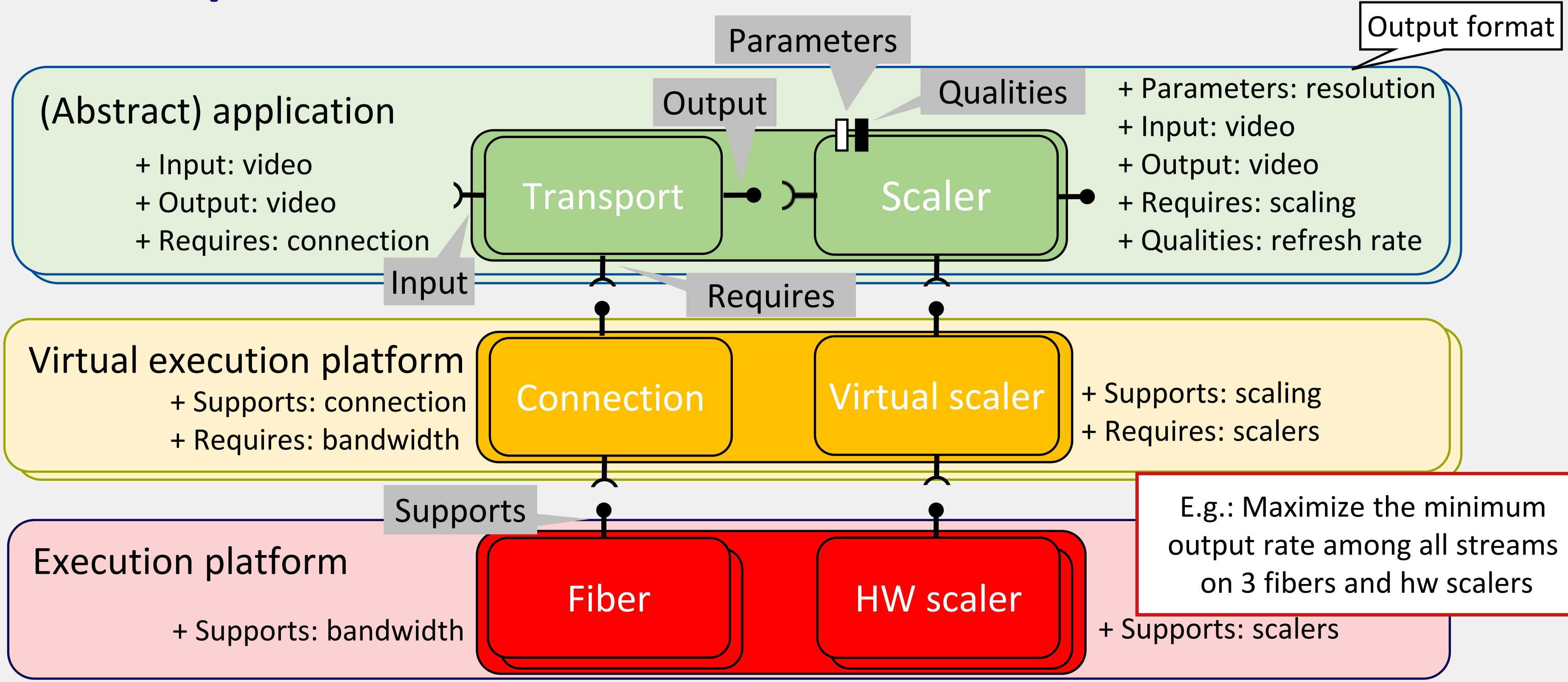
TRANSACT: ECSEL JU grant number 101007260

FitOptiVis: ECSEL JU grant number 783162

Overview

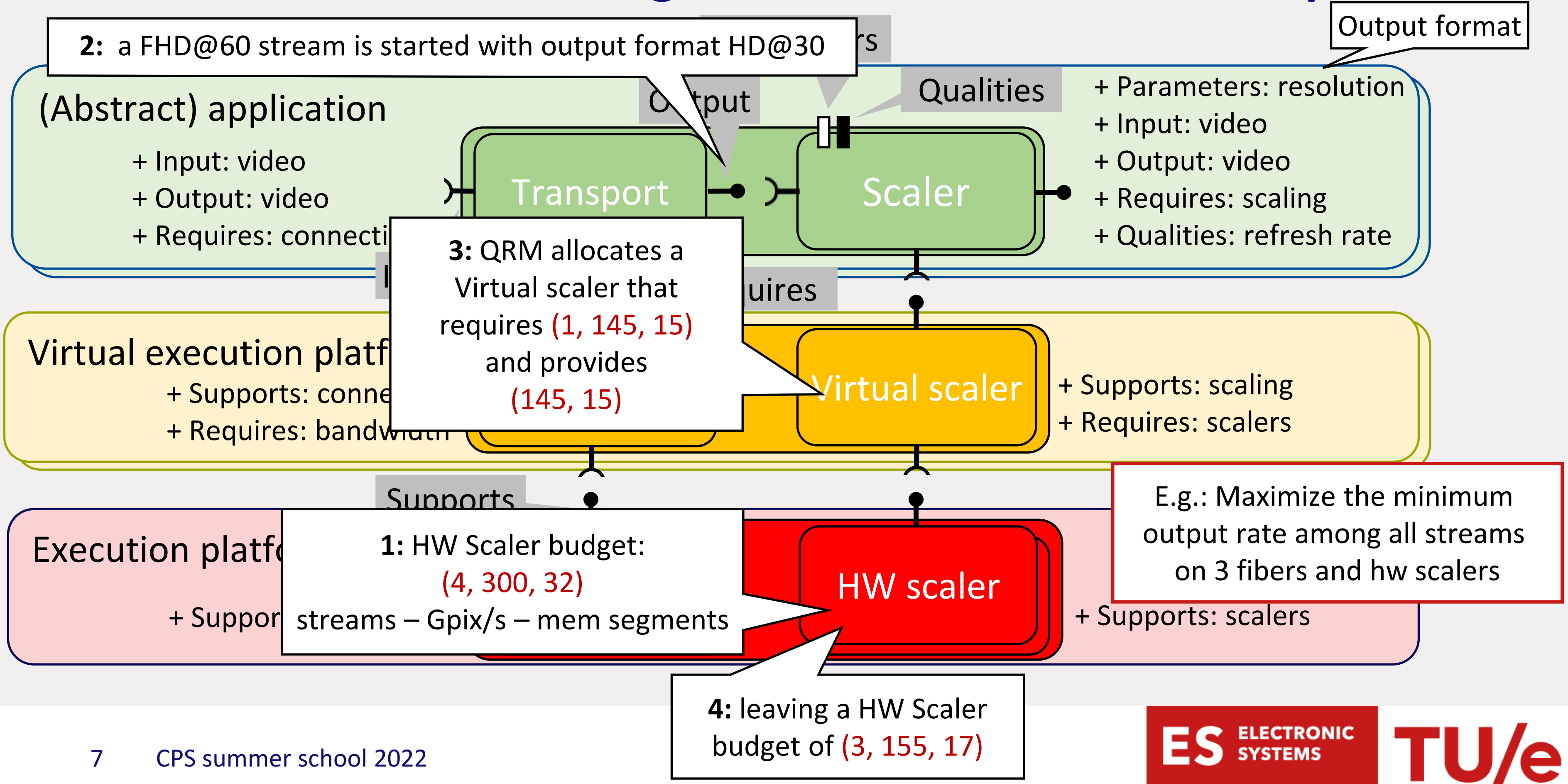
- Quality & Resource Management (QRM)
- Pareto analysis
- Component Interface Model for QRM
- Quality and Resource Management Language and Tools

