
GEEGLEE'S
SE KNOWLEDGE BOOK
V1.0

# Geeglee®

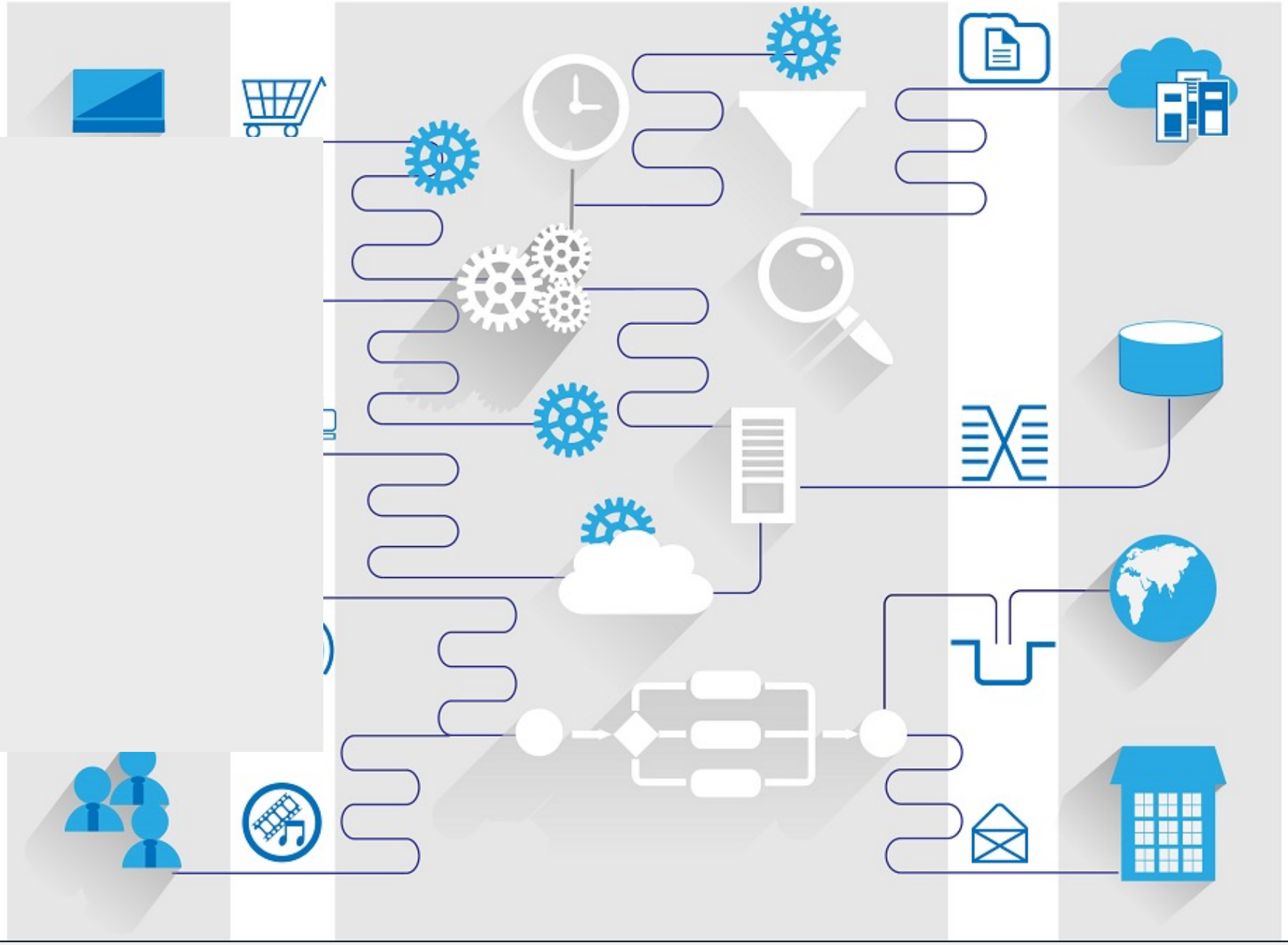
## AUGMENTED HUMAN INTELLIGENCE

Know, Understand, Plan and Act

LA
PLACE
STRATÉGIQUE

# 01

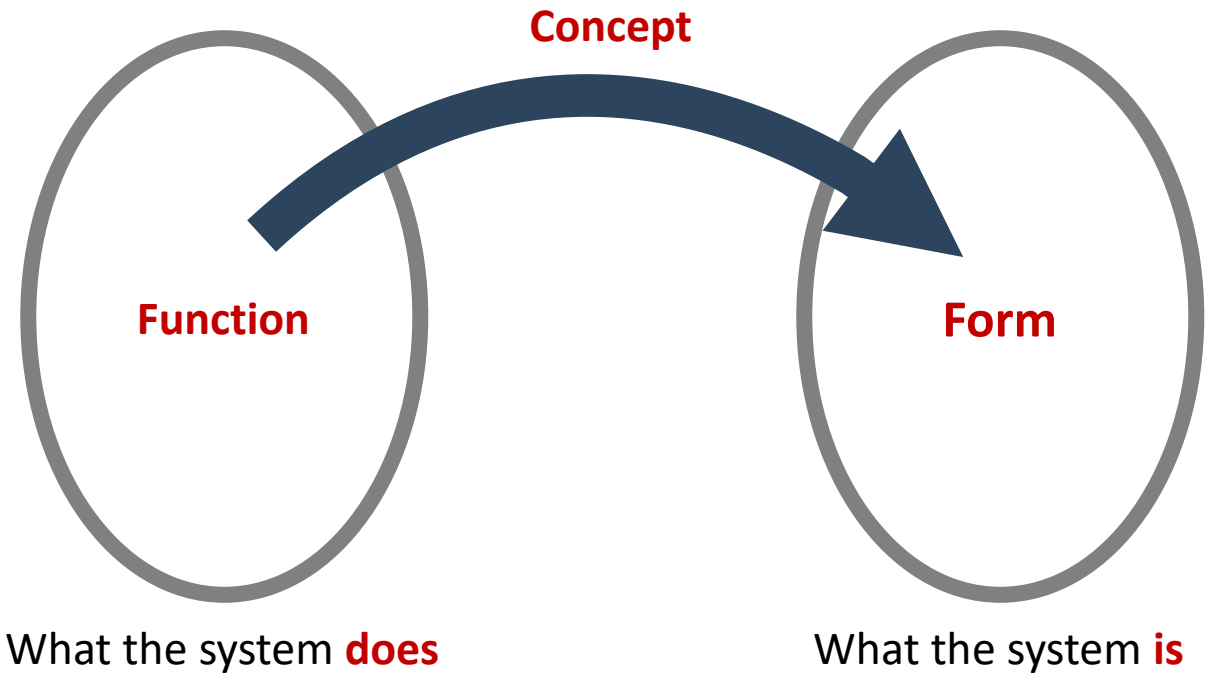
## Architecture



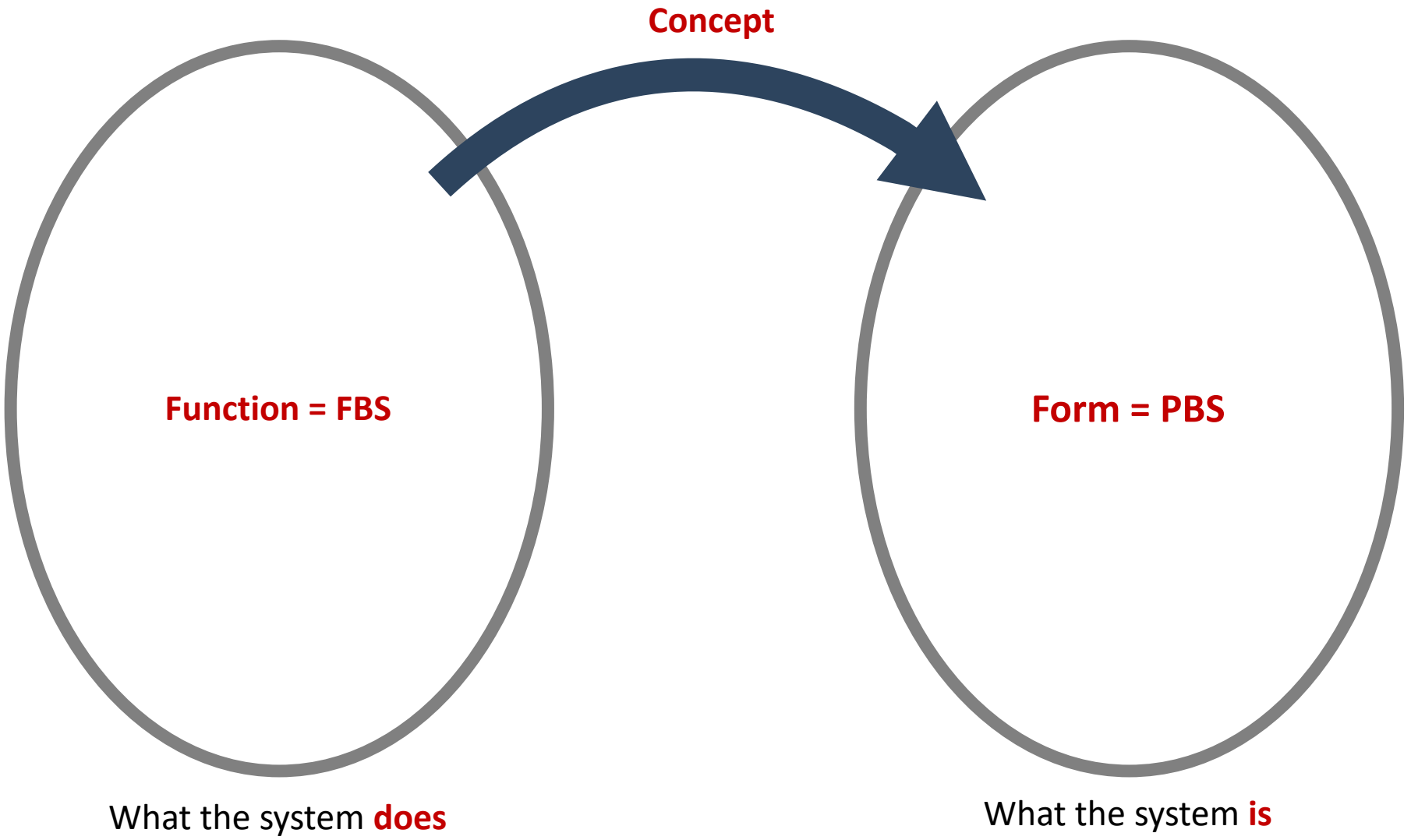
# Architecture definition?

The embodiment of **concept** and the allocation of **function(s)** to elements of **form(s)**, and definition of **interfaces** among the elements and with the surrounding **context**.

*(Crawley)*



# Architecture in detailed view

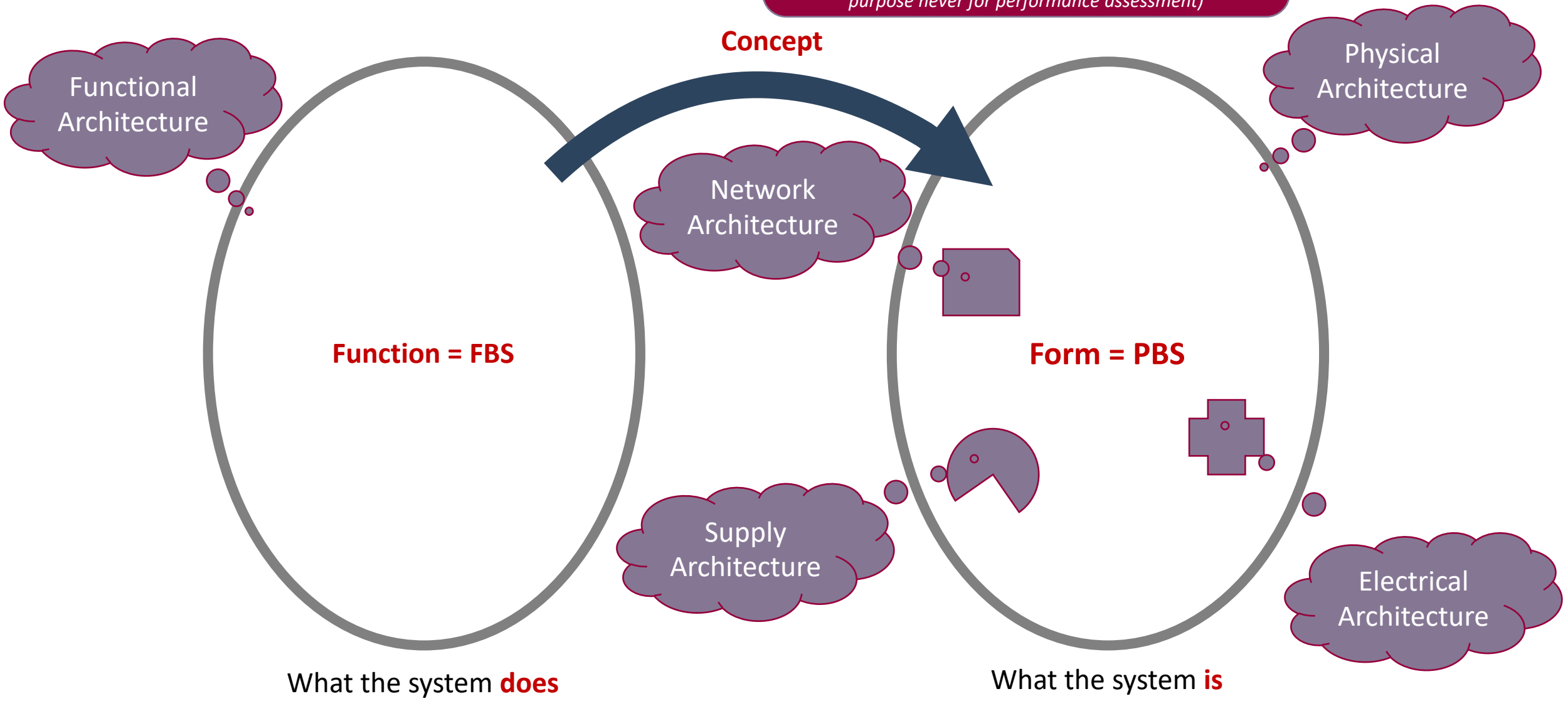


# So… when we talk about…

The architecture is the overall!

$\iiint_1^n architectures$

*(Mapping each individualy is only for understanding purpose never for performance assessment)*

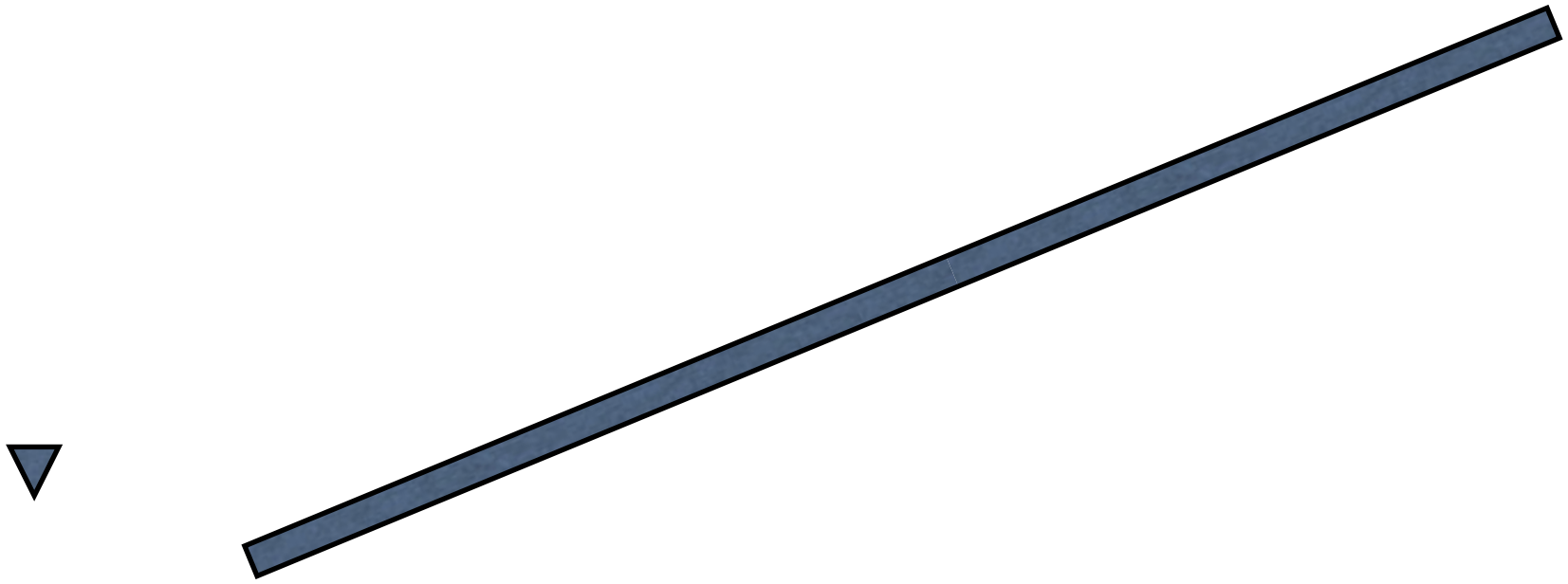


**Emergence**
**is the reason why we build systems!**

$$From \sum Architectures \; to \iiint Architectures$$

# Concept of Emergence

## WHAT WE CAN DO WITH THESE TWO PARTS?



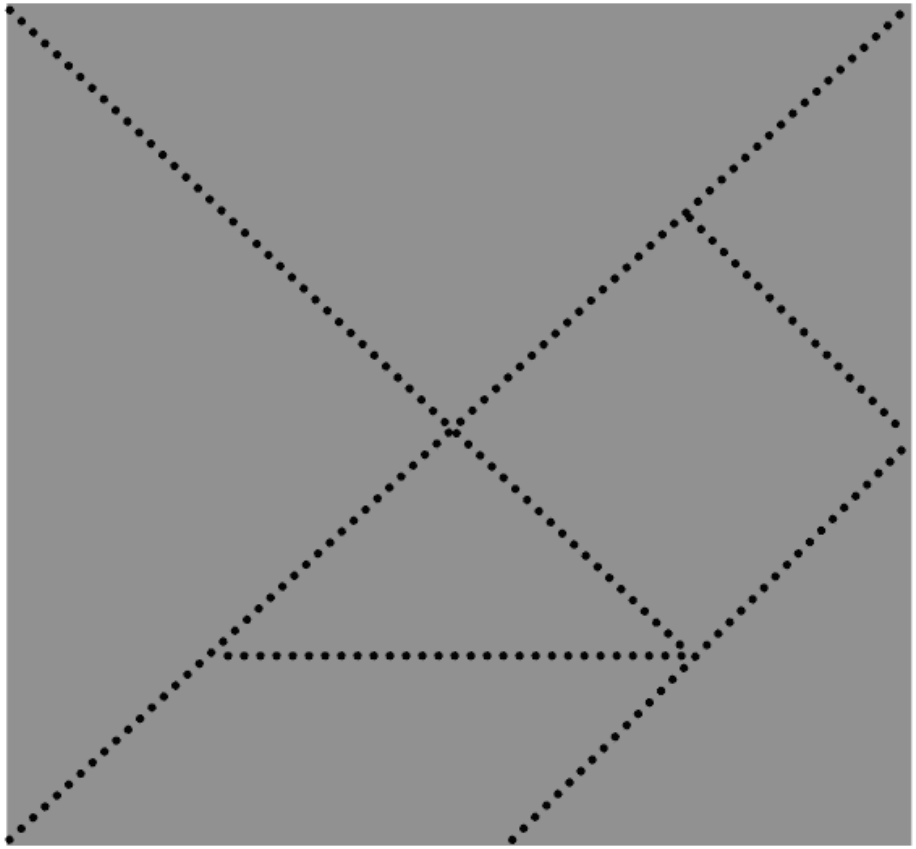
# Concept of Emergence

## WHAT WE CAN DO WITH THESE TWO PARTS?

# Emergence

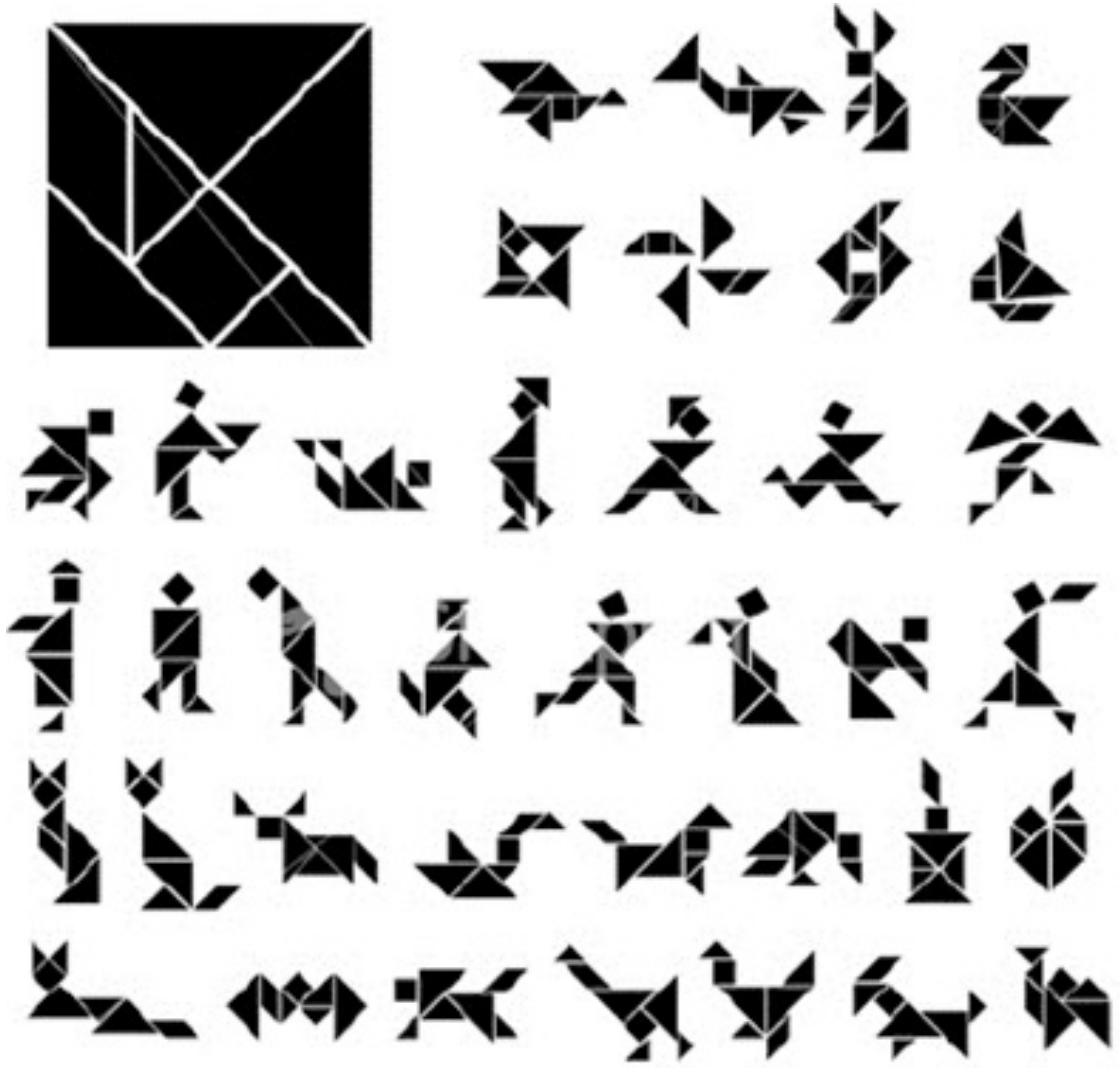
## THE CHINESE TANGRAM PUZZLE

- The goal of the Chinese tangram is to create a new form of shape out of seven polygons
- The unique shape may be a form of animals, other polygons, humans, things, and anything you can think of
- Just make sure that each piece should be connected but not overlapping each other