Component Interface Model for QRM – Driver Case

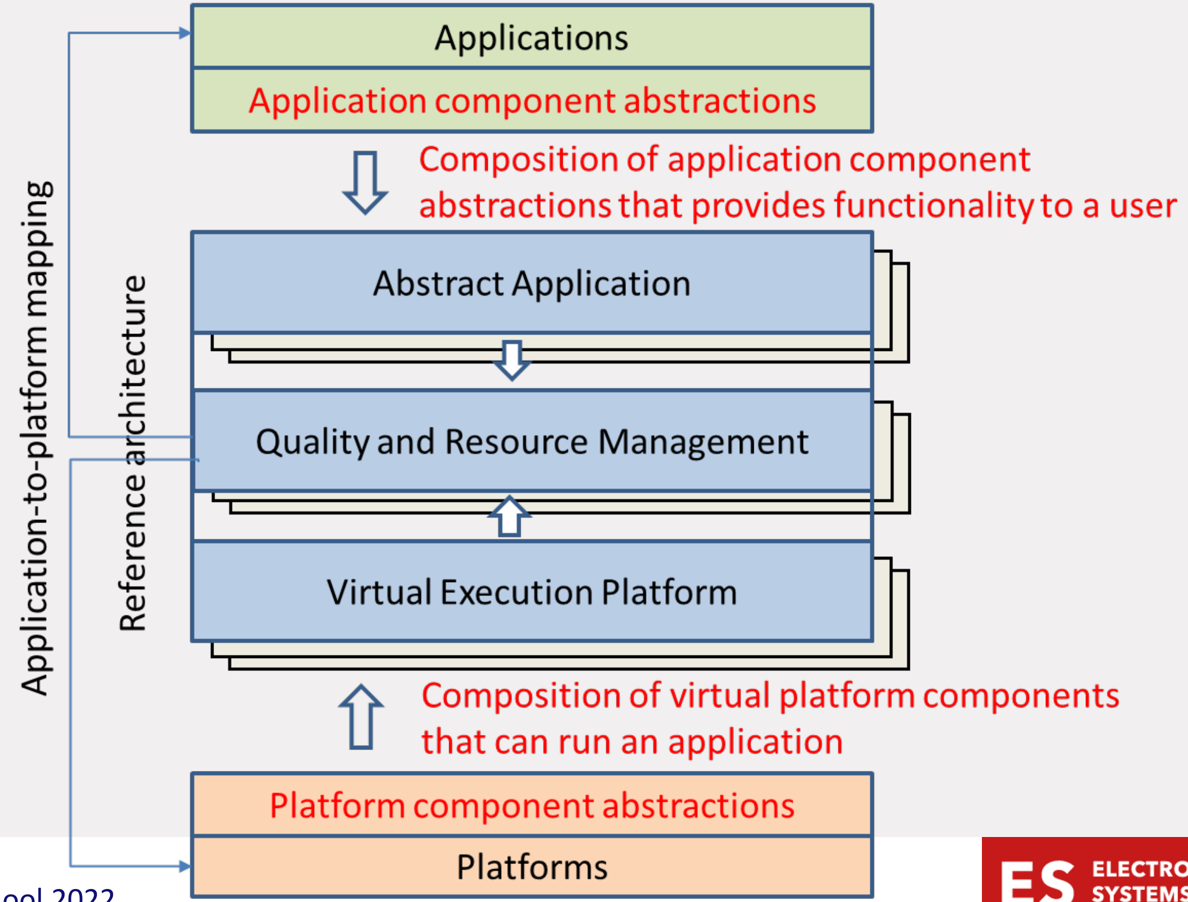


E.g.: Maximize the minimum output rate among all streams on 3 fibers and hw scalers

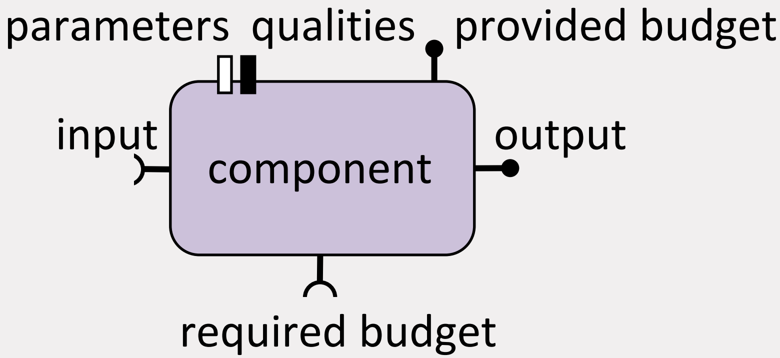
Multi-Source Streaming Driver Case – QRM Example



QRM Architecture

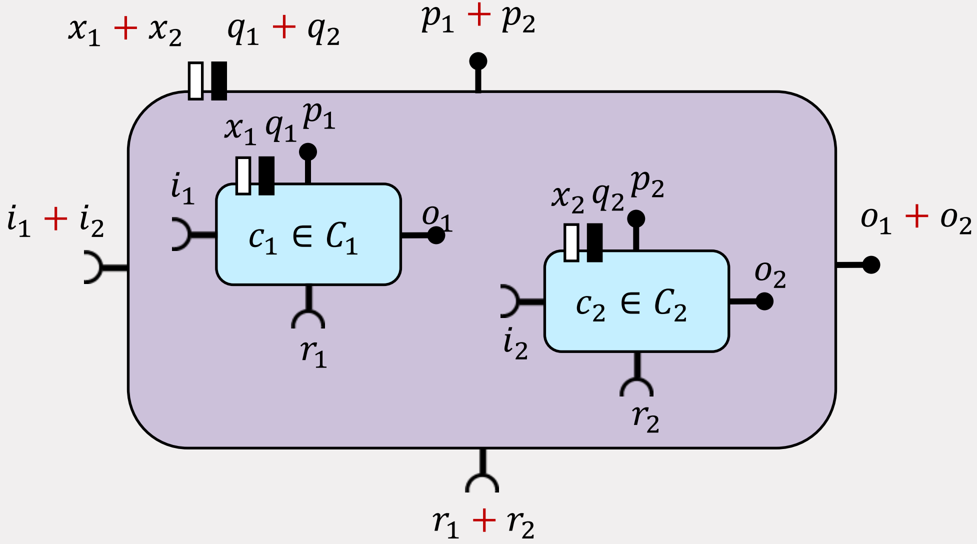


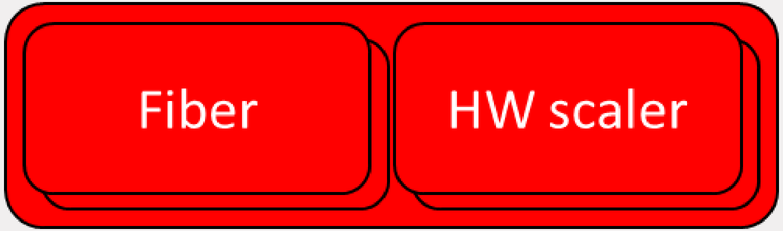
Component QRM Interface Model



component = set of configurations, 6-tuples

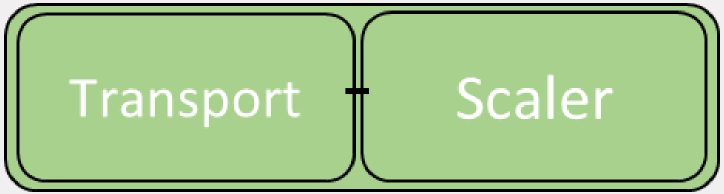
Free Composition

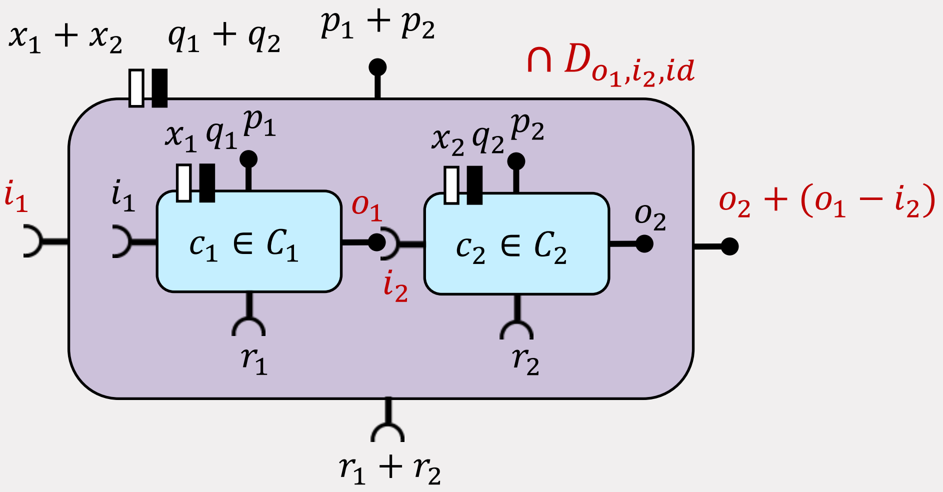


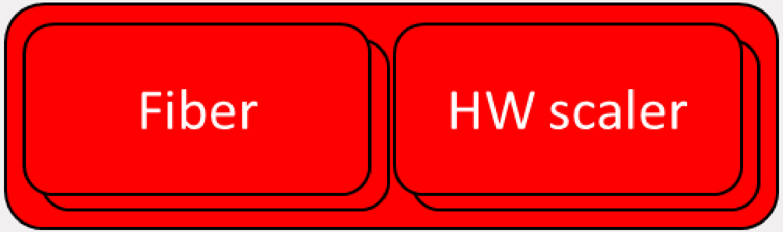


Wrap components, adding parameters, qualities, budgets, inputs, outputs

Horizontal Composition



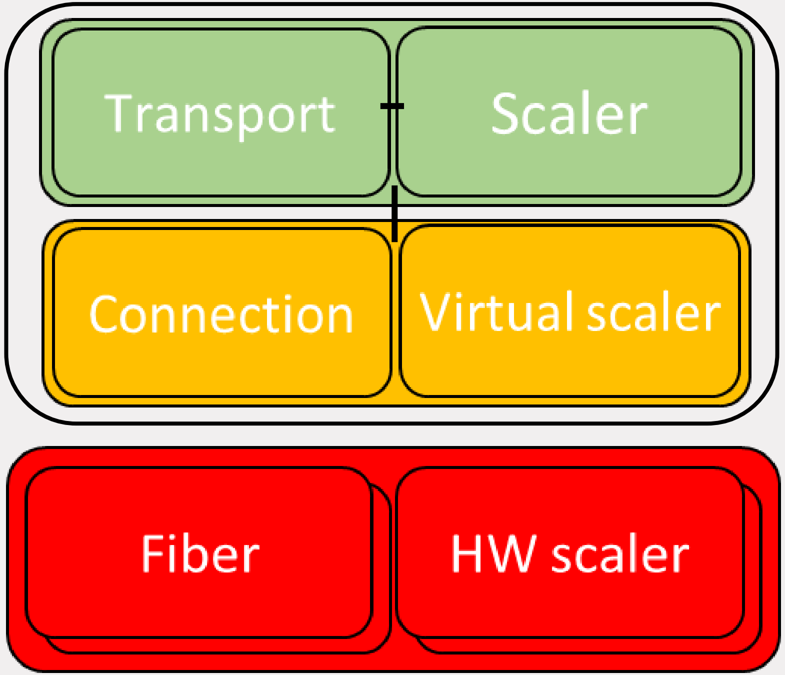


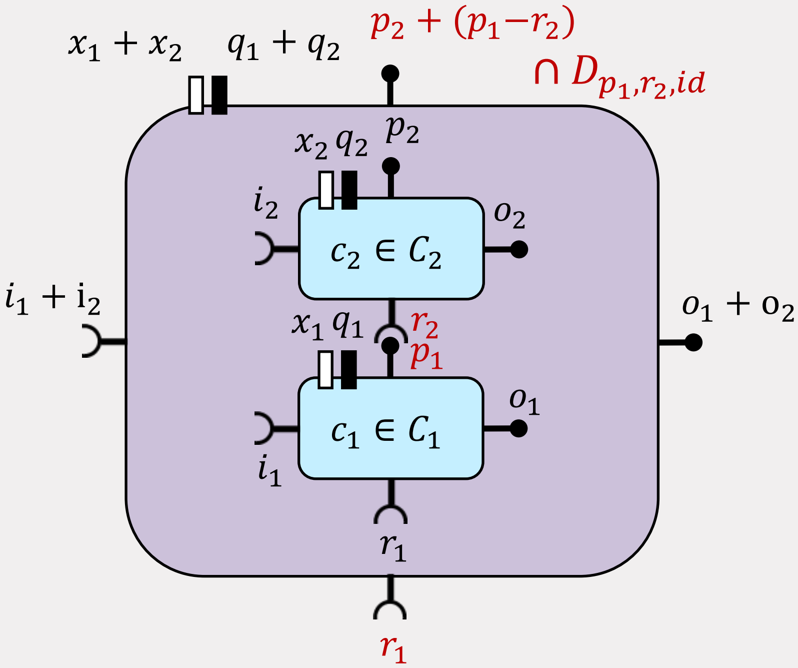


Connect output to input

- matching inputs and outputs ($\leq$)
- adding parameters, qualities, budgets
- exporting remaining inputs, outputs