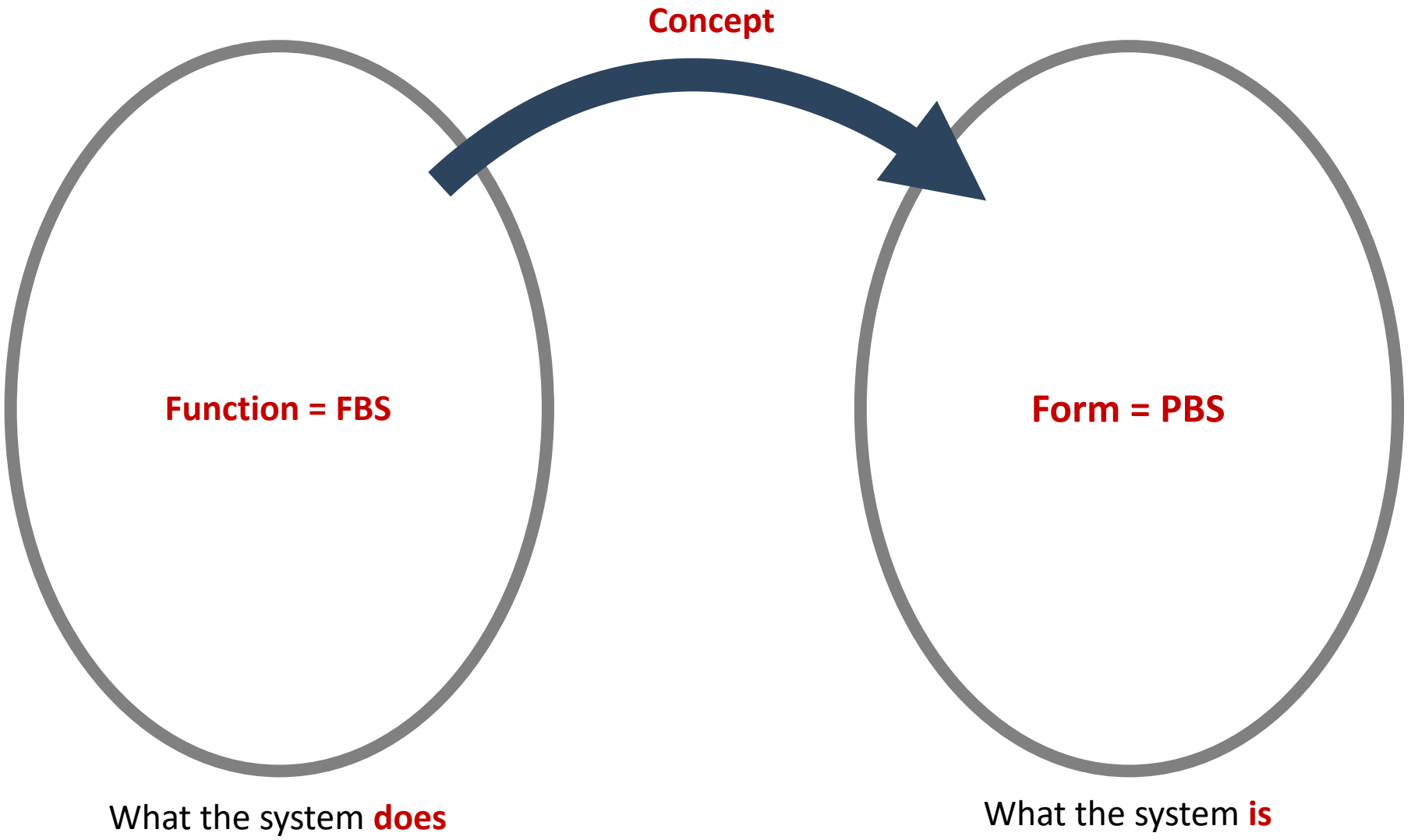
# Emergence

## DID YOU FIND ALL THESE FORMS?



For sure, much more exist!

# Architecture in detailed view

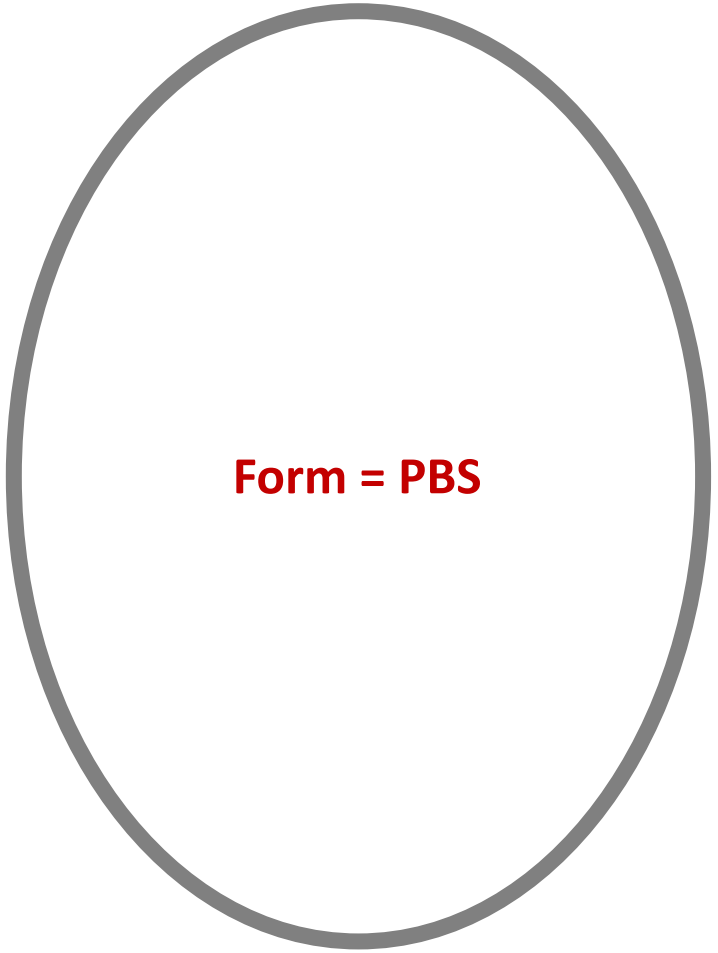


# Architecture in detailed view

## FIRST, LET'S FOCUS ON FORM!

In the course, we introduce form, first, because it's much more easy to think about physics

But, as you will discover during the course, it's much more relevant to think first functionnaly and then physicaly (even if both can create opportunity). That way, you more sure about exploring any opportunities!




What the system **is**

FORM A

FORM B

# Does these forms have the same main functions?

*(main or principal or external)*

FORM A

Yes!

FORM B

# Why these two bridges do not have the same PBS?

FORM A

FORM B

**Constraints are not the same**

*(may be functional requirements too)*

# Does these two forms have the same FBS?

FORM A

FORM B

**No!**

(the same PBS = not the same FBS)

# Constraints and functional requirements are not the only things that might change an architecture!

## TECHNOLOGIES (EMBEDDED IN PBS) LEAD TO OPPORTUNITIES, OR TO CONSTRAINTS, INTO FBS!

Then, we can say that:

- ✓ FBS & PBS are correlated!
- ✓ A decision into one xBS will influence the others!

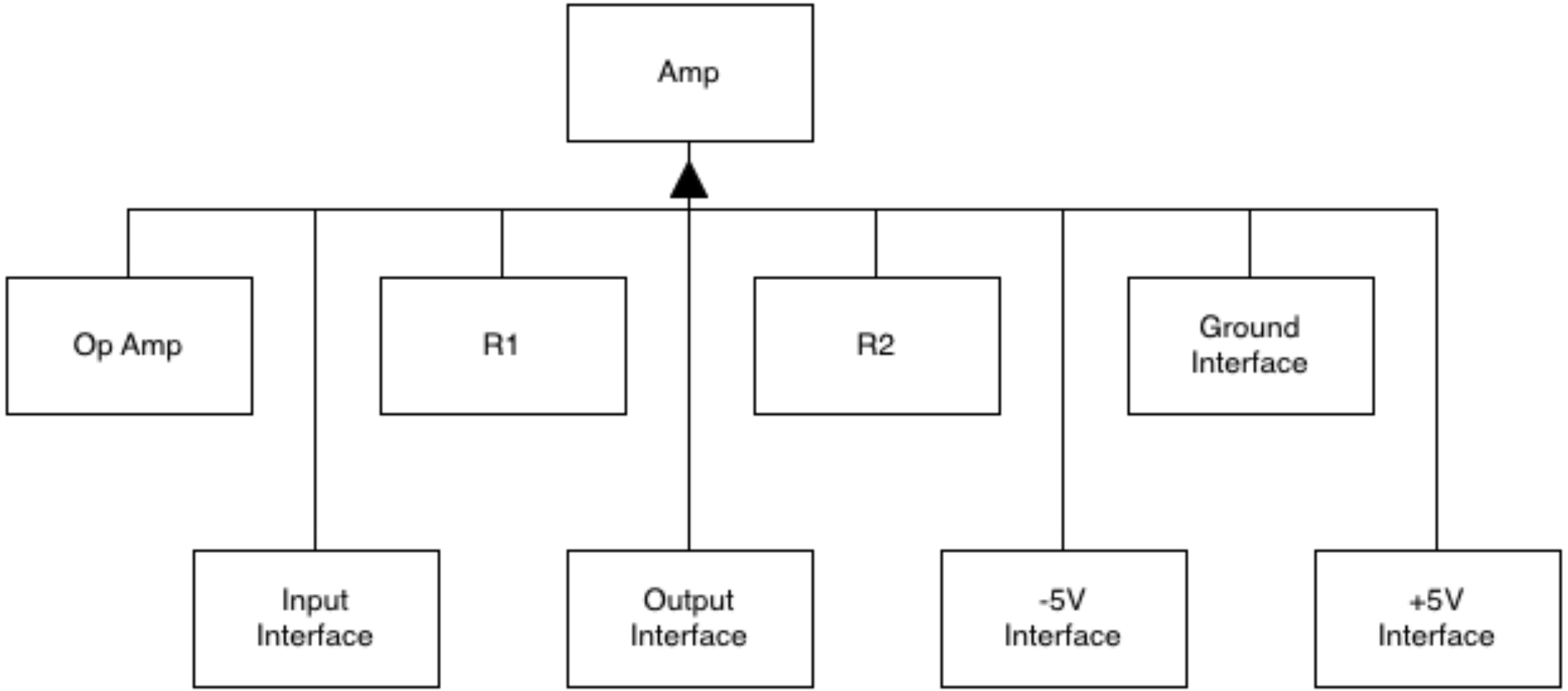
# A recursive Form lead to a recurvive FBS!

**Form** embodies
what the system '**is**'

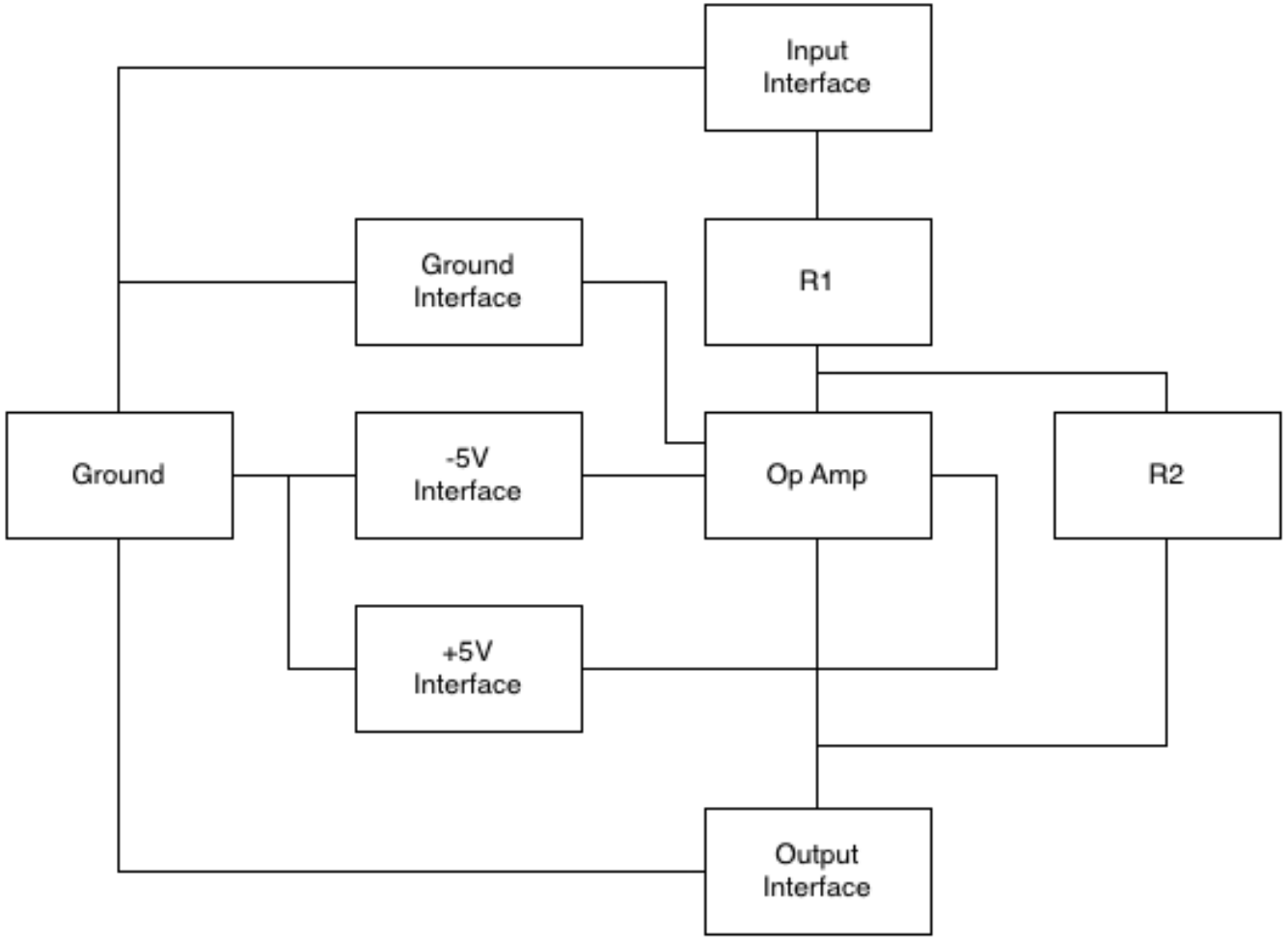
A system is composed of element related by a structure

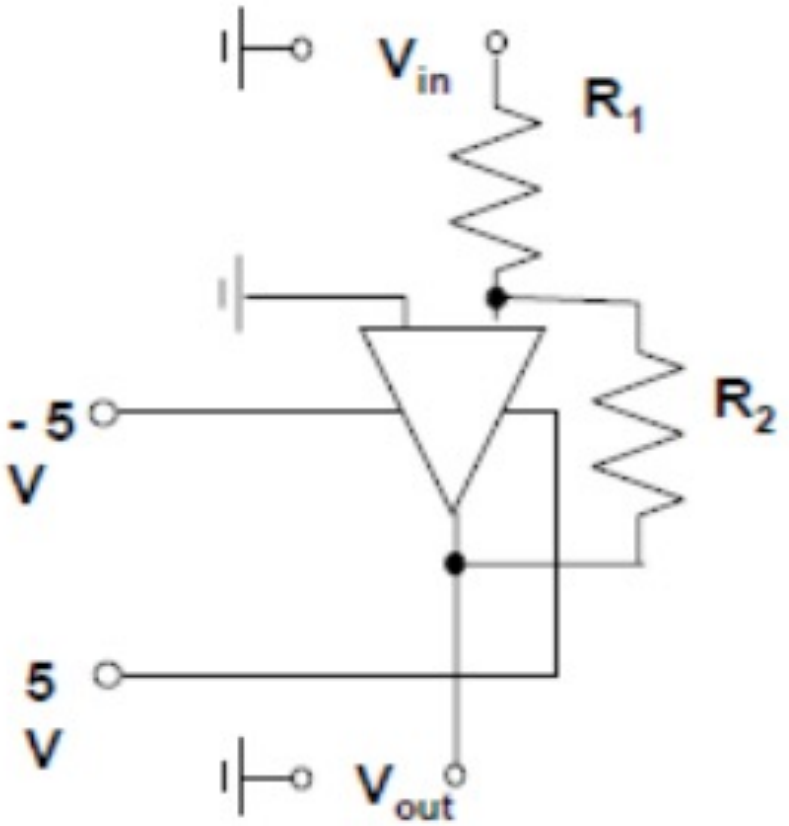
**Product Breakdown Structure**

# Elements = System Decomposition



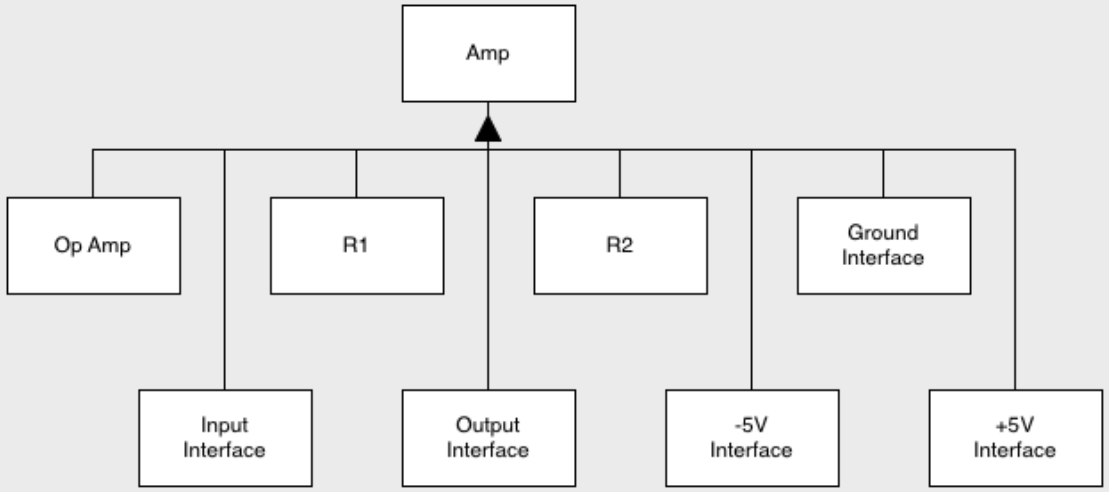
# **Structure** = formal relationships between elements

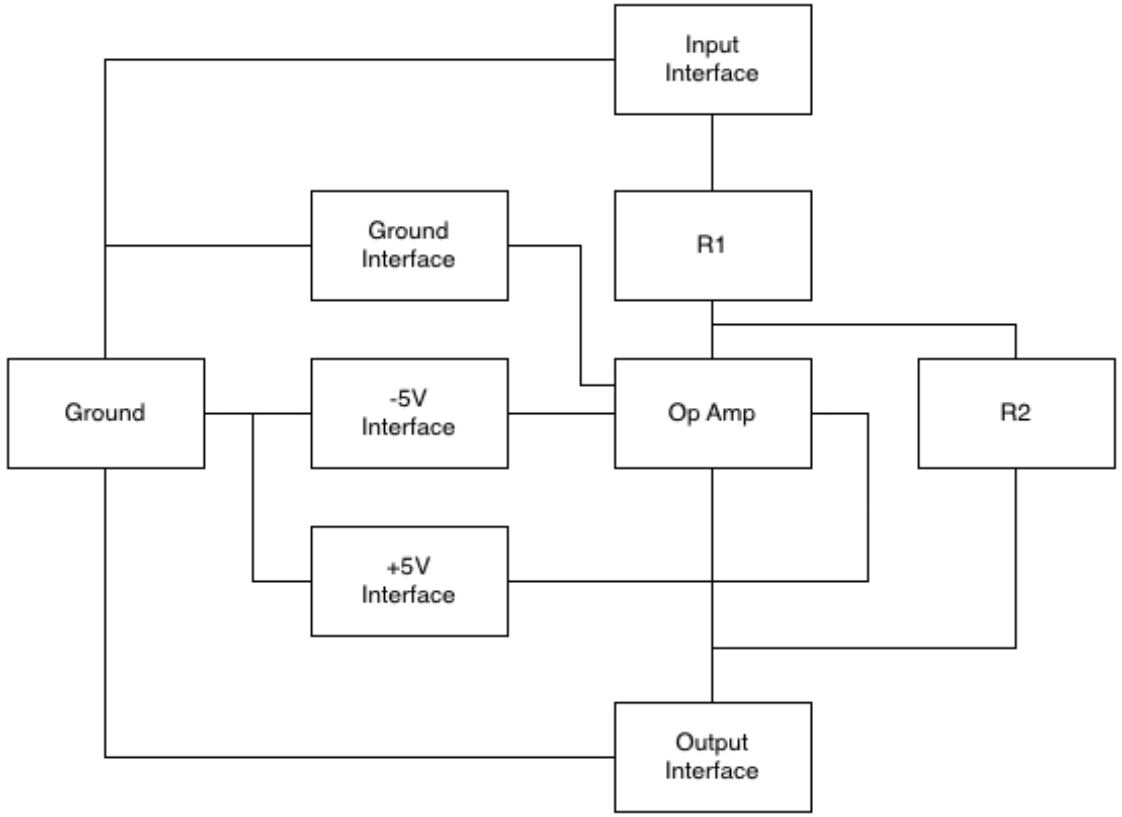


# Product Breakdown Structure (PBS) is thus:

BREAKDOWN & STRUCTURE



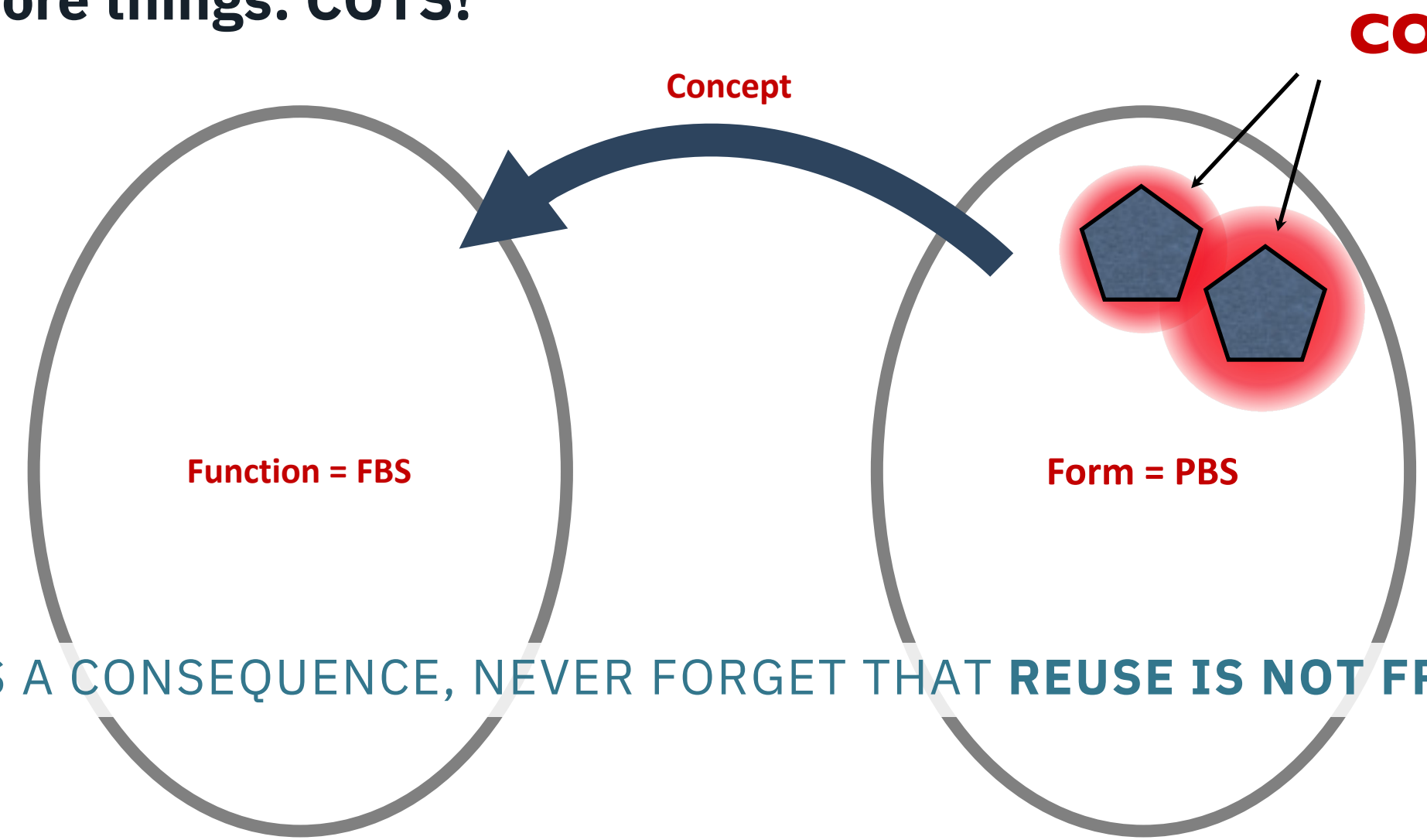




Do not forget interfaces!


90% of performances losses come from multi-disciplinary interfaces

*(2007)*

# One more things: COTS!




AS A CONSEQUENCE, NEVER FORGET THAT **REUSE IS NOT FREE!**

# Reverse Engineering

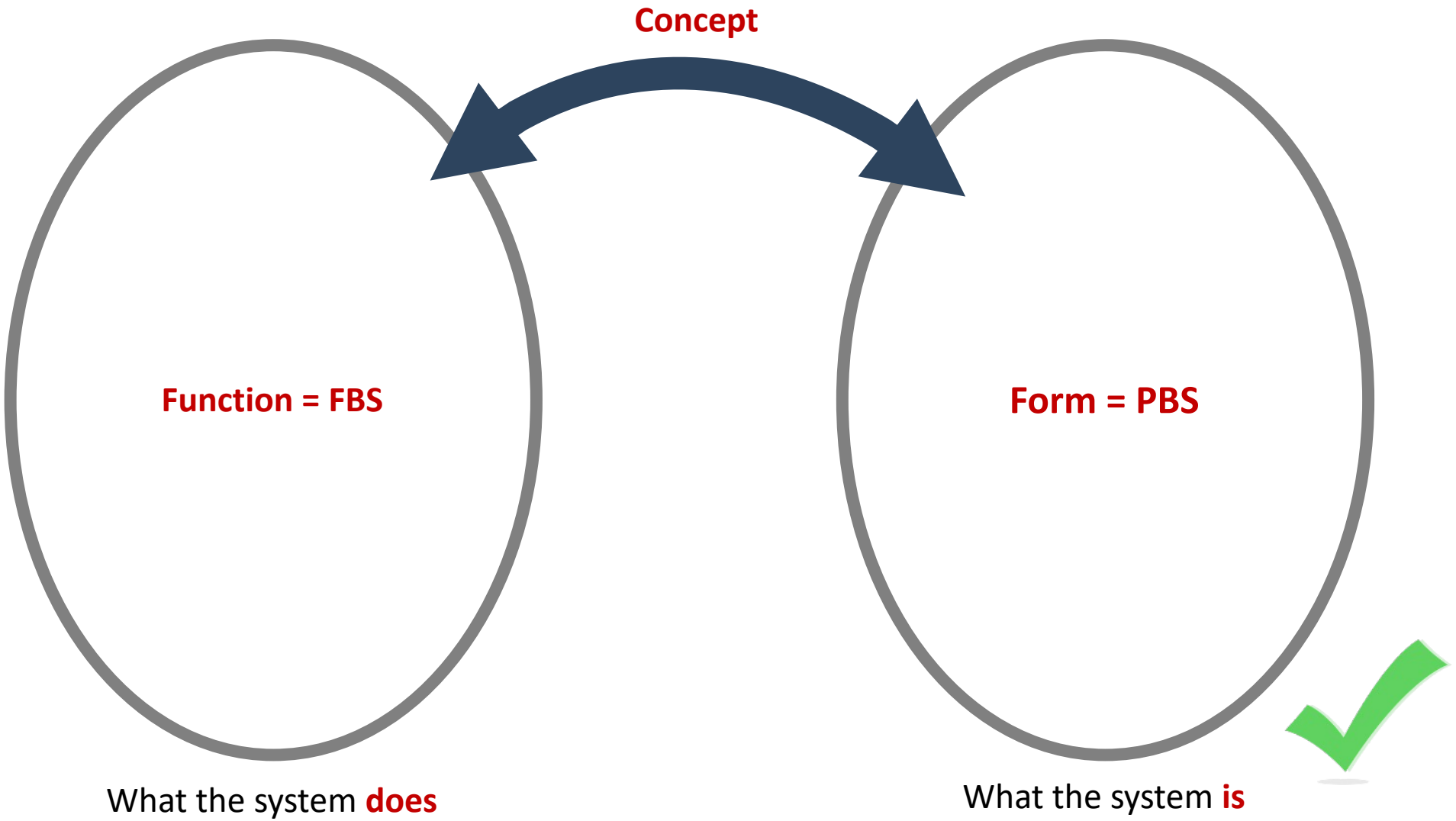
The tool when you need to move from PBS to FBS!

Engineers always think in terms of product, but we are more creative when thinking in term of functions

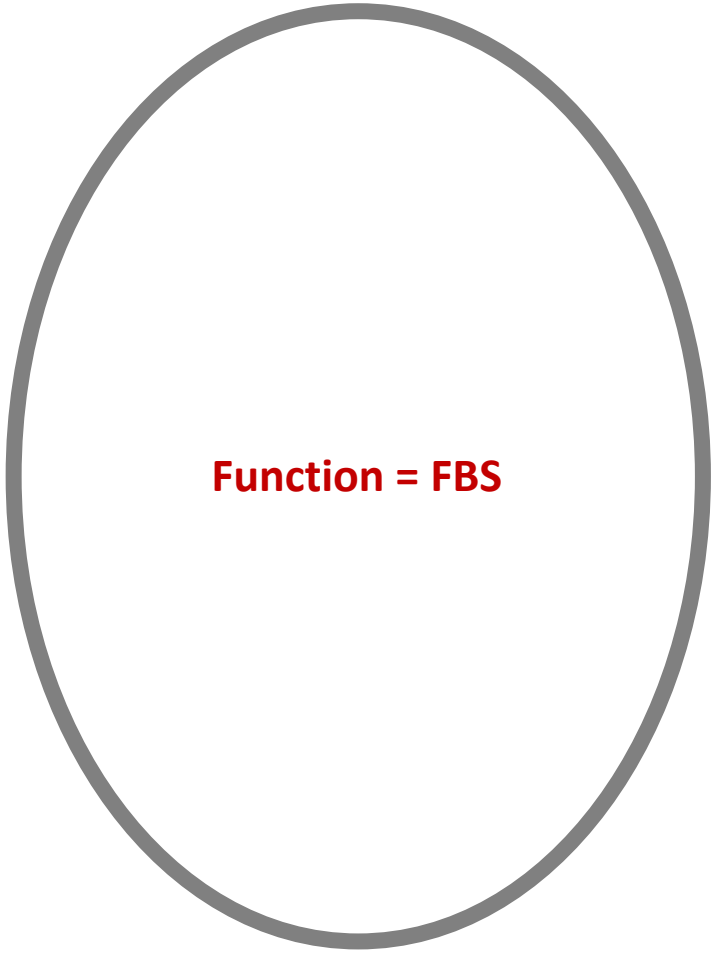
# Start in practice: what are the functions?


P791 Lockheed Martin

# Architecture in detailed view



# Architecture in detailed view




What the system **does**

## SECOND, LET'S FOCUS ON FUNCTION!

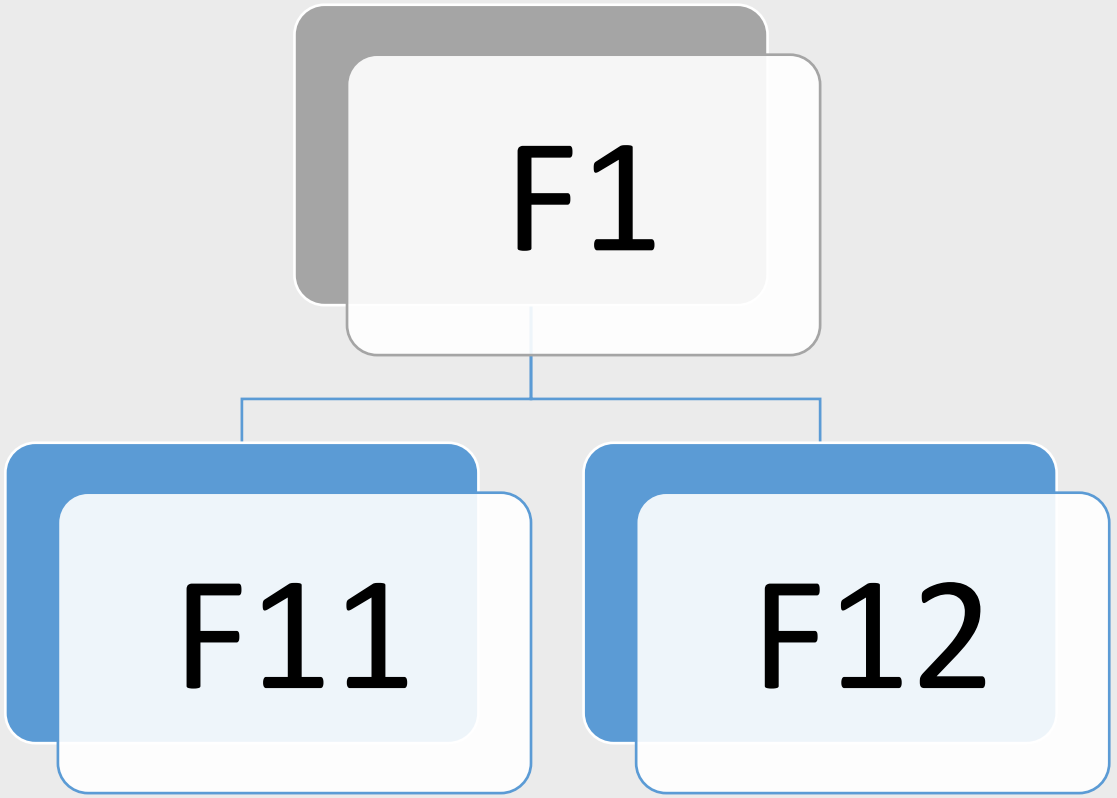
***EVEN IF IT SHOULD BE YOUR FIRST CONCERN!***

As the PBS, the Functional Breakdown Structure is composed of:

- ✓ a Breakdown, and
- ✓ a Structure

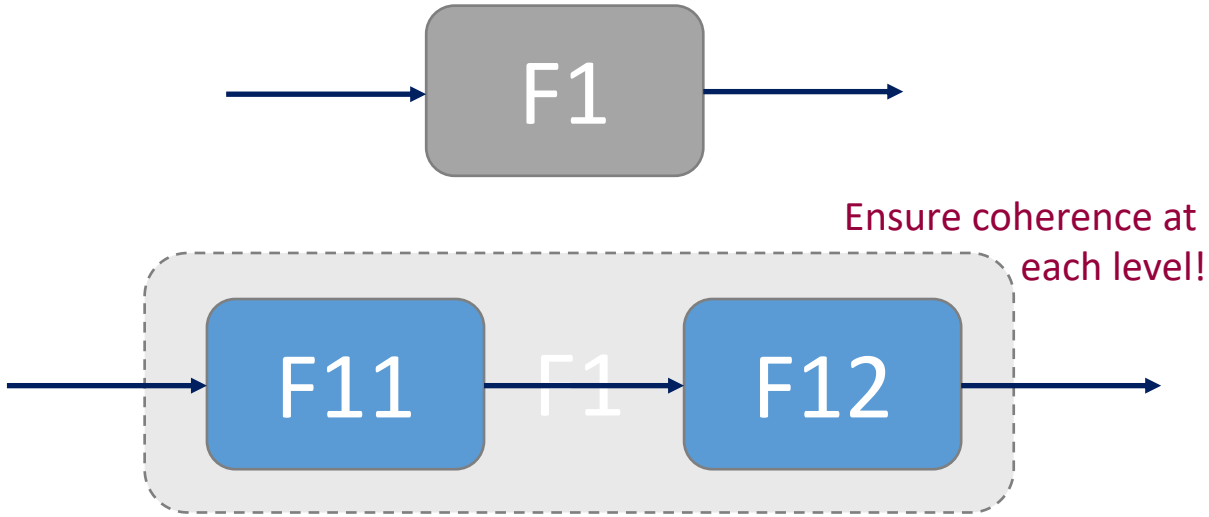
# Function Breakdown Structure (FBS) is thus:

BREAKDOWN



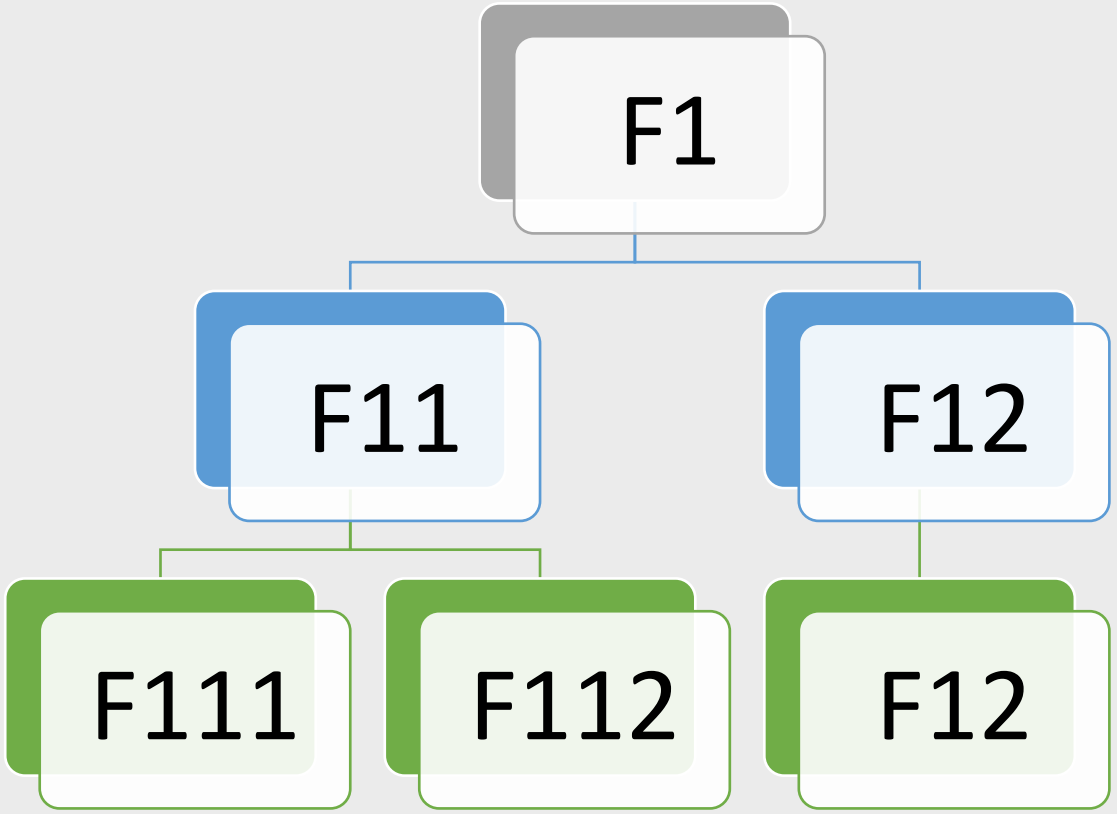

&

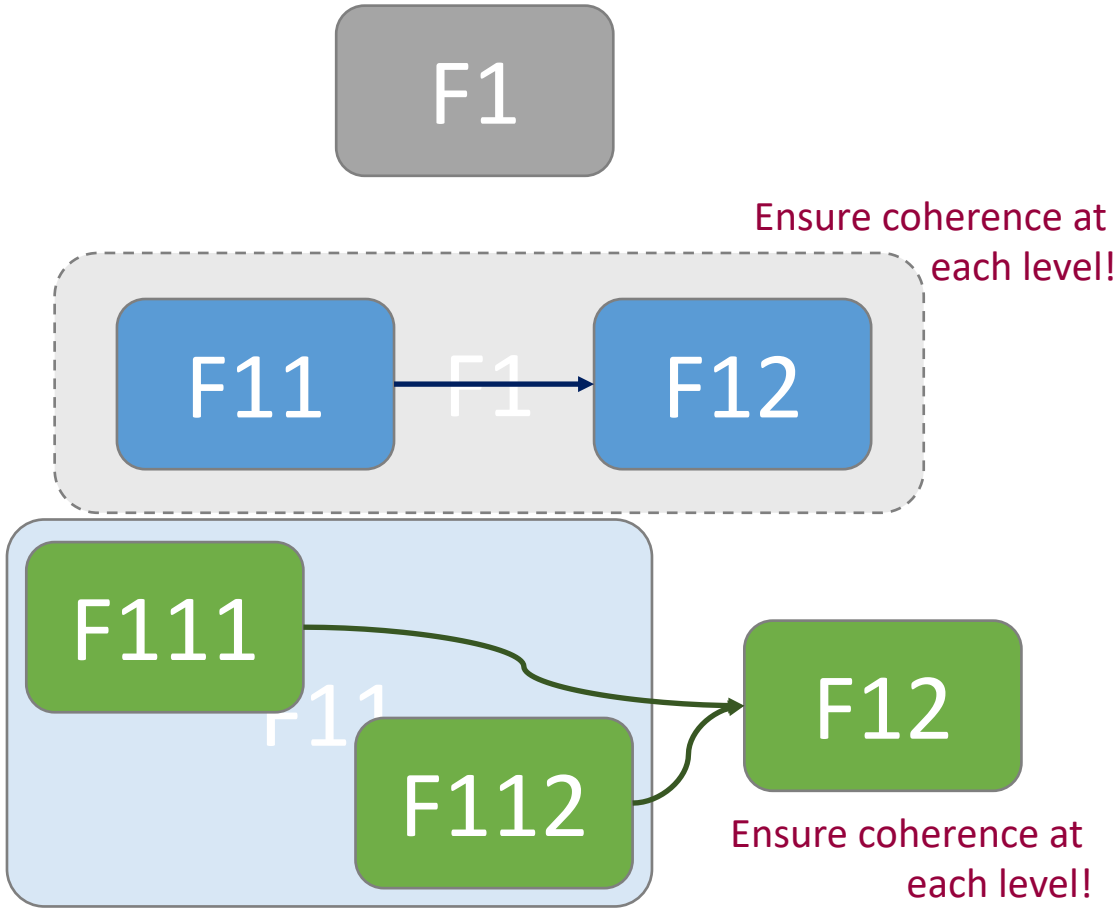
STRUCTURE = FUNCTIONAL FLOW



# Function Breakdown Structure (FBS) is thus:

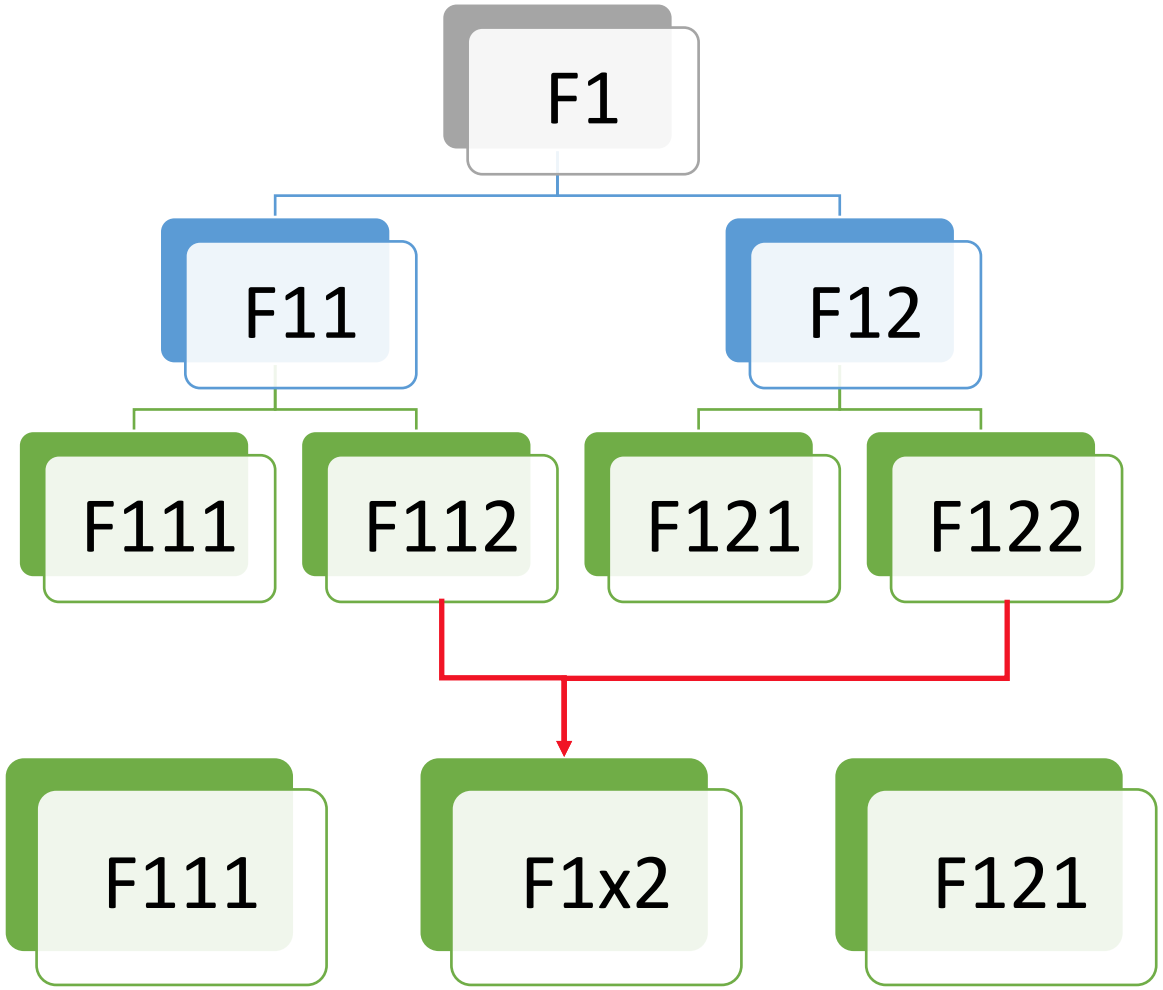
BREAKDOWN & STRUCTURE = FUNCTIONAL FLOW



# Merged Function?

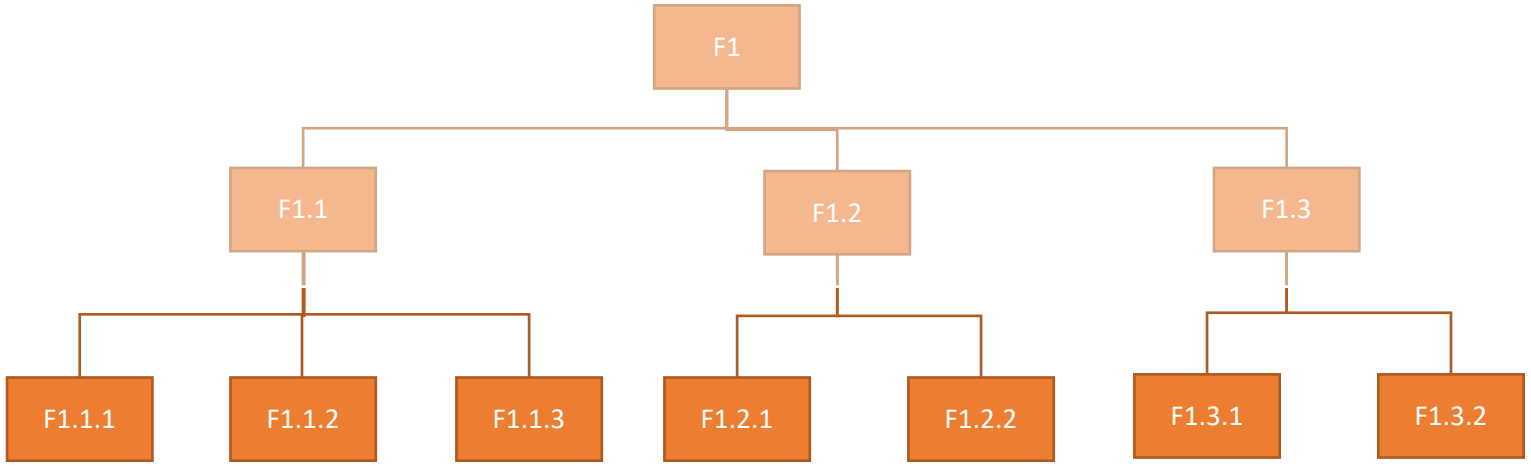
## CAN WE MERGE FUNCTIONS?




Yes and No! It depends:

- You CANNOT merge two functions that was given directly by the over-system.
  - If there was separeted, it's probably for a safety reason... (ex: speed CPU)
- If it make sens, you CAN merge two functions that are under your responsability (functions into your Internal Functional Analysis)
- In any case, it's highly recommanded to valide the merge with the over-system architect.

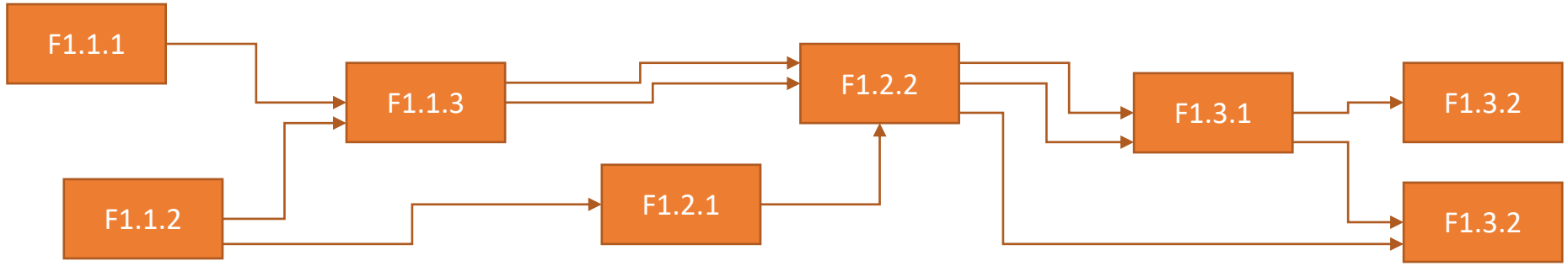
# Two main modeling techniques for function modelling




**Functional decomposition**

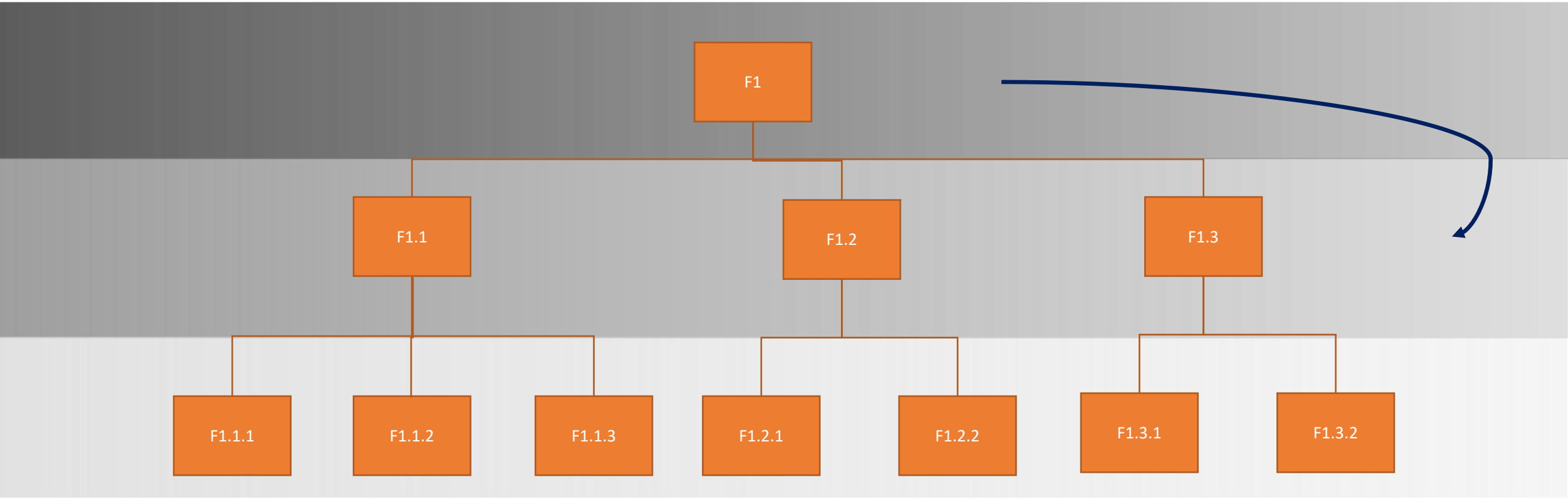
*At each level, you MUST manage the coherence!!!*

**Functional Flows**



# One more thing

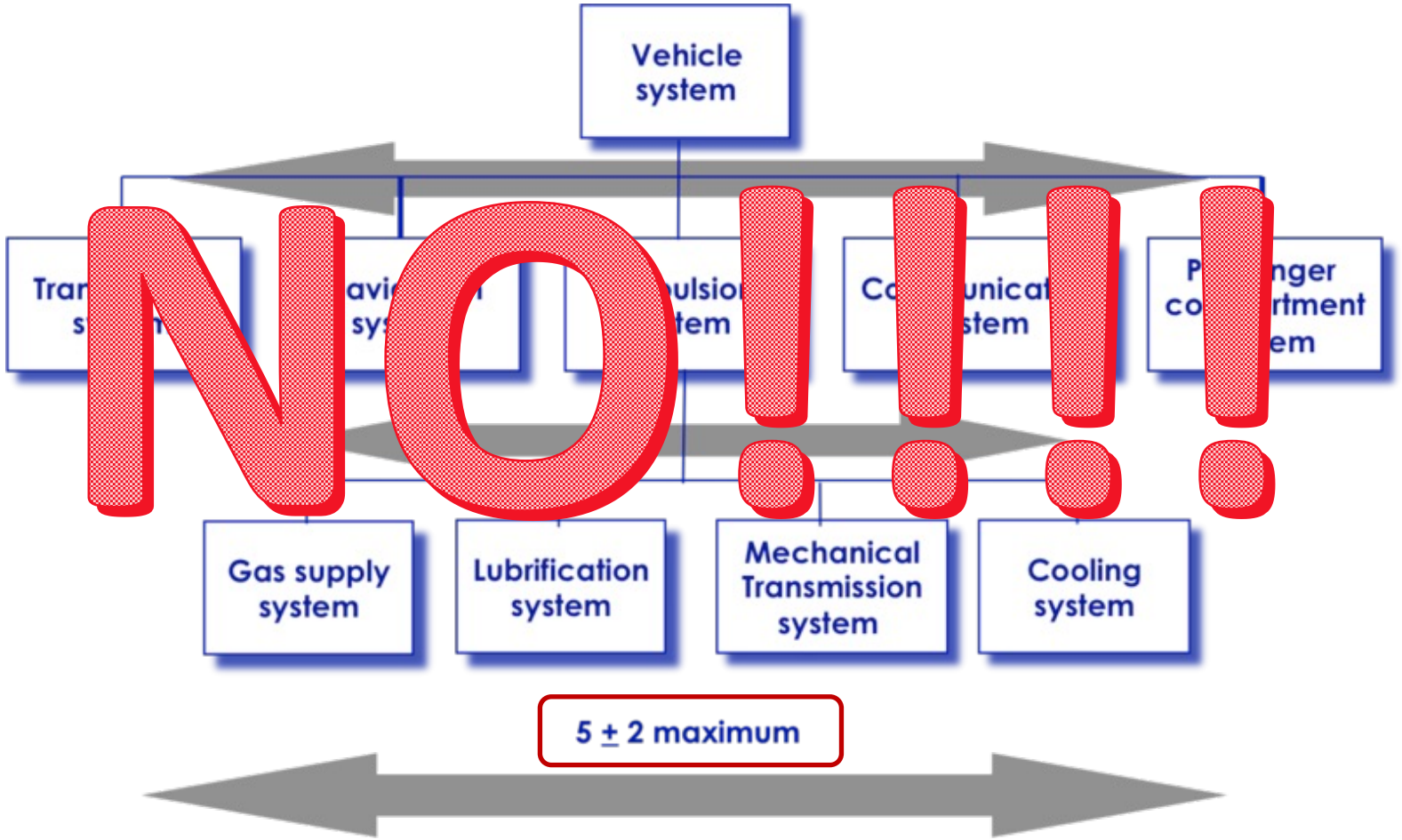
## HOW DO YOU GO FROM ONE LEVEL TO THE NEXT ONE?




**Functional decomposition**

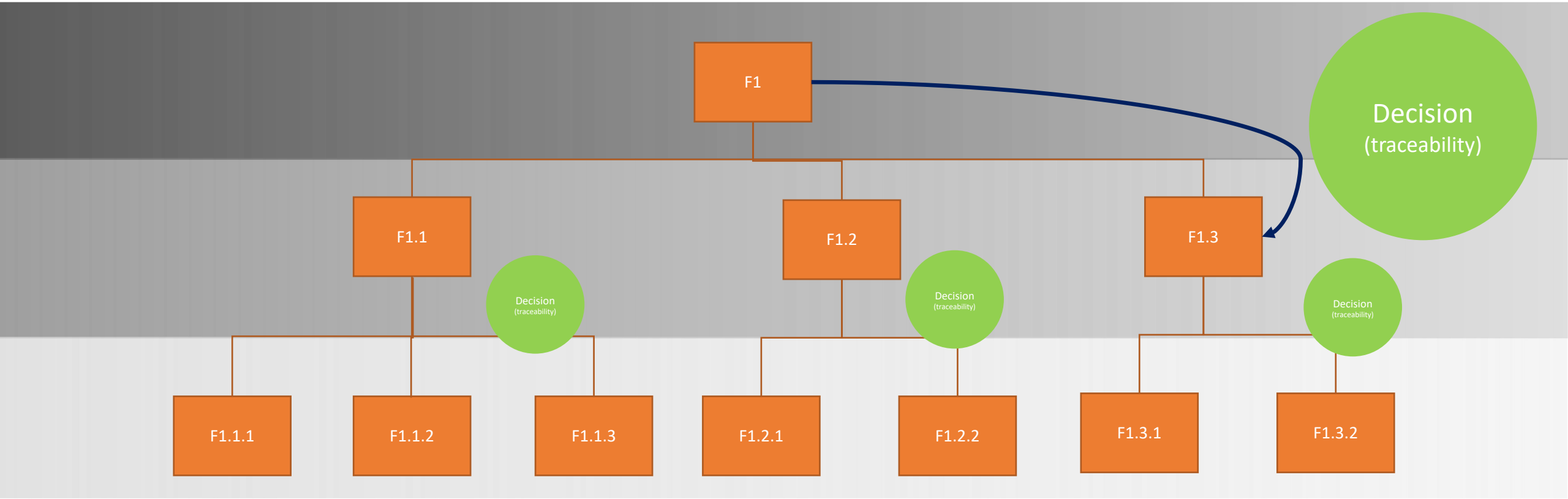
# One more thing

## HOW DO YOU GO FROM ONE LEVEL TO THE NEXT ONE?



# One more thing: Decision!

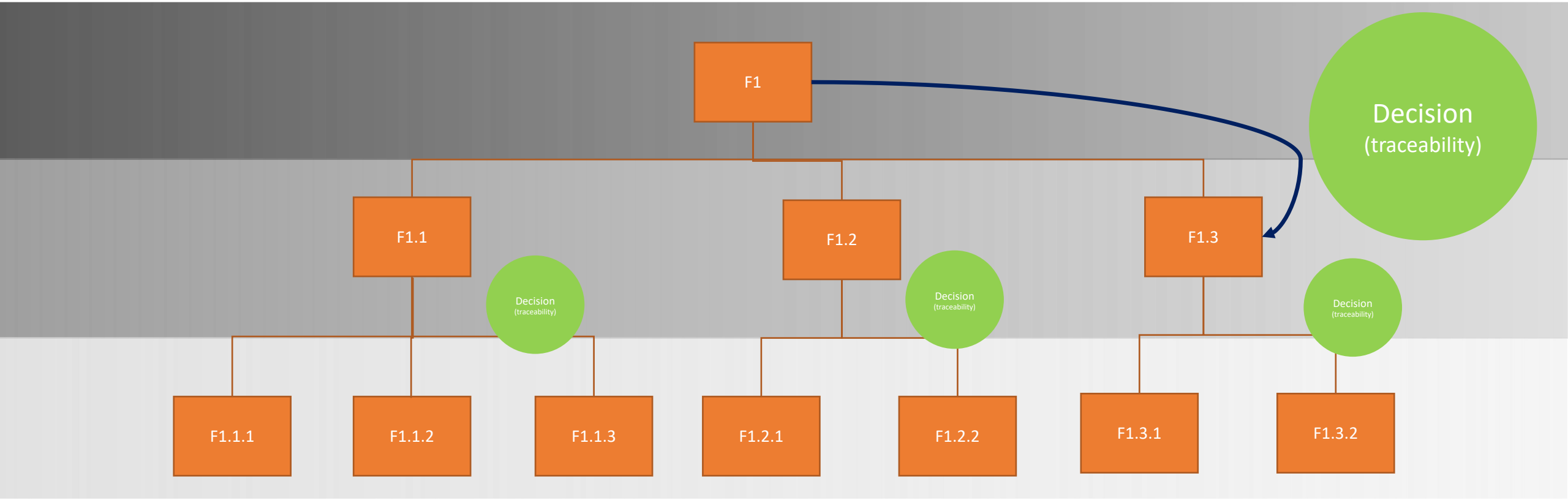
## YOU CAN TAKE MAXIMUM ONE DECISION PER LAYER AND BREAKDOWN!