Vertical Composition

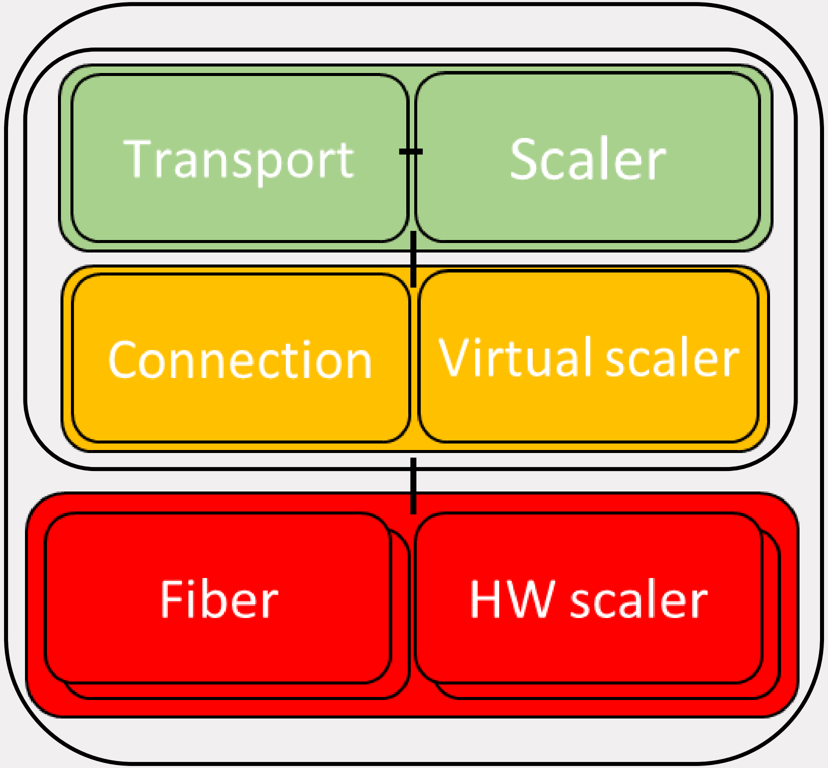


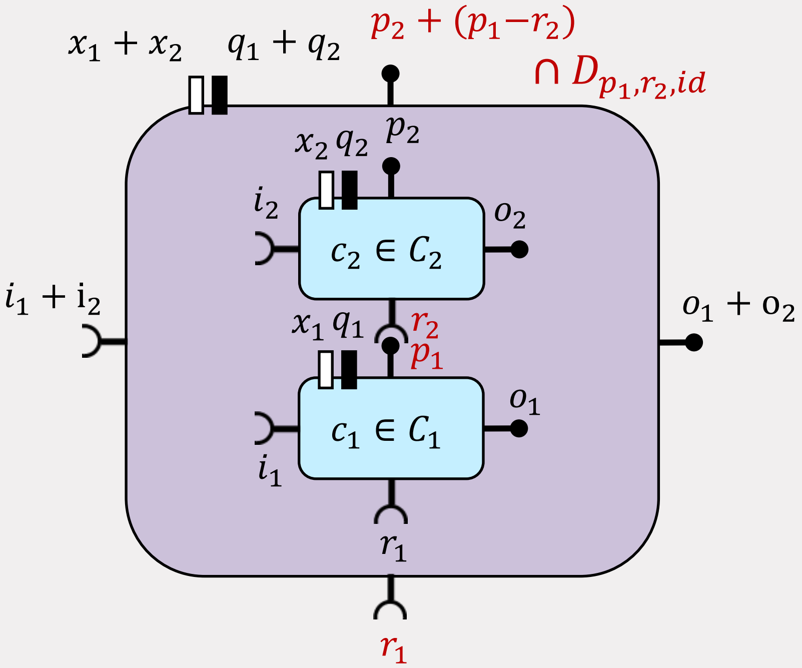


Connect provided and required budgets

- matching budgets ($\leq$)
- adding parameters, qualities, inputs, outputs
- exporting remaining budgets

Vertical Composition



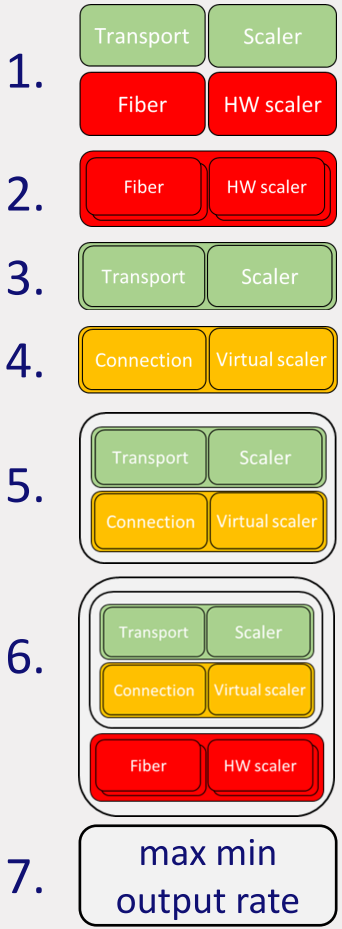


Connect provided and required budgets

- matching budgets ($\leq$)
- adding parameters, qualities, inputs, outputs
- exporting remaining budgets

A Quality & Resource Manager for the Driver Case

1. Input all application, platform component models
2. Compose platform (free composition)
3. Compose all streams (horizontal)
4. Create veps (free)
5. Deploy streams on veps (vertical)
6. For all bindings
   - Compose stream-vep components with platform component (vertical)
   - Keep Pareto-optimal configurations
7. Select optimum for some given cost function