**Functional decomposition**

# Introduction to decision sequence management

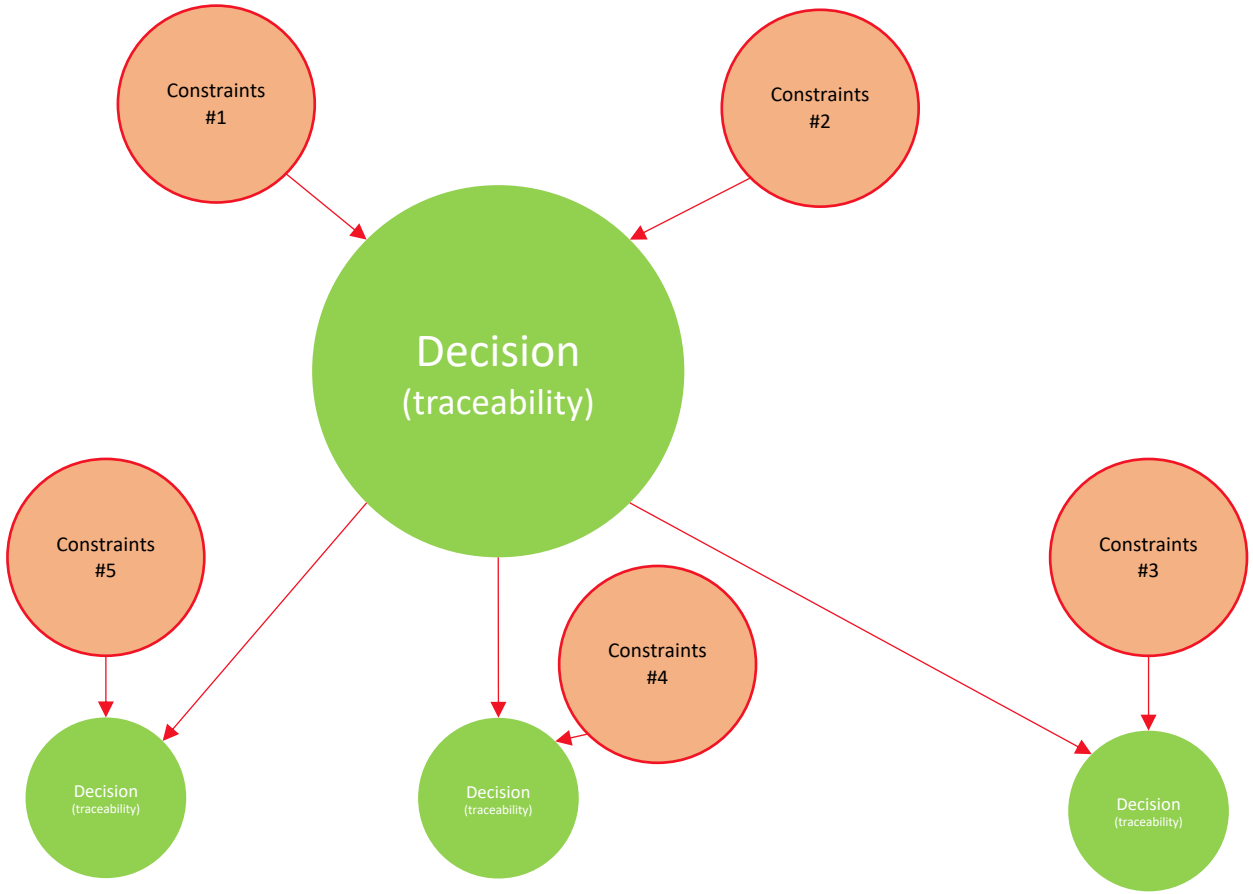
## WHAT ORDER SHOULD YOU MAKE?




**Functional decomposition**

# Introduction to decision sequence management (DBS)

## WHAT ORDER SHOULD YOU MAKE?

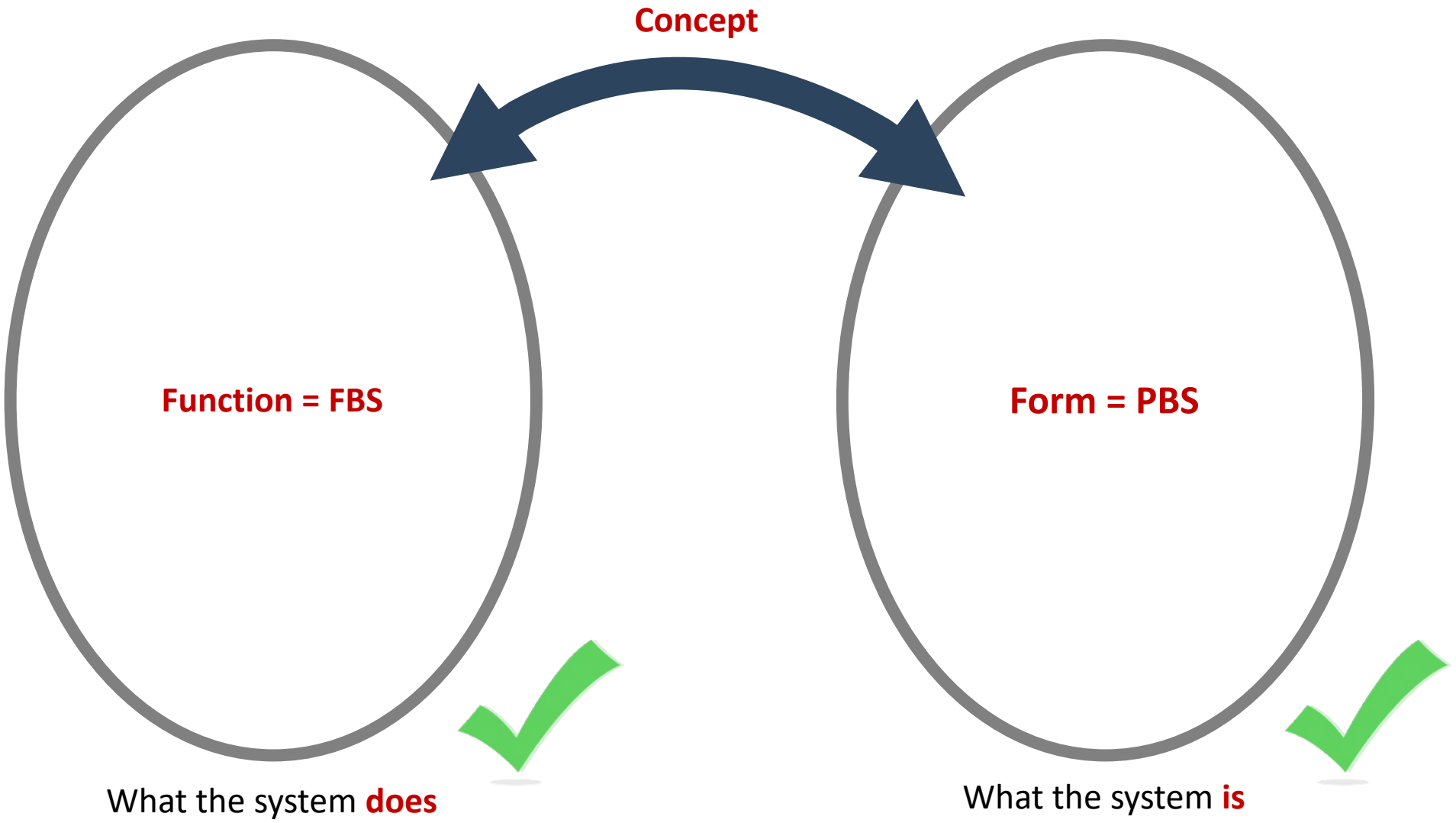



First decision must be robust (if not, all design should be reconsidered).

Thus, the first decisions must be taken based on intengible contraints.

The more volatile decisions will arise the later possible.

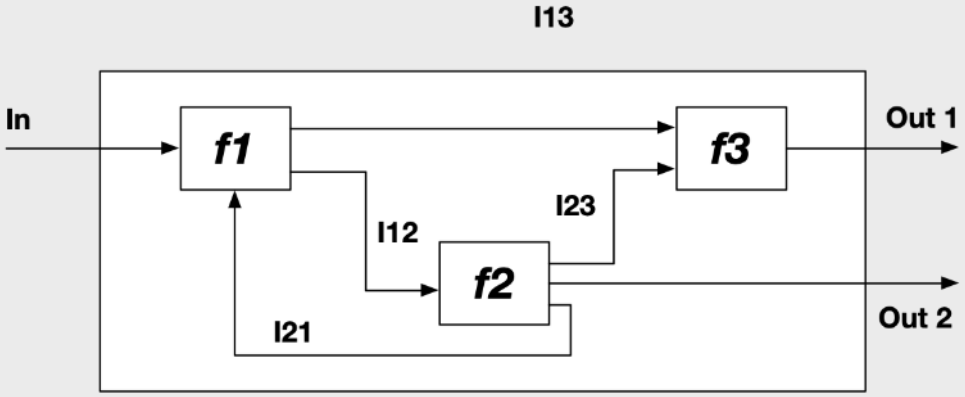
# Architecture in detailed view



# In practice: the allocation of functions into forms

# Function & Product Breakdown Structures MUST be threaded in parallel!

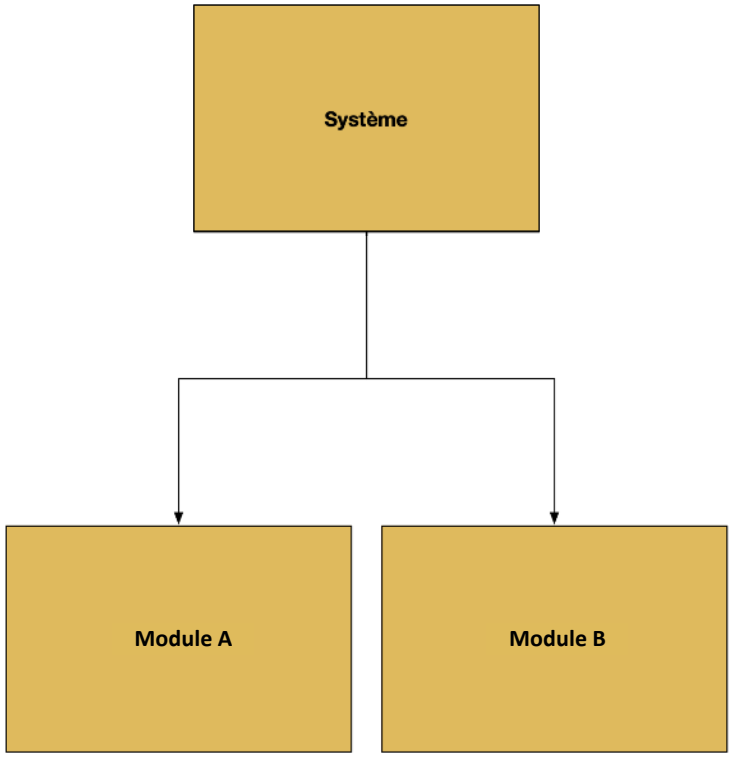
## FUNCTIONS


**Several layers of analysis**

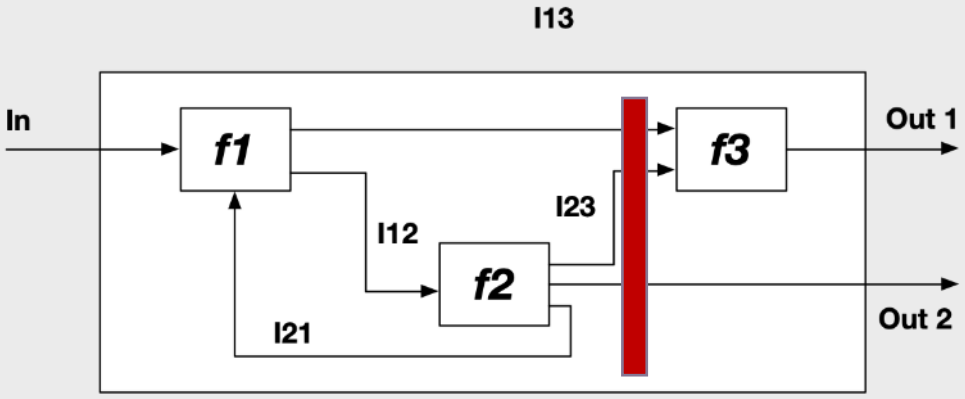
&

## PRODUCT



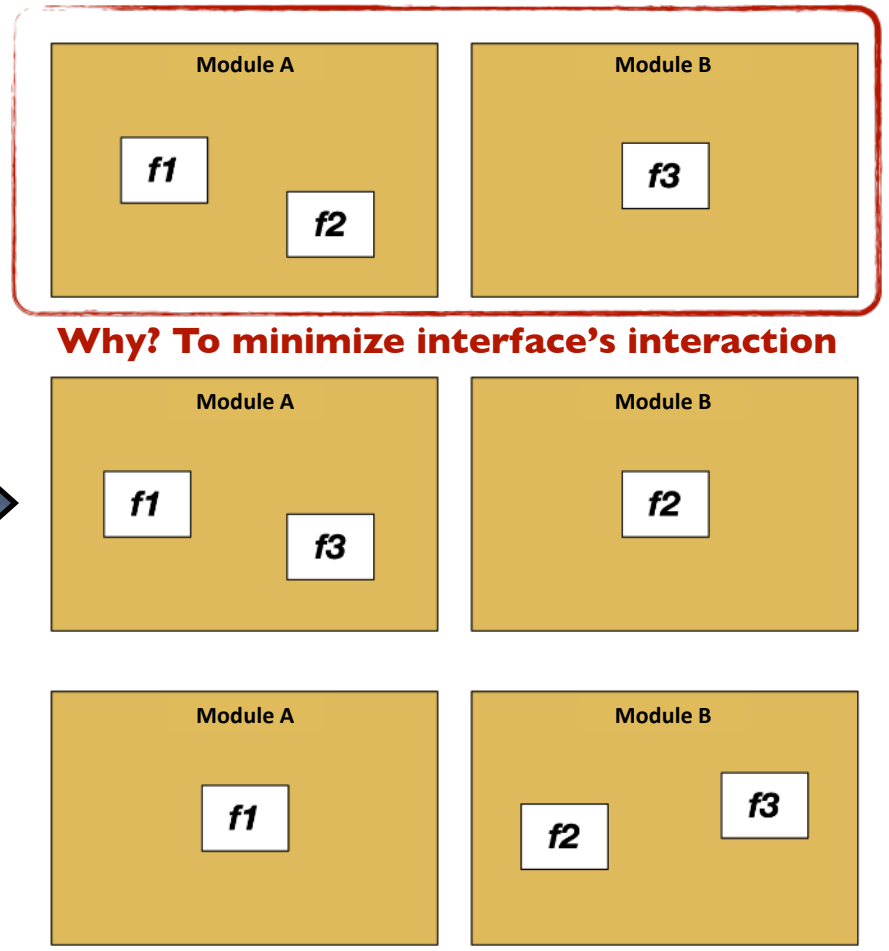
# At the end, Functions are allocated to Form

FUNCTIONS

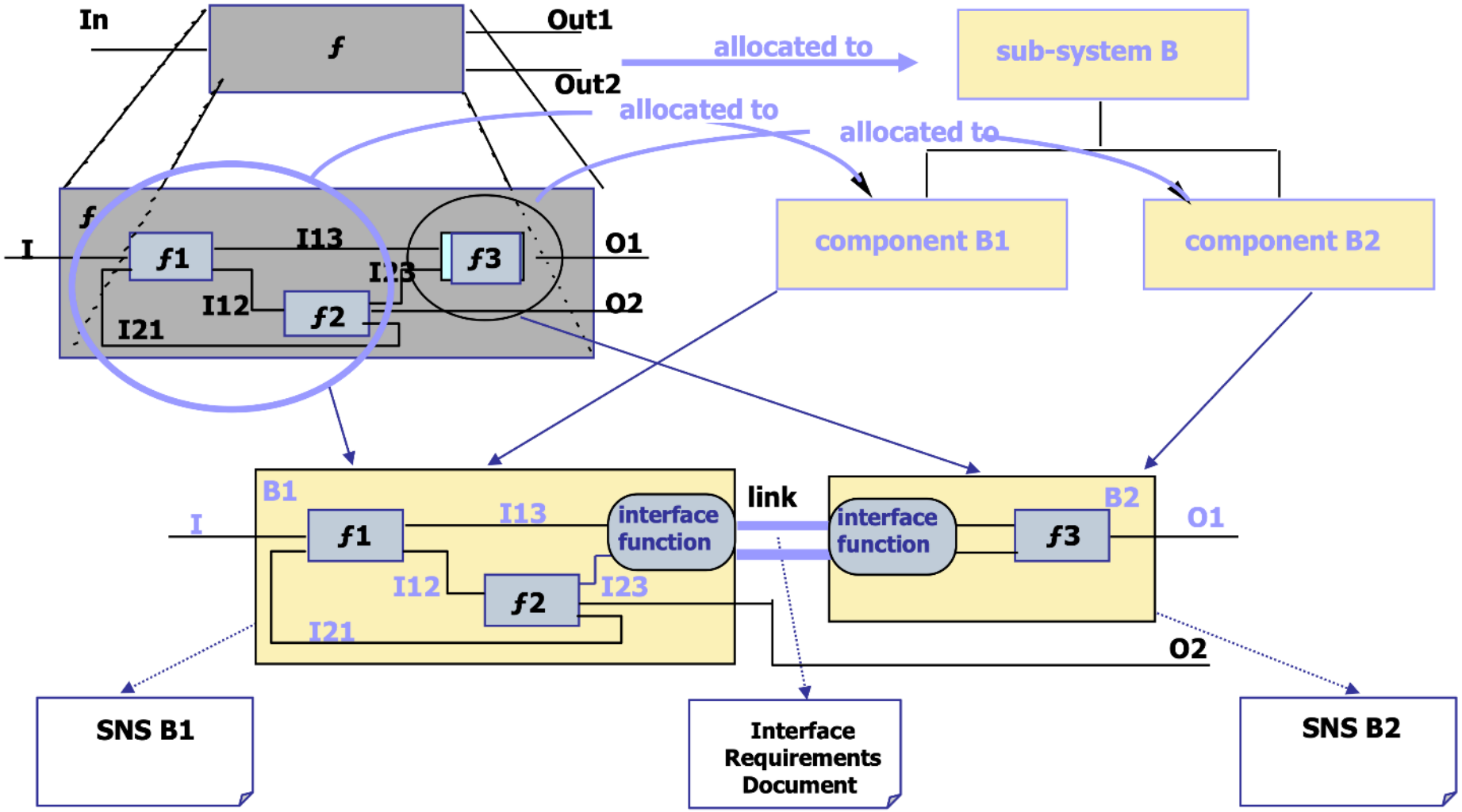




Which allocation?