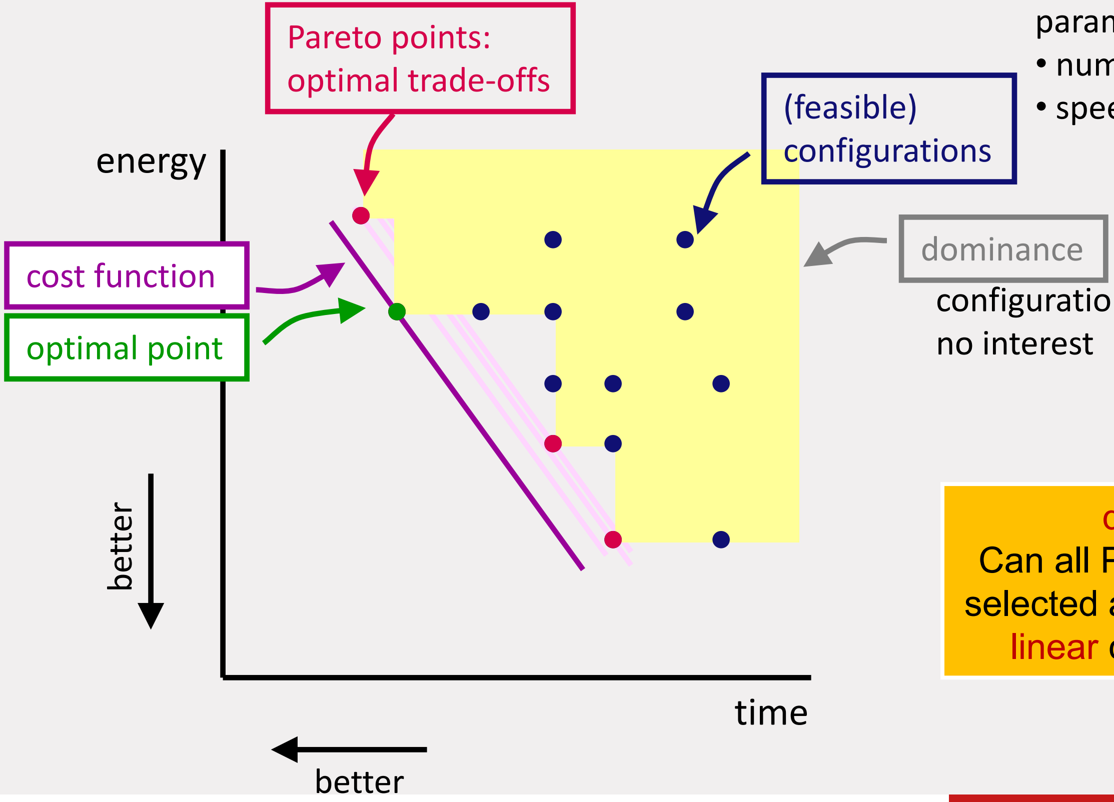
Overview

- Quality & Resource Management (QRM)
- Pareto analysis
- Component Interface Model for QRM
- Quality and Resource Management Language and Tools

ES ELECTRONIC SYSTEMS TU/e

Optimization - a Pareto space

multiple objectives,
e.g. **time** and **energy**



parameters:
- number of processors
- speed of processors

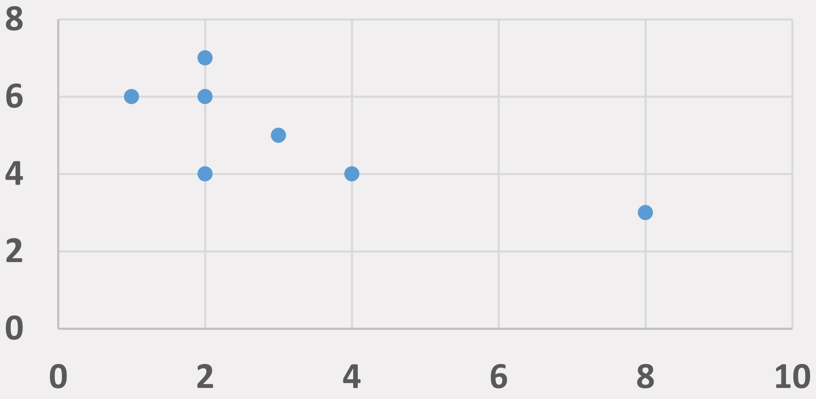
configurations of
no interest

question
Can all Pareto points be selected as optimal with a linear cost function?

Pareto points - Exercises

less = better

8	3
2	4
3	5
4	4
1	6
2	7
2	6

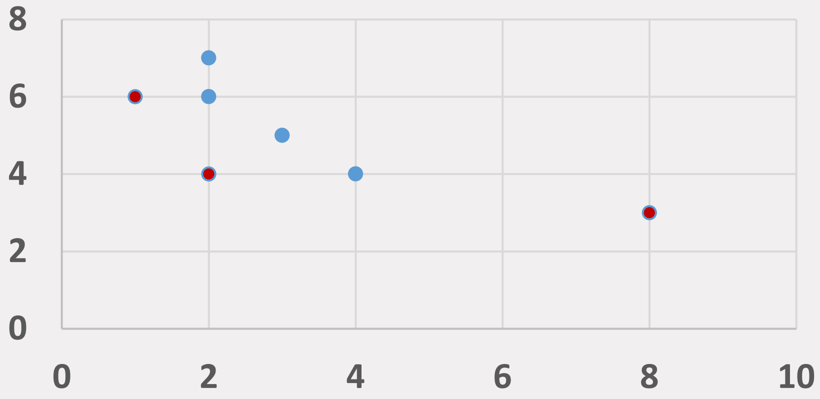


(How many) Pareto points ?

Pareto points - Exercises

less = better

8	3
2	4
3	5
4	4
1	6
2	7
2	6



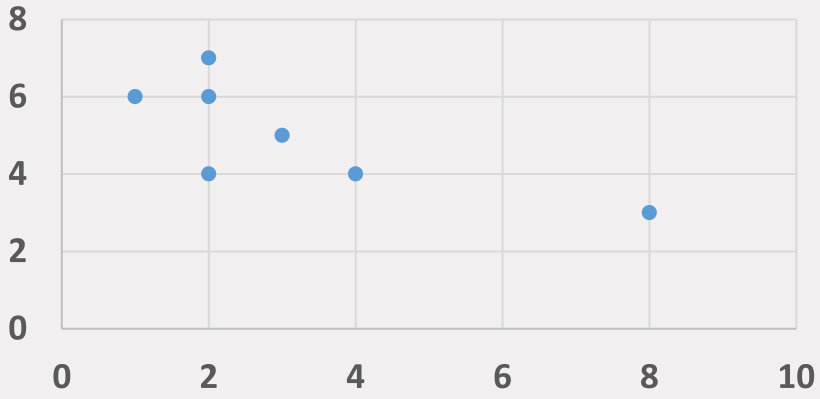
(How many) Pareto points ?

3: (1,6), (2,4), (8,3)

Pareto points - Exercises

greater = better

8	3
2	4
3	5
4	4
1	6
2	7
2	6

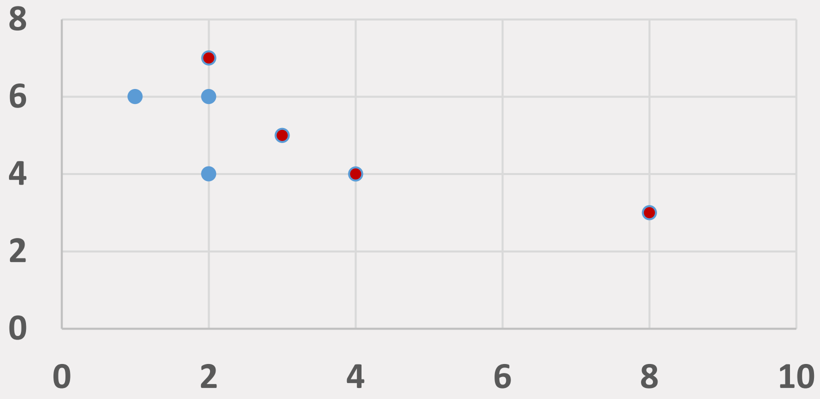


(How many) Pareto points ?

Pareto points - Exercises

greater = better

8	3
2	4
3	5
4	4
1	6
2	7
2	6



(How many) Pareto points ?

4: (8,3), (4,4), (3,5), (2,7)

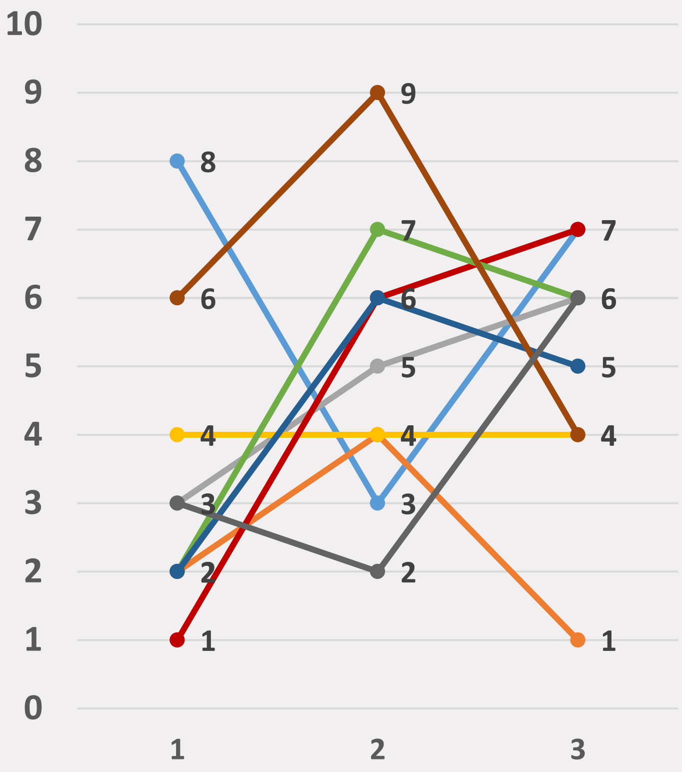
Pareto points - Exercises

less = better

8	3	7
2	4	1
3	5	6
4	4	4
1	6	7
2	7	6
2	6	5
6	9	4
3	2	6

(How many) Pareto points ?

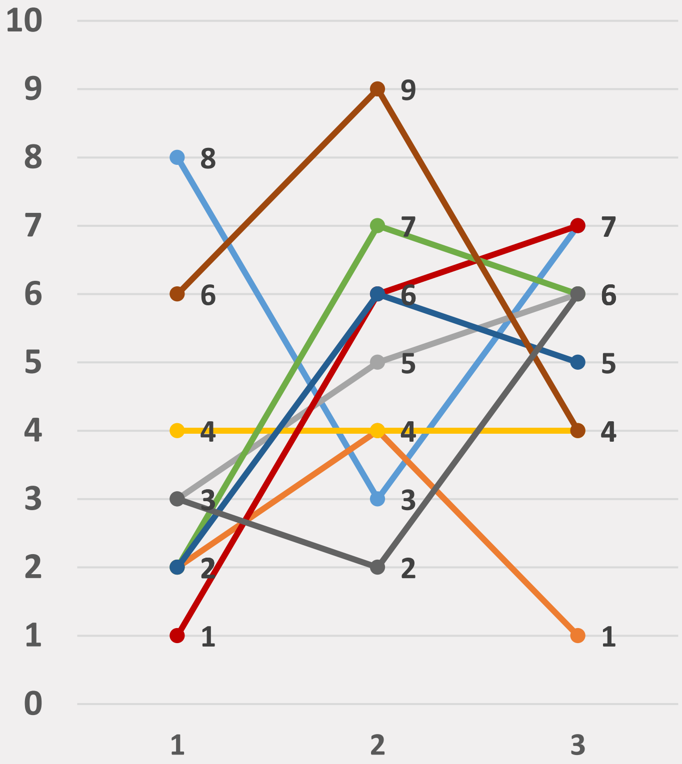
3: (1,6,7), (2,4,1), (3,2,6)



Pareto points - Exercises

greater = better

8	3	7
2	4	1
3	5	6
4	4	4
1	6	7
2	7	6
2	6	5
6	9	4
3	2	6



(How many) Pareto points ?

5: (6,9,4), (8,3,7), (2,7,6), (1,6,7), (3,5,6)

Pareto Algebra

The elements: sets of configurations

(Relevant) basic operators:

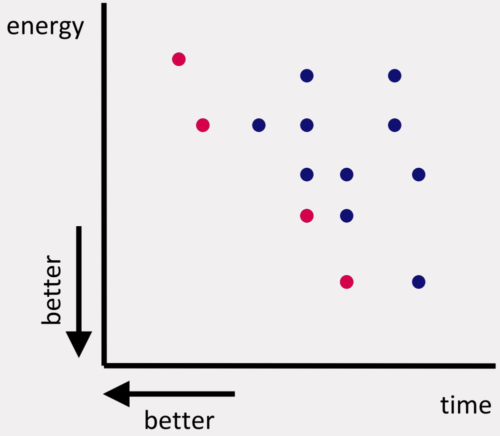
Minimization – Gives the Pareto points (optimal trade-offs)

Product – Cartesian product of configurations
e.g. application and platform configurations

Constraint – Selects solutions according to constraints
e.g. all application configurations with some minimal quality

Abstraction – Discards information about solutions
e.g. bandwidth usage in bandwidth × energy × quality configurations

Derived metric – Derive a new metric from other metrics
e.g. total power from power of components