PRODUCT

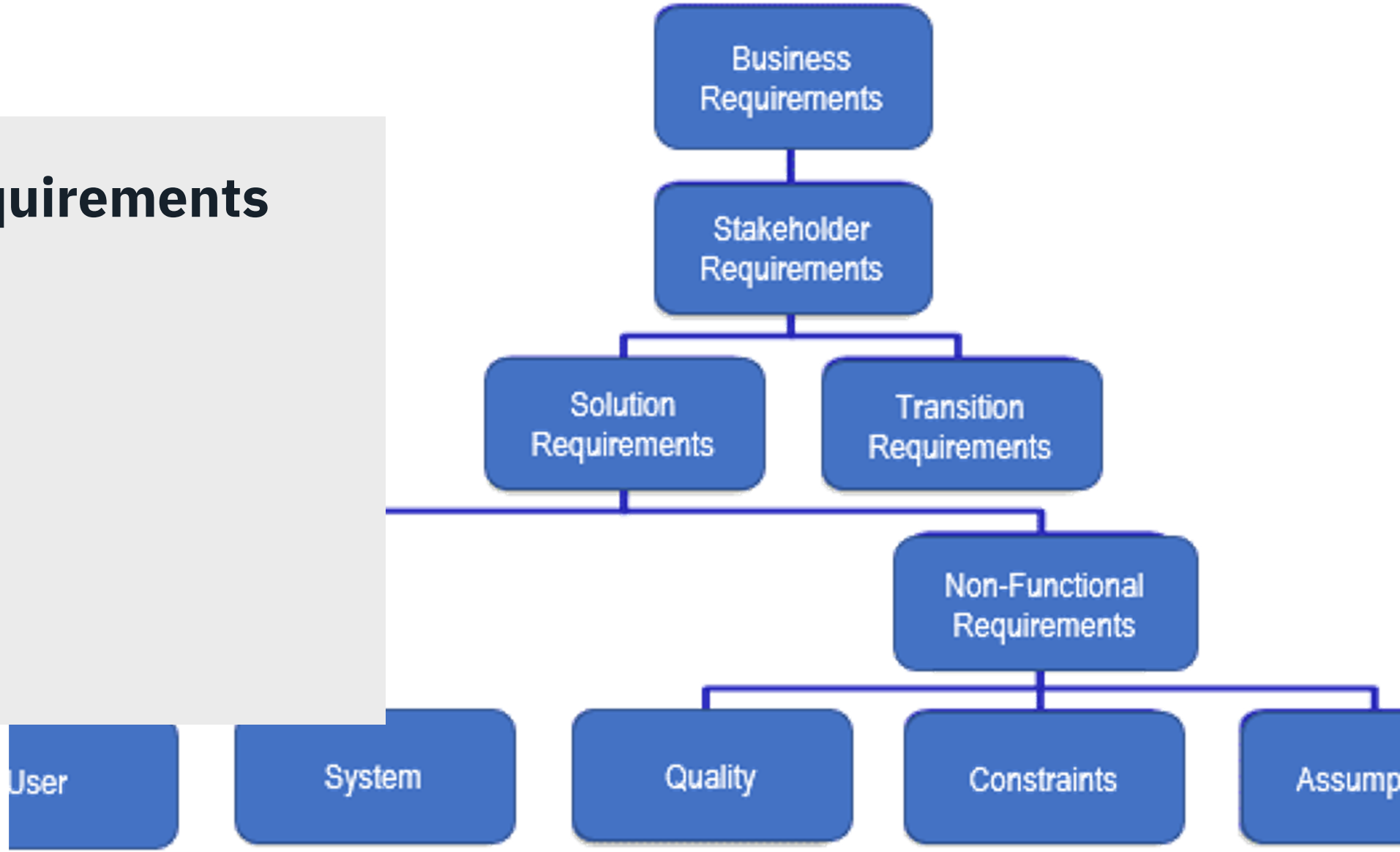


# This work is the heart of architecting of systems

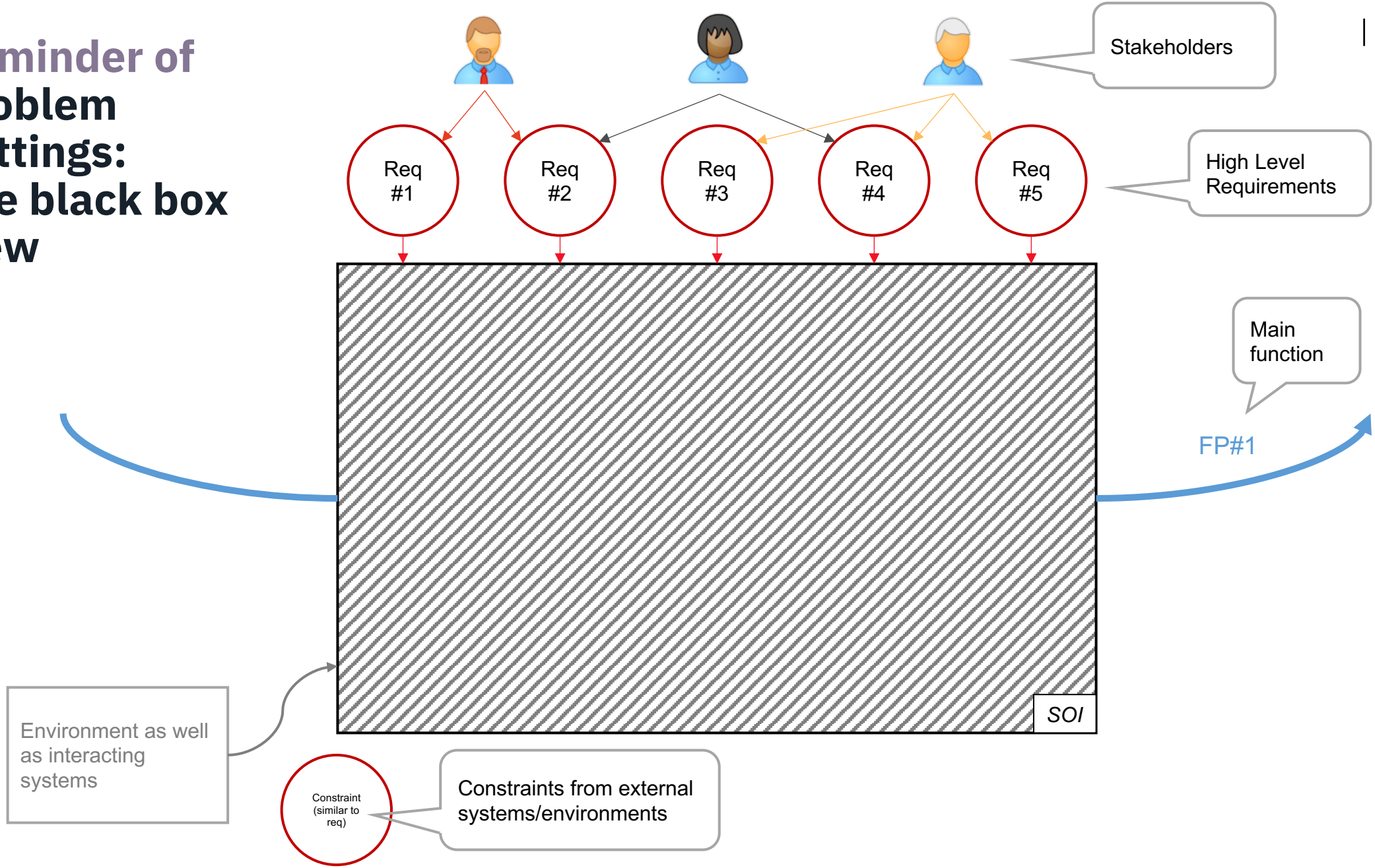


# 02

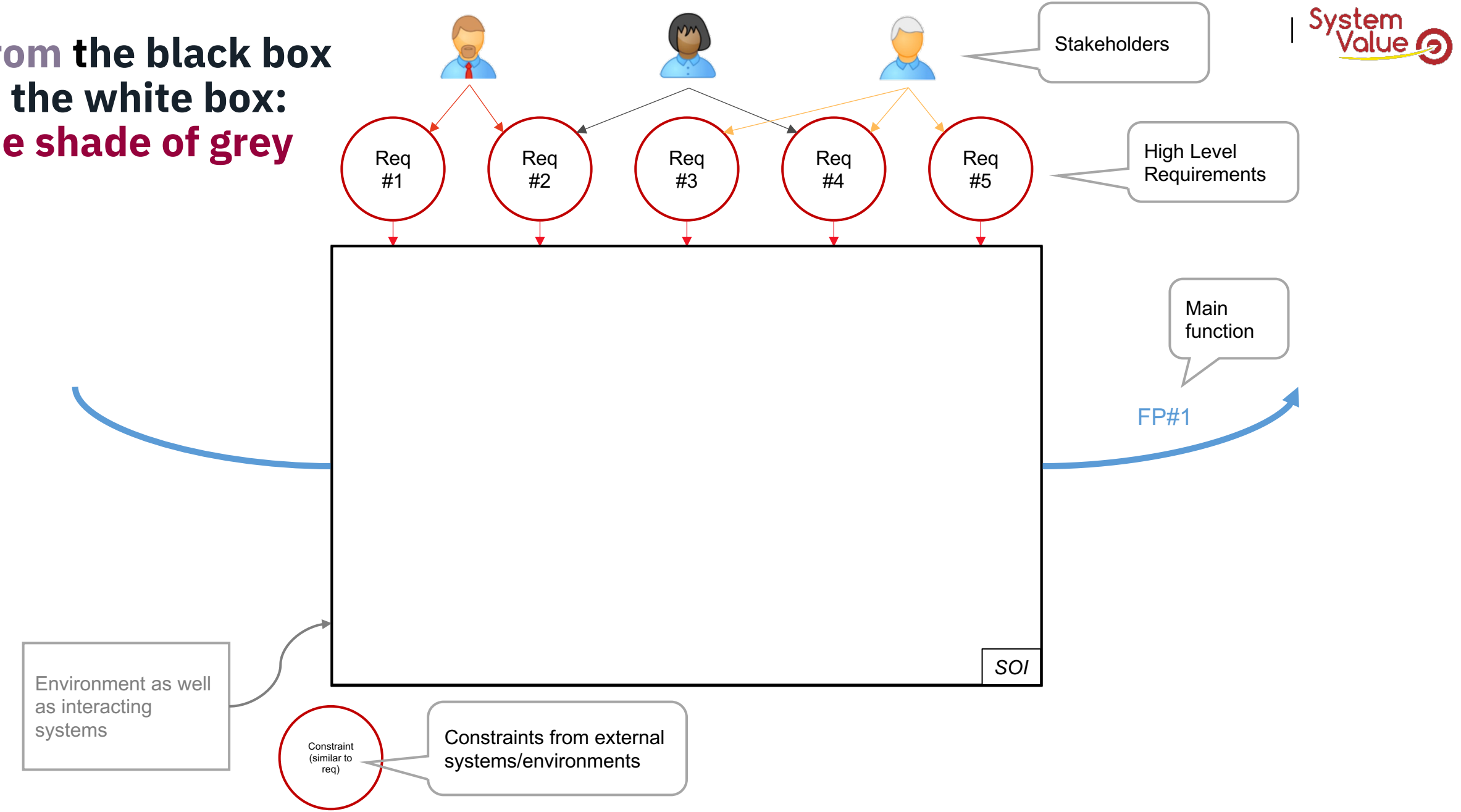
## What about requirements (RBS)?



# Reminder of Problem Settings: The black box view

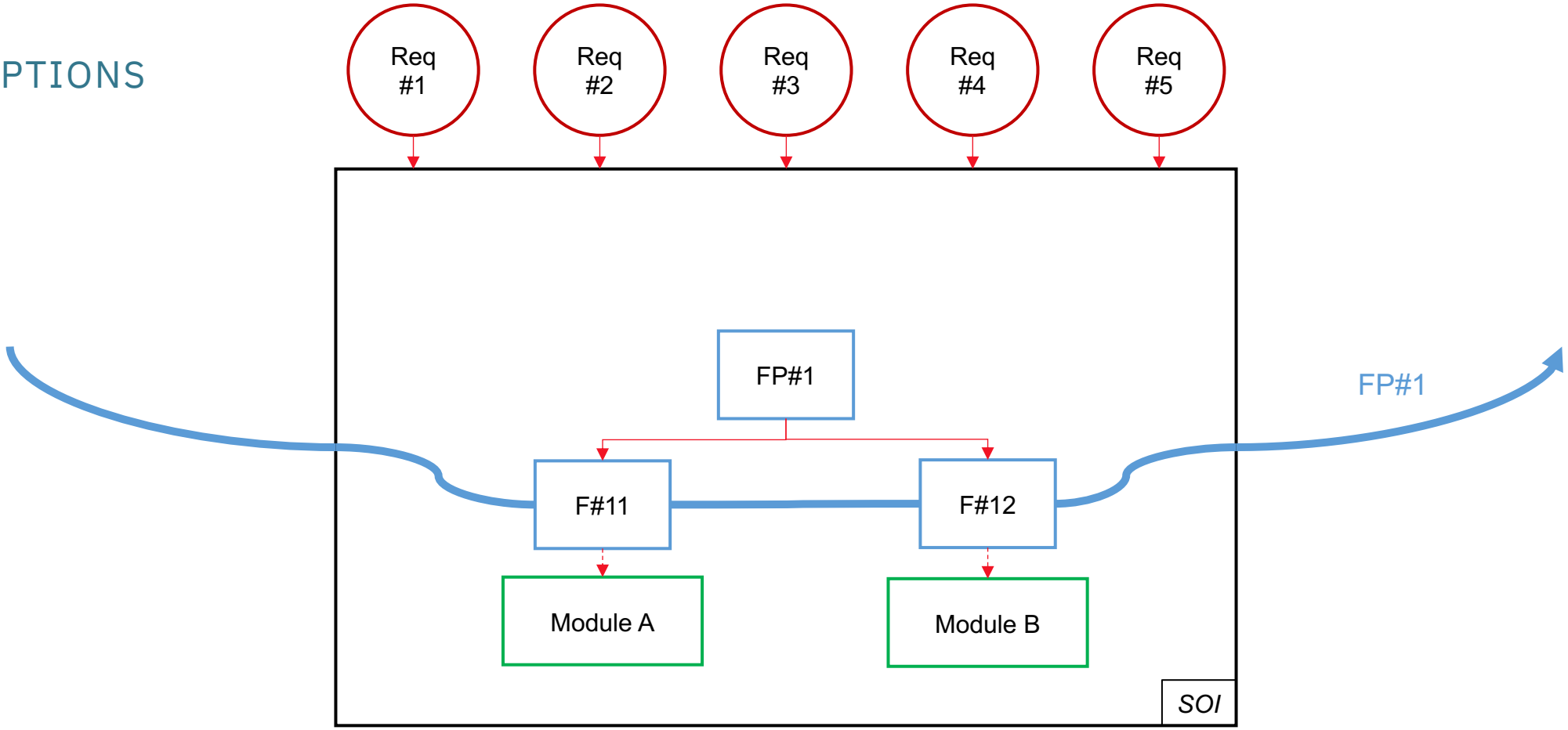


# From the black box to the white box: the shade of grey



# From the black box to the white box: the shade of grey

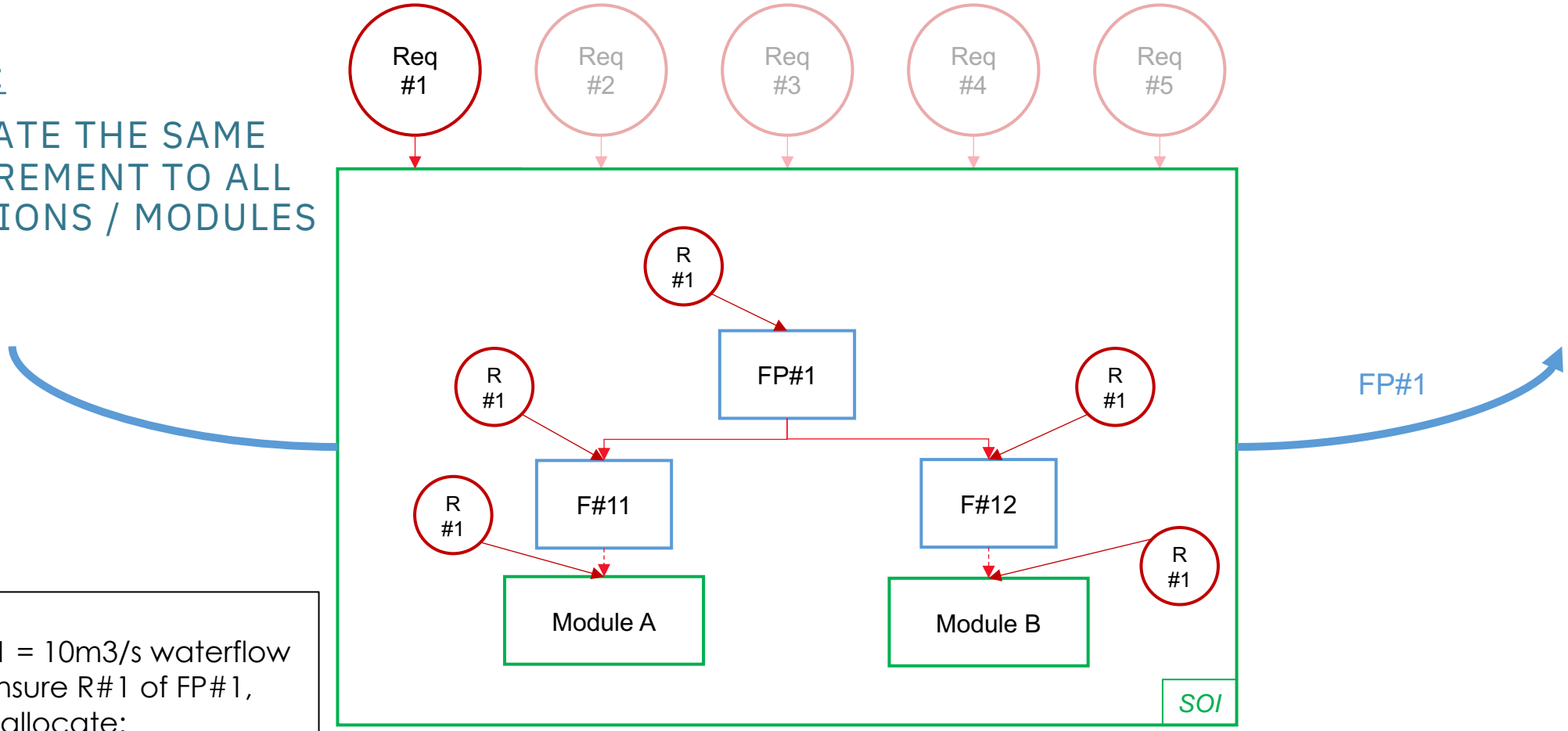
## ASSUMPTIONS



# From the black box to the white box: 5 principles of RBS

TYPE I:

ALLOCATE THE SAME REQUIREMENT TO ALL FUNCTIONS / MODULES



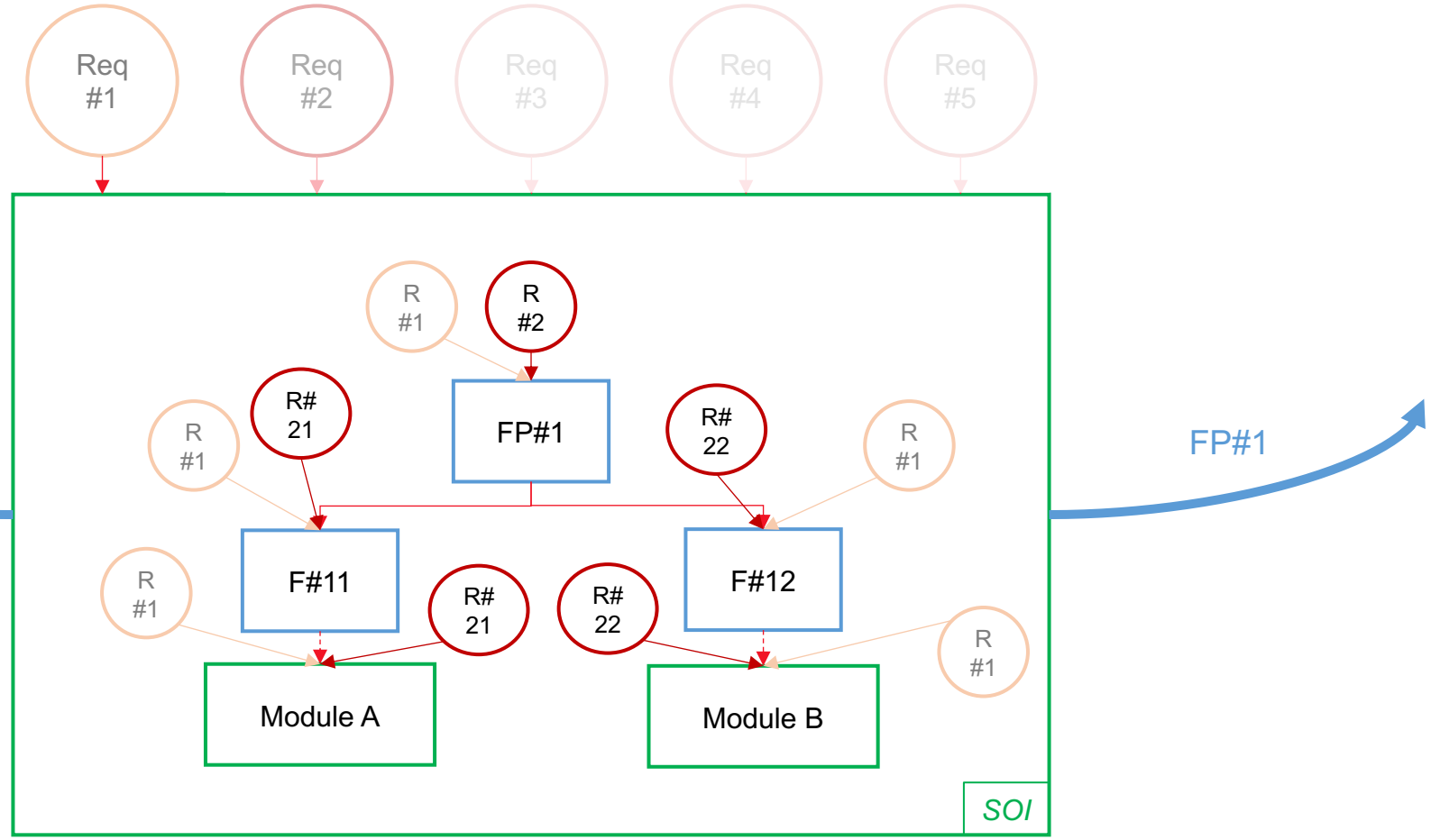

*Example:*

- ✓ Req#1 = 10m3/s waterflow
  - To insure R#1 of FP#1, we allocate:
    - R#1 of 10m3/s to both F#11 & F#12

# From the black box to the white box: 5 principles of RBS

TYPE II:

BREAKDOWN REQUIREMENT INTO DIFFERENT INTERVALS AND ALLOCATE THEM TO FUNCTIONS / MODULES




*Example:*

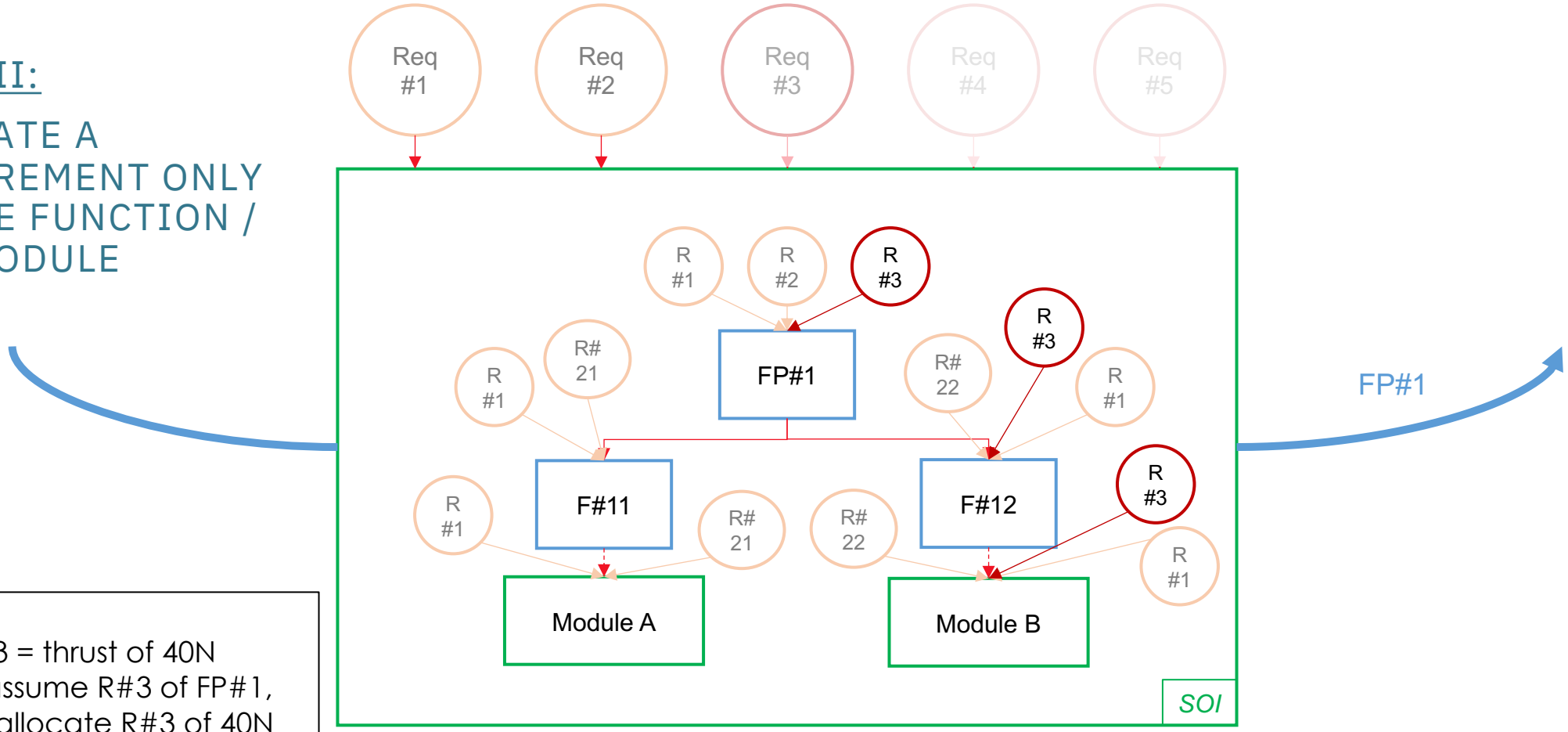
- ✓ Req#2 = Part cost of 10€
  To assume R#2 of FP#1, we allocate R#21 of 7€ for F#11 and R#22 of 3€ for F#12

*The question is still: « does my allocation make sens for system perspective regarding discipline margin? »*

# From the black box to the white box: 5 principles of RBS

TYPE III:

ALLOCATE A REQUIREMENT ONLY ON ONE FUNCTION / ONE MODULE



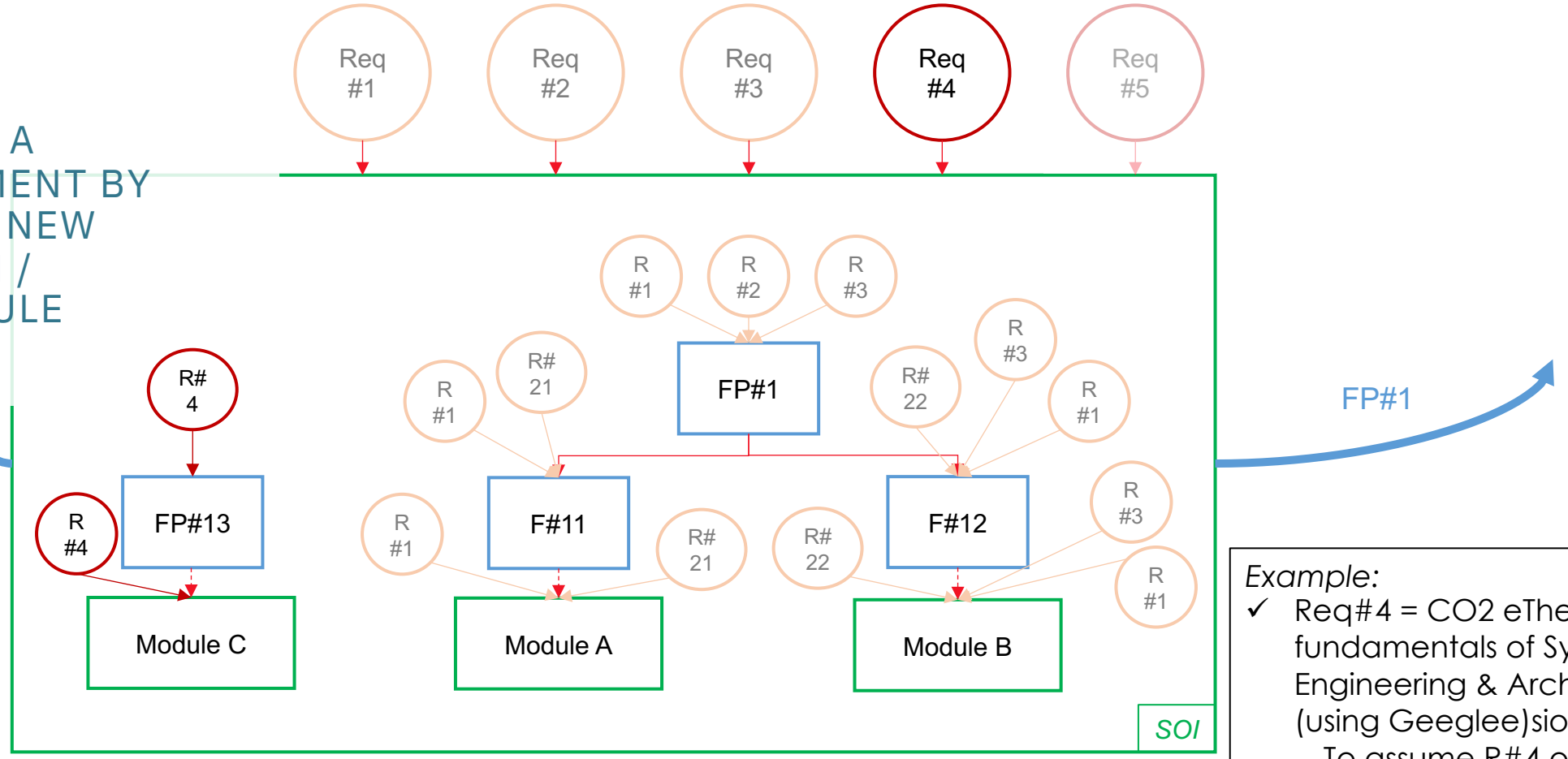

*Example:*

✓ Req#3 = thrust of 40N
To assume R#3 of FP#1, we allocate R#3 of 40N to F#12 (thruster function)

# From the black box to the white box: 5 principles of RBS

TYPE IV:

ALLOCATE A REQUIREMENT BY ADDING A NEW FUNCTION / NEW MODULE



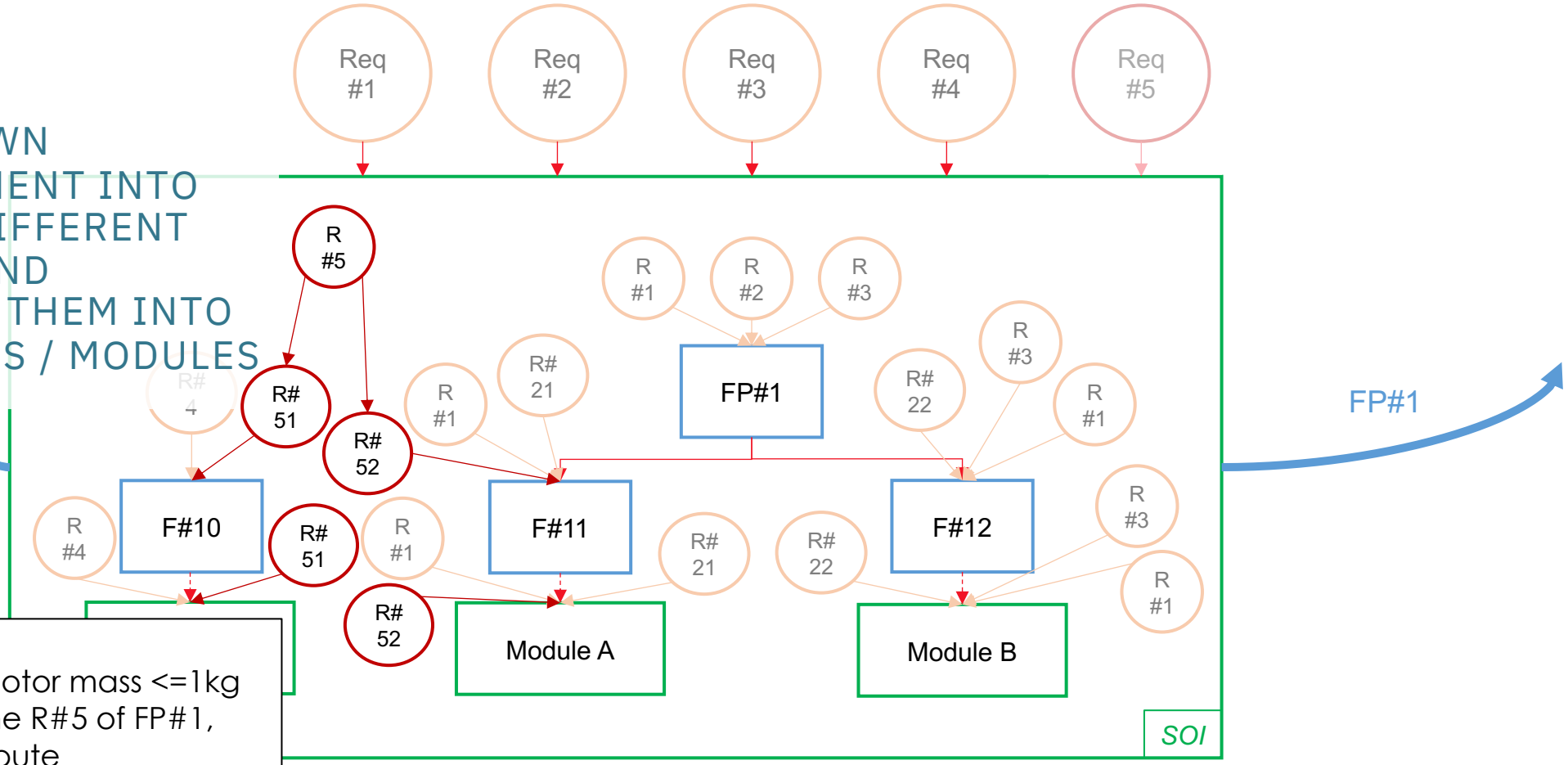

*Example:*

- ✓ Req#4 = CO2 eThe fundamentals of Systems Engineering & Architecting (using Geeglee)sion<230g
  To assume R#4 of FP#1, we create FP#13 and allocate to them R#4 below 230g (FP#13 is the pollutants removal function)

# From the black box to the white box: 5 principles of RBS

TYPE V:

BREAKDOWN REQUIREMENT INTO REQ. OF DIFFERENT NATURE AND ALLOCATE THEM INTO FUNCTIONS / MODULES



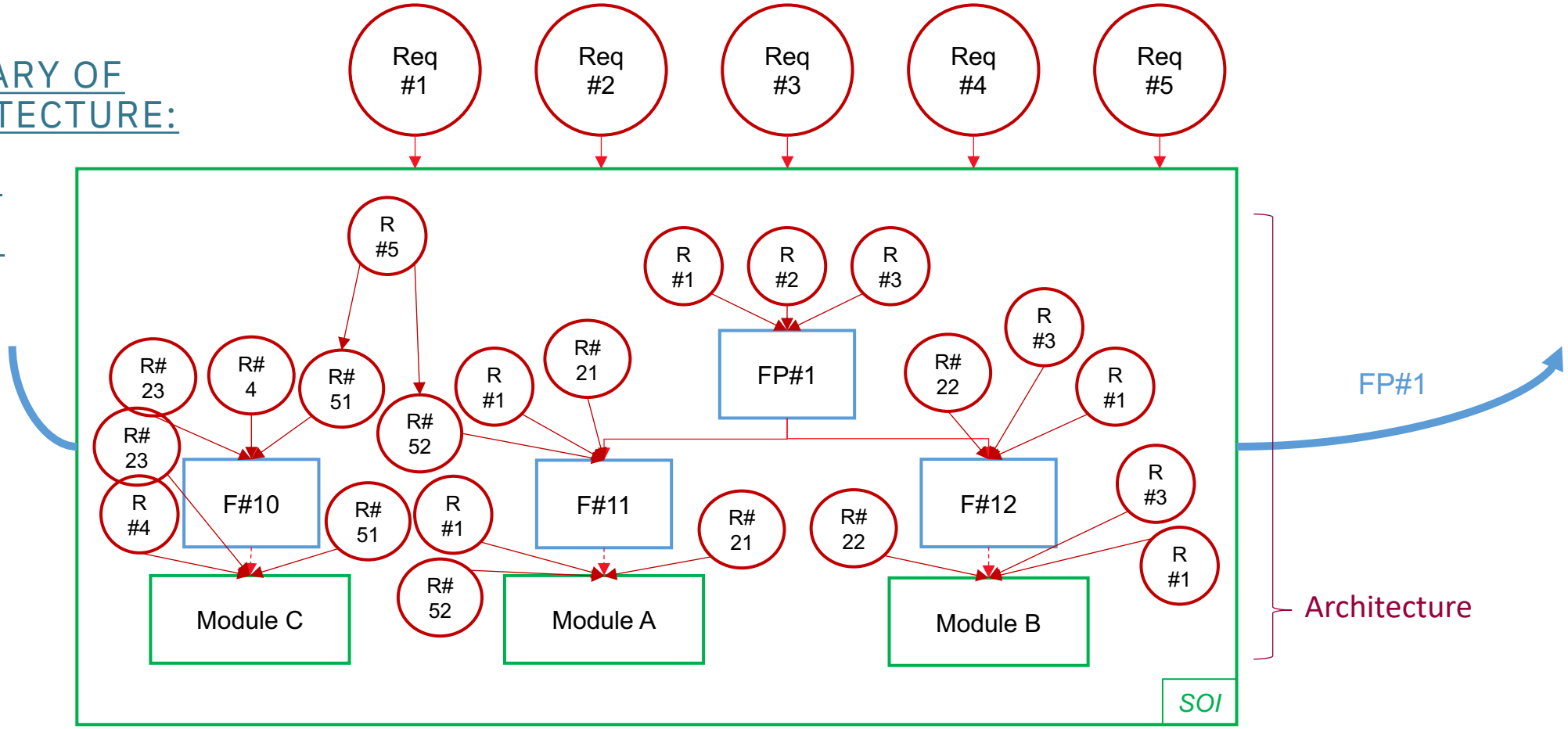

*Example:*

✓ Req#5 = Motor mass <=1kg
To assume R#5 of FP#1, we compute
R#51 <= 4 motors for F#10
& R#52 <= 250g for F#12

# From the black box to the white box: 5 principles of RBS

## SUMMARY OF ARCHITECTURE:

- ✓ FBS,
- ✓ PBS,
- ✓ RBS

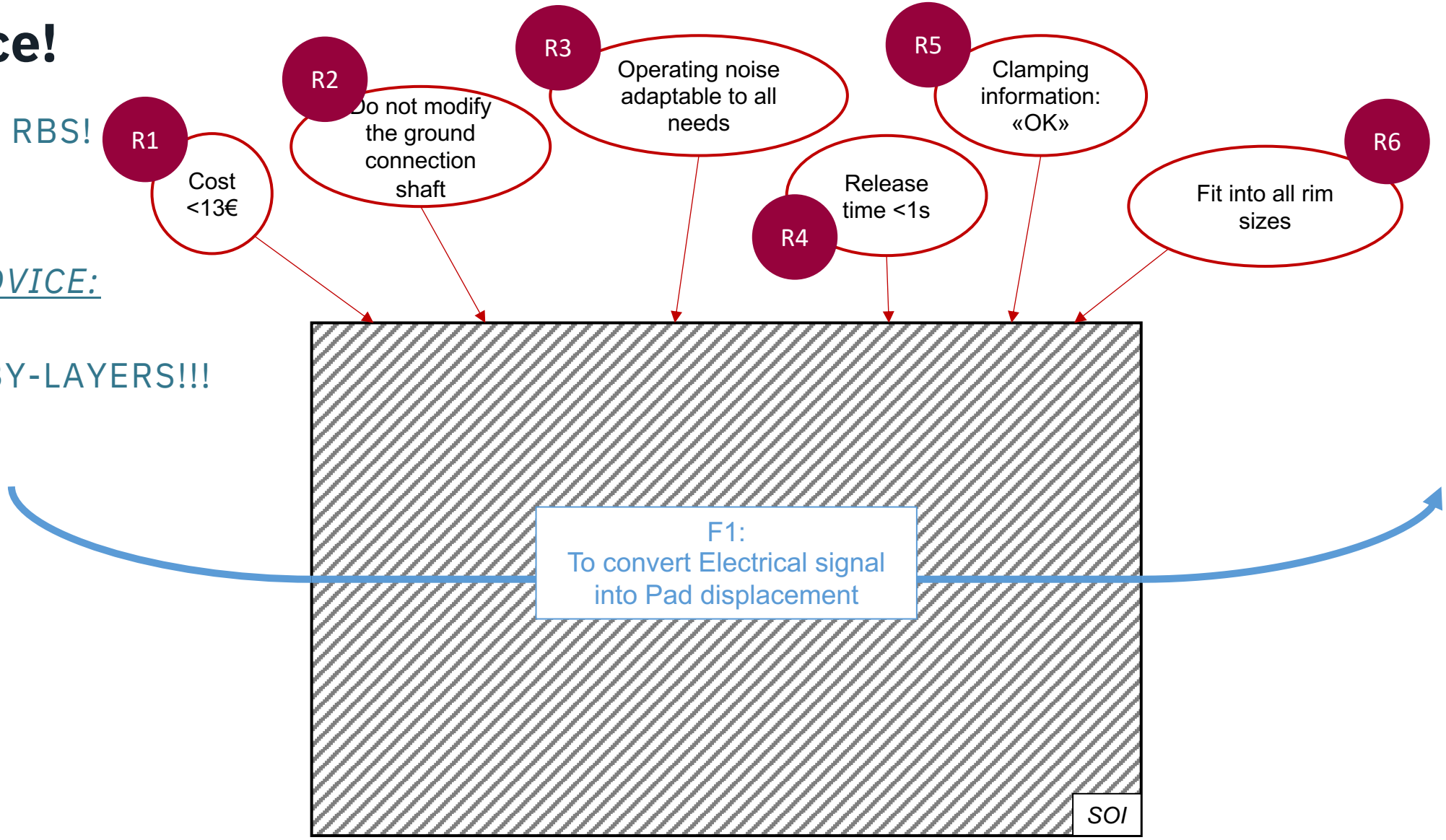


# Practice!

FBS, PBS, RBS!

*LITTLE ADVICE:*

DO IT
LAYERS-BY-LAYERS!!!



# Practice!

FBS, PBS, RBS!