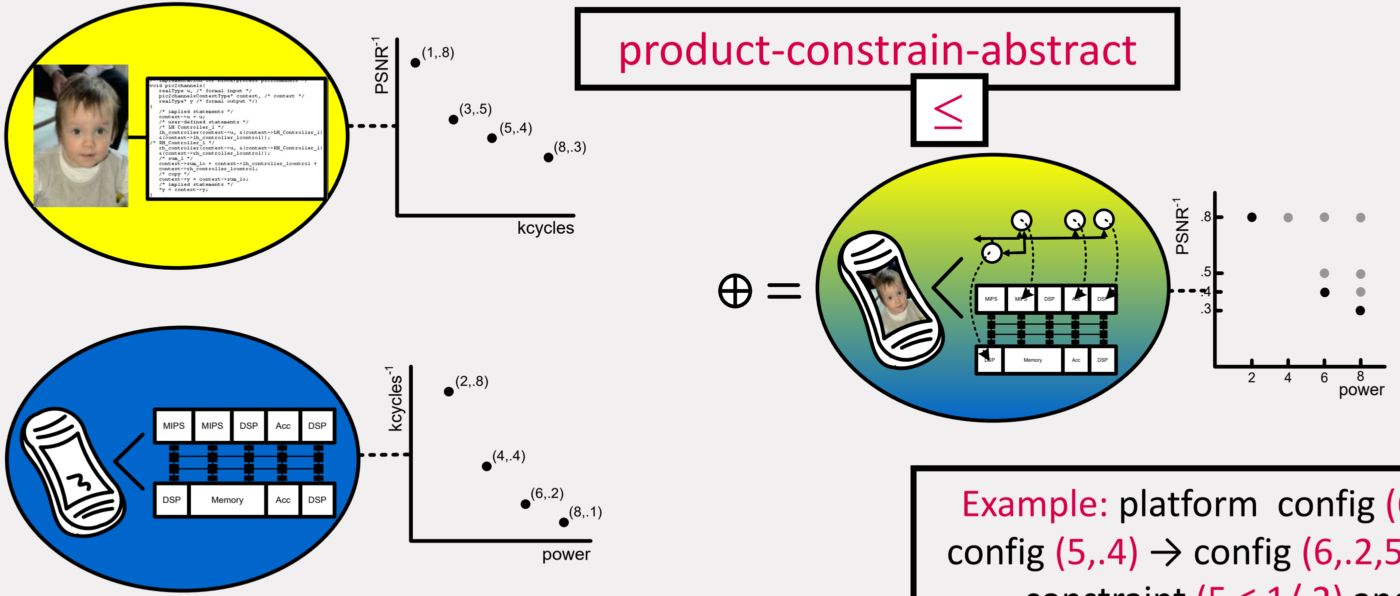


prerequisite

monotonicity

Pareto Algebra: compound operations

- producer-consumer operation – an example of a vertical composition

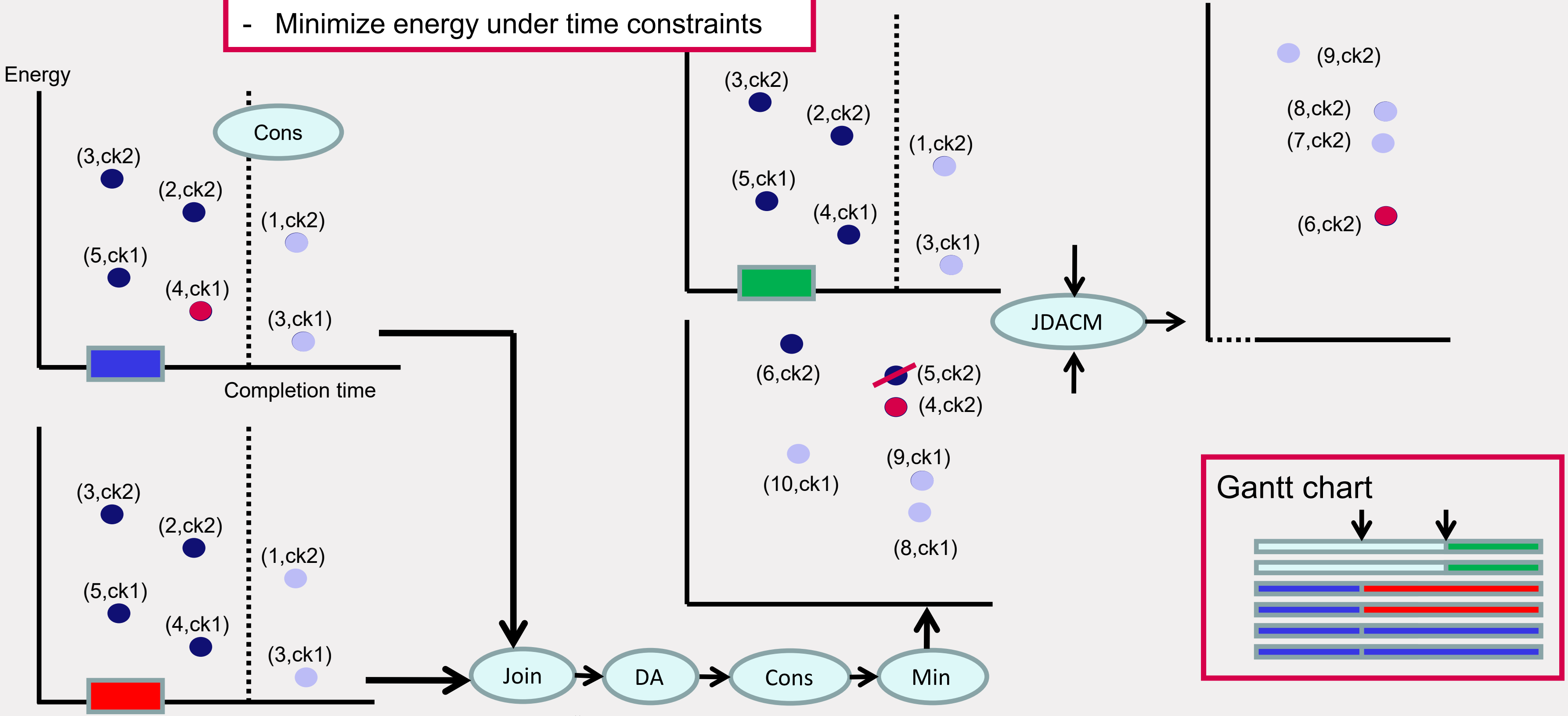


Example: platform config (6,.2), application config (5,.4) → config (6,.2,5,.4) → (6,.4) after constraint (5 ≤ 1/.2) and abstraction

- join operation (relational databases) = product-constrain-abstract

QRM

- Applications starting/stopping over time
- 6 procs, slow (ck1) and fast clocks (ck2)
- Minimize energy under time constraints



“enforce the same clock”

“derive cost, abstract”

ES ELECTRONIC SYSTEMS TU/e

QRM - Exercises

#proc	clock	time	energy
3	2	3	11
5	1	4	5
2	2	9	9
4	1	10	3
1	2	15	7
3	1	16	1

3	2	3	11
5	1	4	5
2	2	9	9
4	1	10	3
1	2	15	7
3	1	16	1

Constrain time < 12

(How many) points pruned ?
(from each of the tasks)

QRM - Exercises

#proc	clock	time	energy
3	2	3	11
5	1	4	5
2	2	9	9
4	1	10	3

3	2	3	11
5	1	4	5
2	2	9	9
4	1	10	3

Join: enforce the same clock

- Product
- Constrain – clock = clock
- Abstract – one clock

How many points does the end result have?

How many dimensions does the end result have ?

QRM - Exercises

#proc	clock	time	energy
3	2	3	11
5	1	4	5
2	2	9	9
4	1	10	3

3	2	3	11
5	1	4	5
2	2	9	9
4	1	10	3

#proc	clock	time	energy	#proc	time	energy
3	2	3	11	3	3	11
3	2	3	11	2	9	9
5	1	4	5	5	4	5
5	1	4	5	4	10	3
2	2	9	9	3	3	11
2	2	9	9	2	9	9
4	1	10	3	5	4	5
4	1	10	3	4	10	3

eight 7d configurations

How many points does the end result have?

How many dimensions does the end result have ?

QRM - Exercises

#proc	clock	time	energy
6	2	3	22
5	2	9	20
10	1	4	10
9	1	10	8
5	2	9	20
4	2	9	18
9	1	10	8
8	1	10	6

#proc	clock	time	energy	#proc	time	energy
3	2	3	11	3	3	11
3	2	3	11	2	9	9
5	1	4	5	5	4	5
5	1	4	5	4	10	3
2	2	9	9	3	3	11
2	2	9	9	2	9	9
4	1	10	3	5	4	5
4	1	10	3	4	10	3

eight 7d configurations

- Derive total #proc (+), time (max), energy (+)
- Abstract task properties

How many/which points result?

QRM - Exercises

#proc	clock	time	energy
6	2	3	22
5	2	9	20
10	1	4	10
9	1	10	8
5	2	9	20
4	2	9	18
9	1	10	8
8	1	10	6

six 4d configurations remain
(two symmetric configurations)

- Derive total #proc (+), time (max), energy (+)
- Abstract task properties

QRM - Exercises

#proc	clock	time	energy
6	2	3	22
5	2	9	20
10	1	4	10
9	1	10	8
5	2	9	20
4	2	9	18
9	1	10	8
8	1	10	6

six 4d configurations remain
(two symmetric configurations)

Constrain #proc <= 6

How many/which points result?

QRM - Exercises

#proc	clock	time	energy
6	2	3	22
5	2	9	20
4	2	9	18

three configurations remain

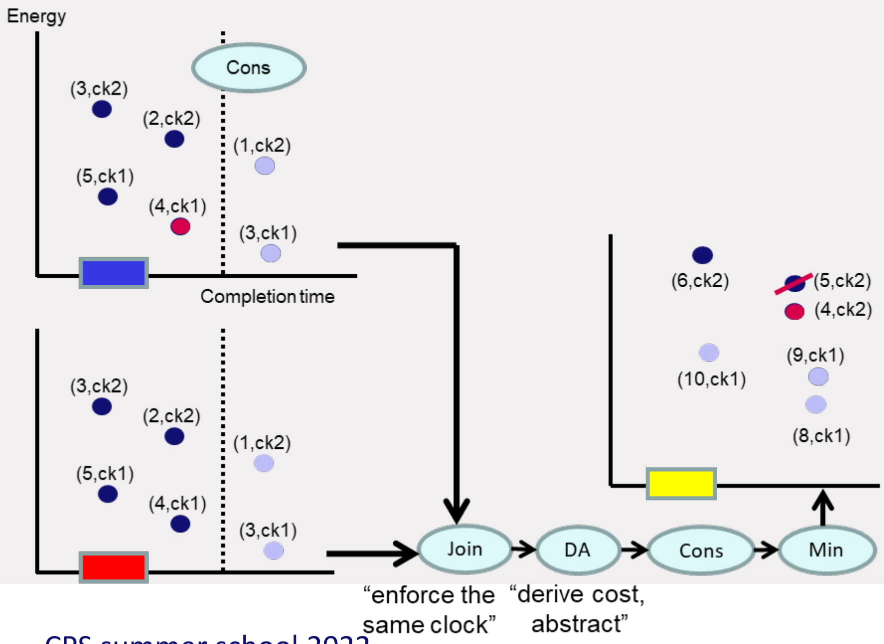
Min: Pareto points

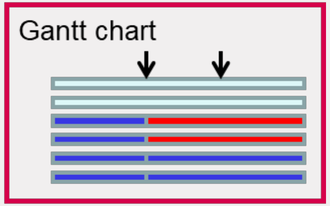
Pareto points?

QRM - Exercises

#proc	clock	time	energy
6	2	3	22
5	2	9	20
4	2	9	18

two Pareto points !





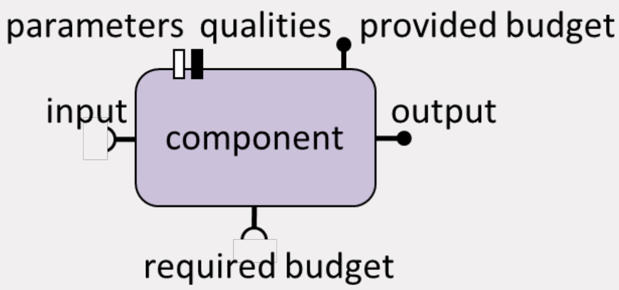
Overview

- Quality & Resource Management (QRM)
- Pareto analysis
- Component Interface Model for QRM
- Quality and Resource Management Language and Tools

Conclusions

model-driven QRM

- make QRM interfaces explicit
- orthogonal to functional component-based design
- founded in Pareto algebra

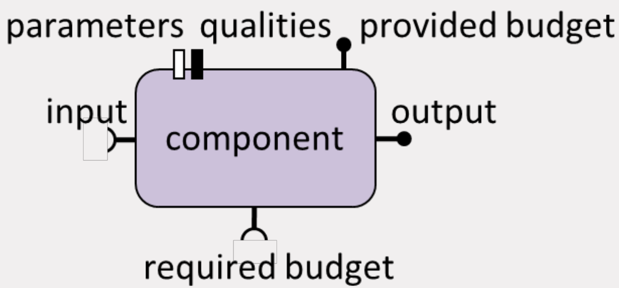


M. Hendriks, M. Geilen, K. Goossens, R. De Jong, T. Basten. Interface Modeling for Quality and Resource Management. Logical Methods in Computer Science, LMCS. **17**(2), 19:1-19:34, 26 May 2021, **open access**.

Questions?

model-driven QRM

- make QRM interfaces explicit
- orthogonal to functional component-based design
- founded in Pareto algebra



M. Hendriks, M. Geilen, K. Goossens, R. De Jong, T. Basten. Interface Modeling for Quality and Resource Management. Logical Methods in Computer Science, LMCS. **17**(2), 19:1-19:34, 26 May 2021, **open access**.

qrml.org

Twan Basten
a.a.basten@tue.nl