R1 Cost <13€

R2 Do not modify the ground connection shaft

R3 Operating noise adaptable to all needs

R4 Release time <1s

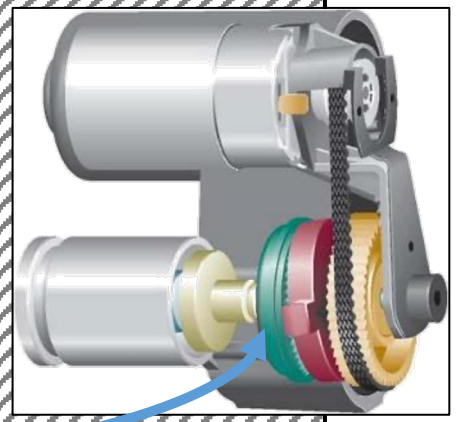
R5 Clamping information: «OK»

R6 Fit into all rim sizes

F1: Convertir un Signal Electrique en mouvement de déplacement mécanique

F11: Convertir un Signal Electrique en couple mécanique

F12: Convertir un couple mécanique en mouvement de déplacement mécanique



Interface CM

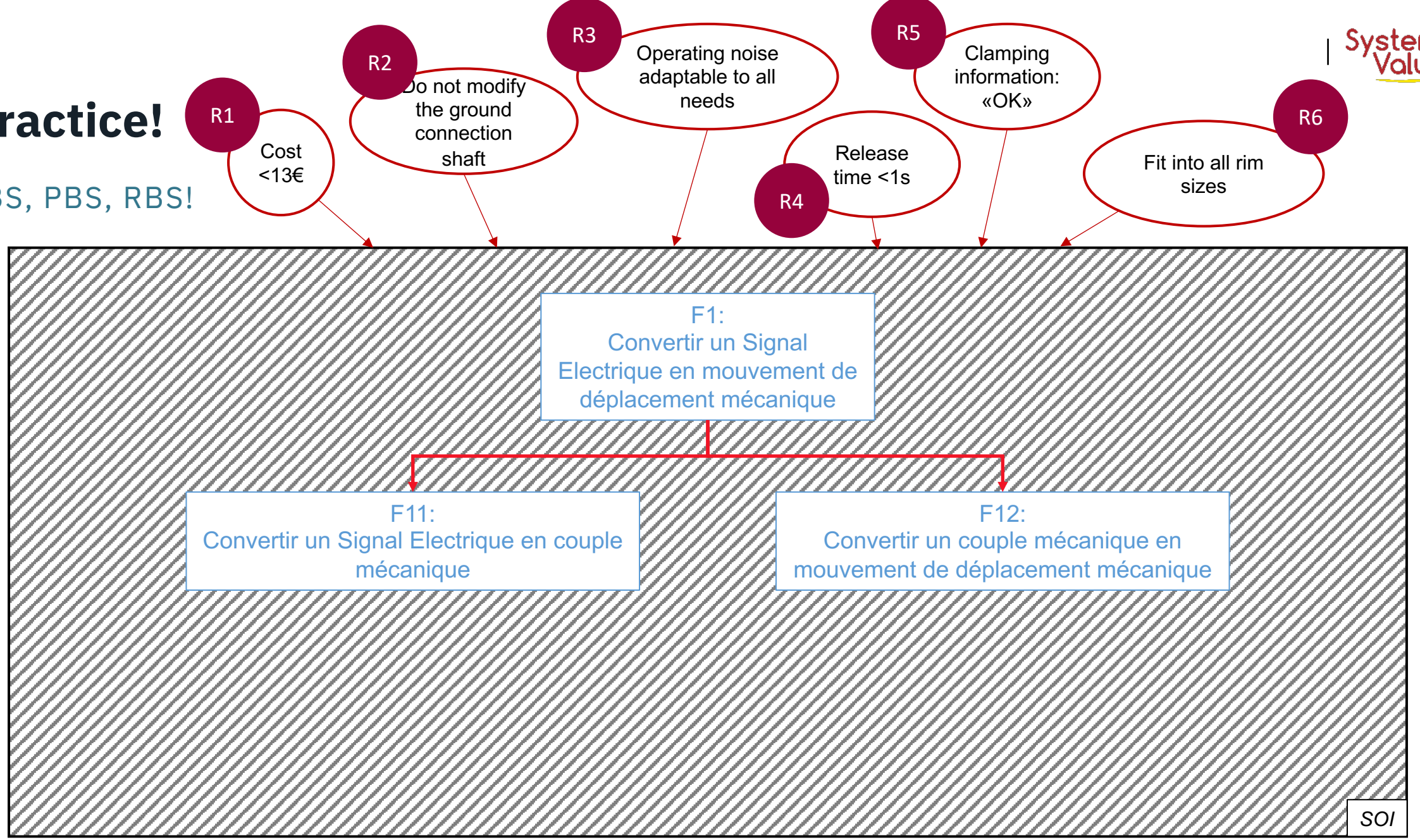
Other way to inteface?

Post-it session:
- Hydraulic
- Pneumatic
- Magnetic…

All potential architecture candidate

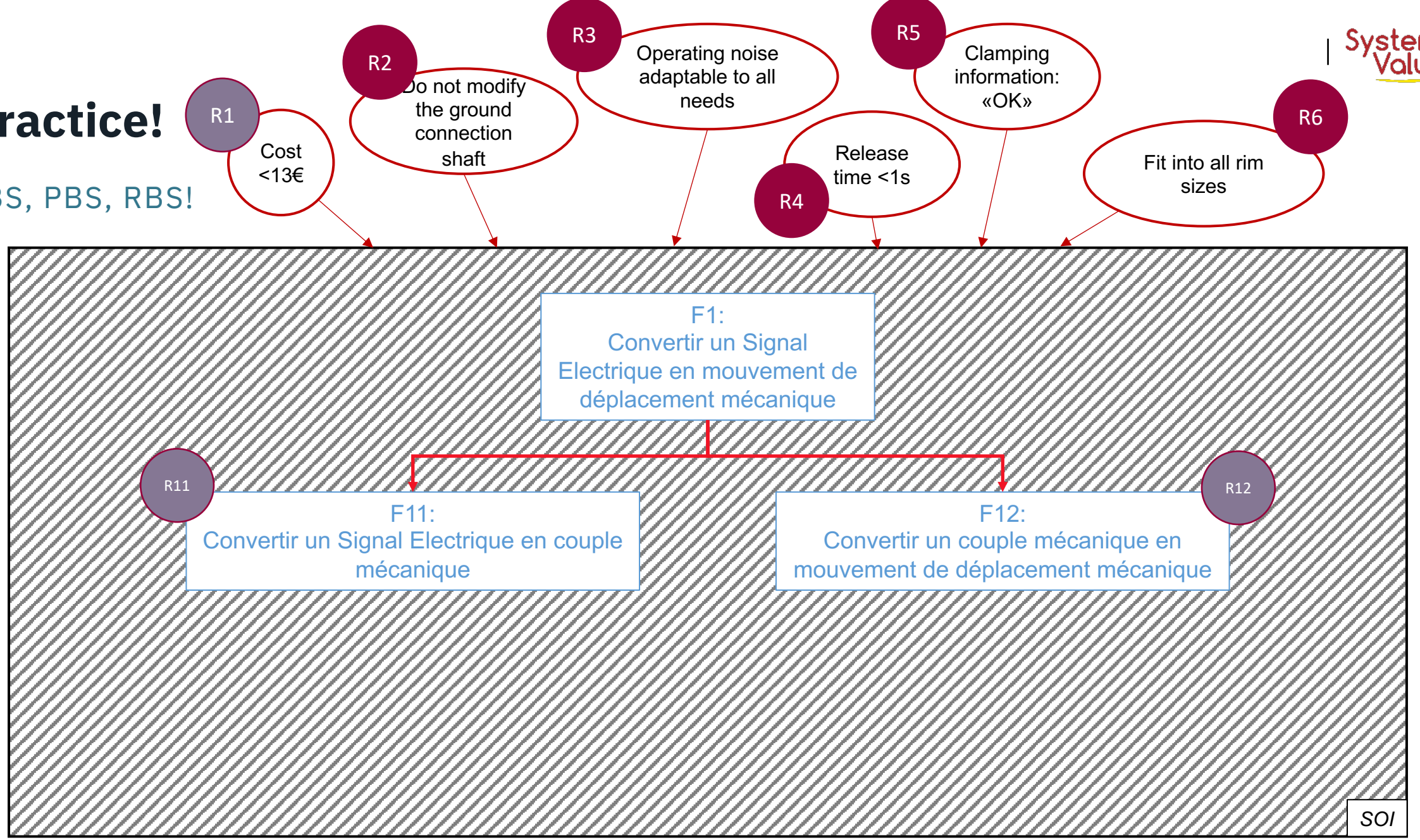
# Practice!

FBS, PBS, RBS!



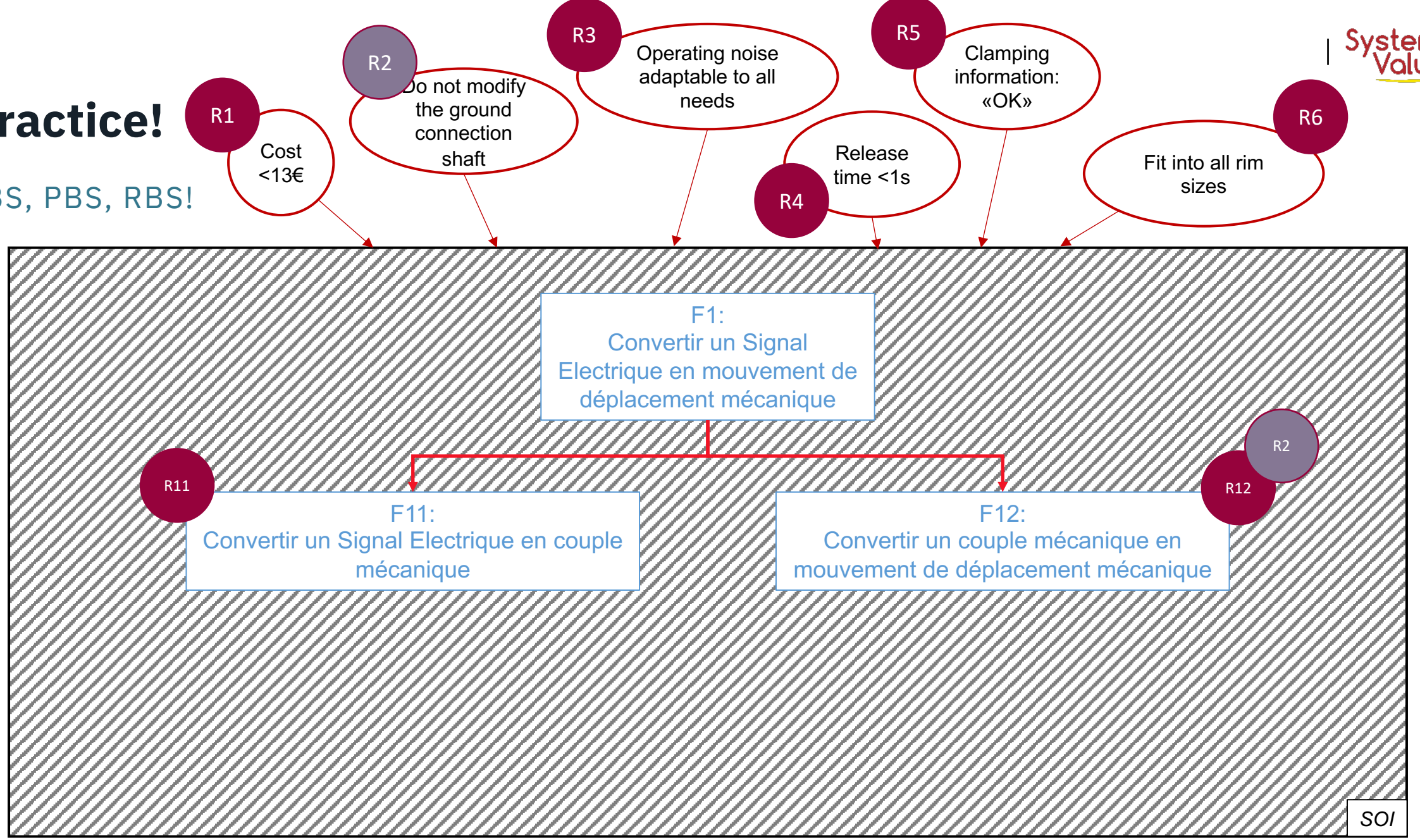
# Practice!

FBS, PBS, RBS!



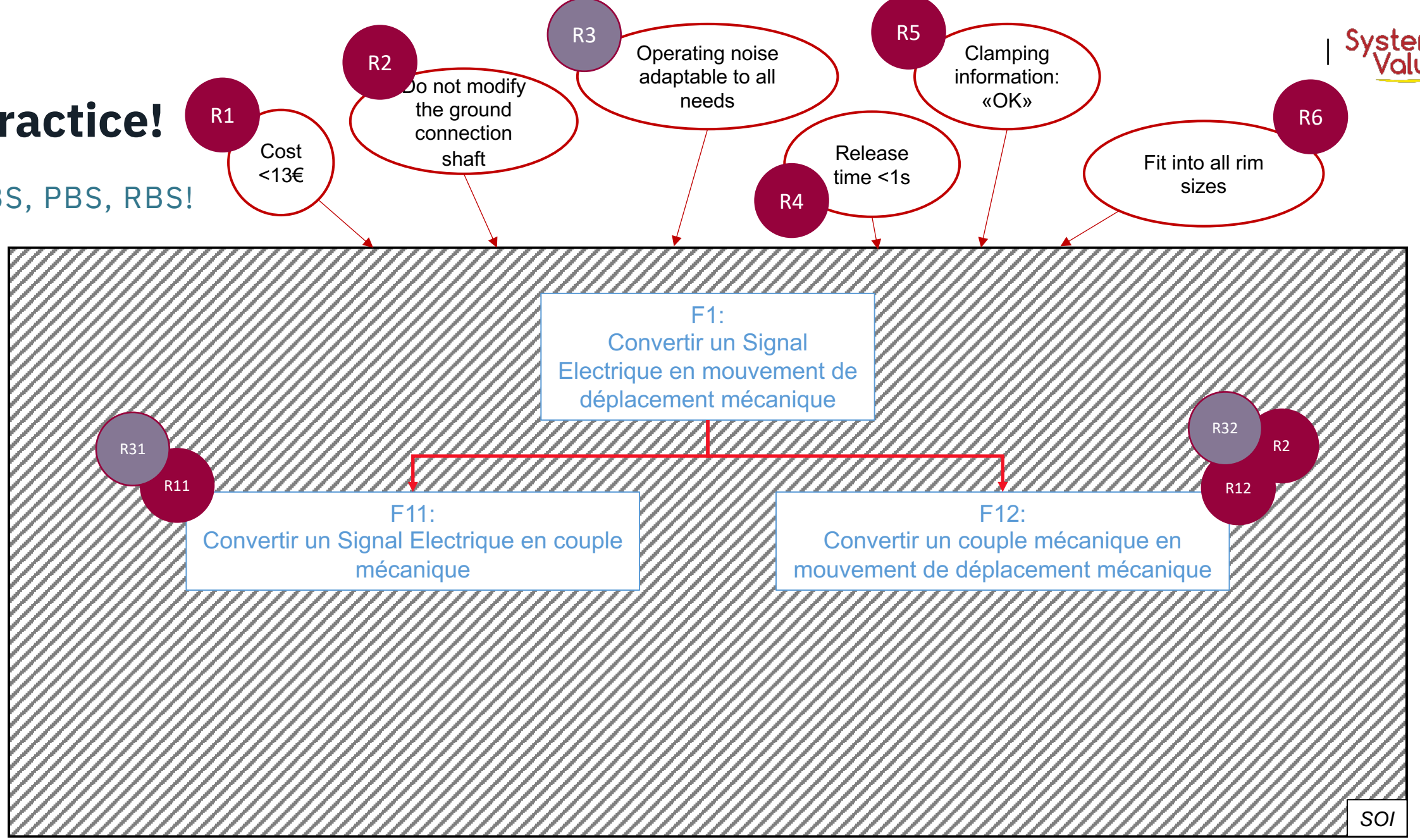
# Practice!

FBS, PBS, RBS!



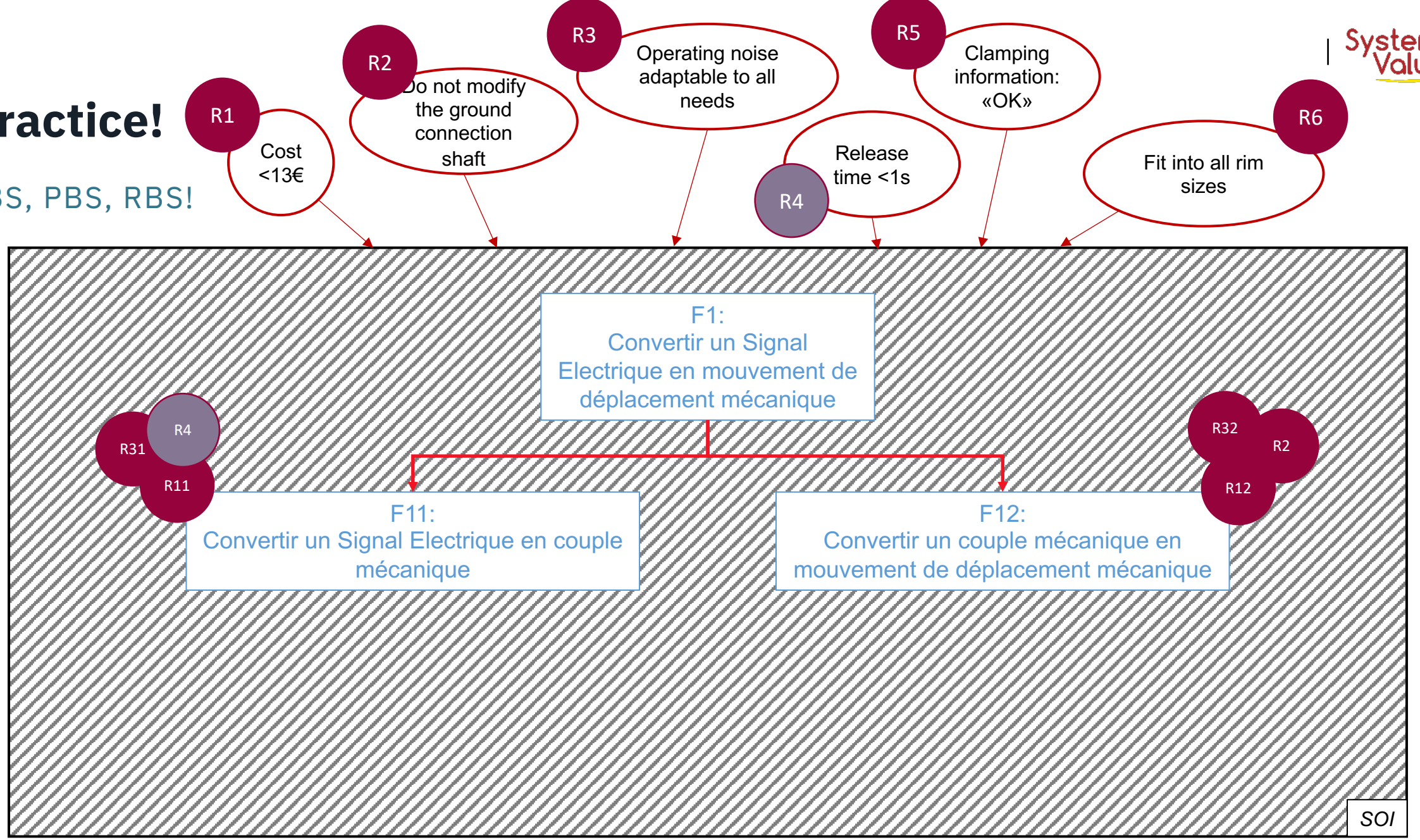
# Practice!

FBS, PBS, RBS!



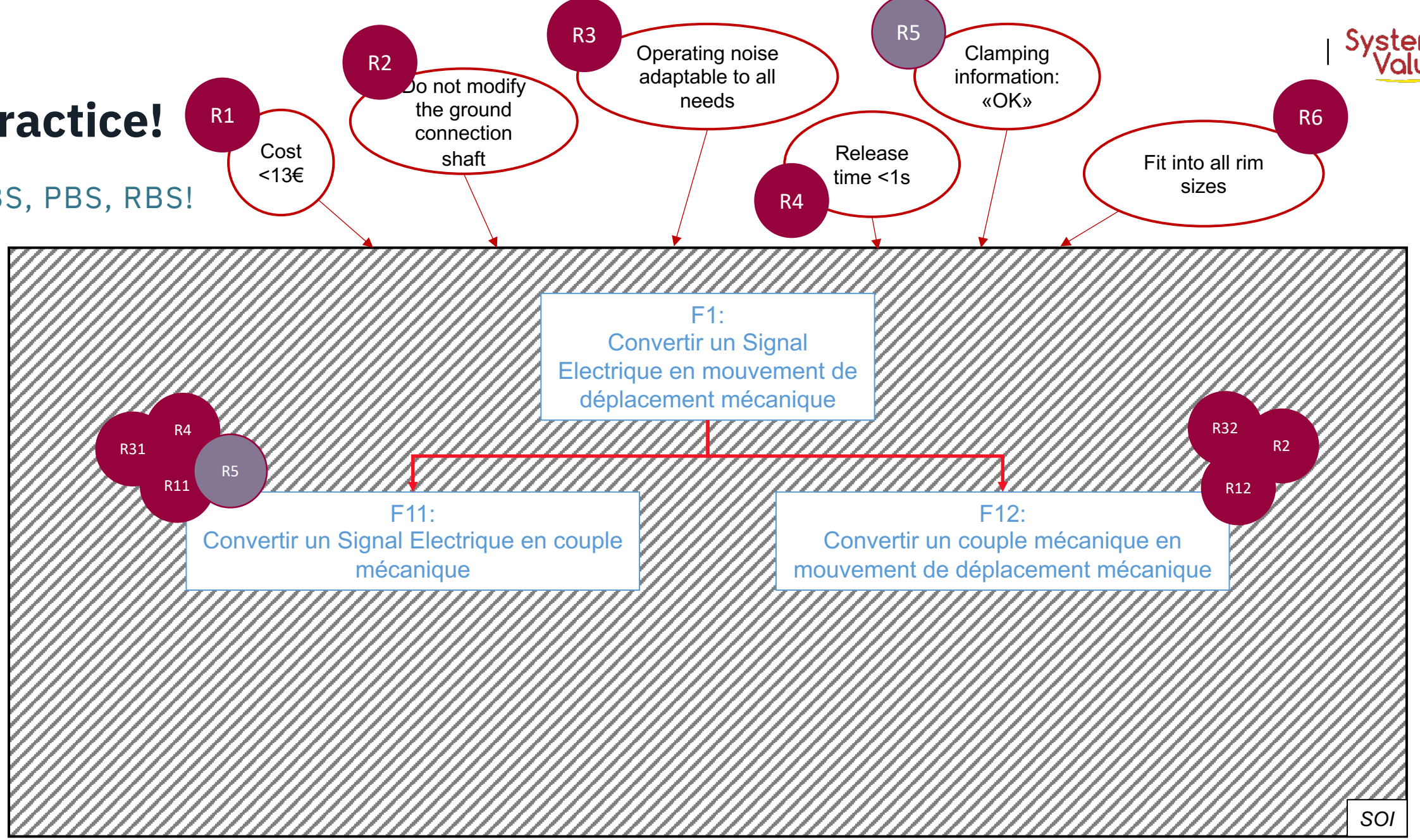
# Practice!

FBS, PBS, RBS!



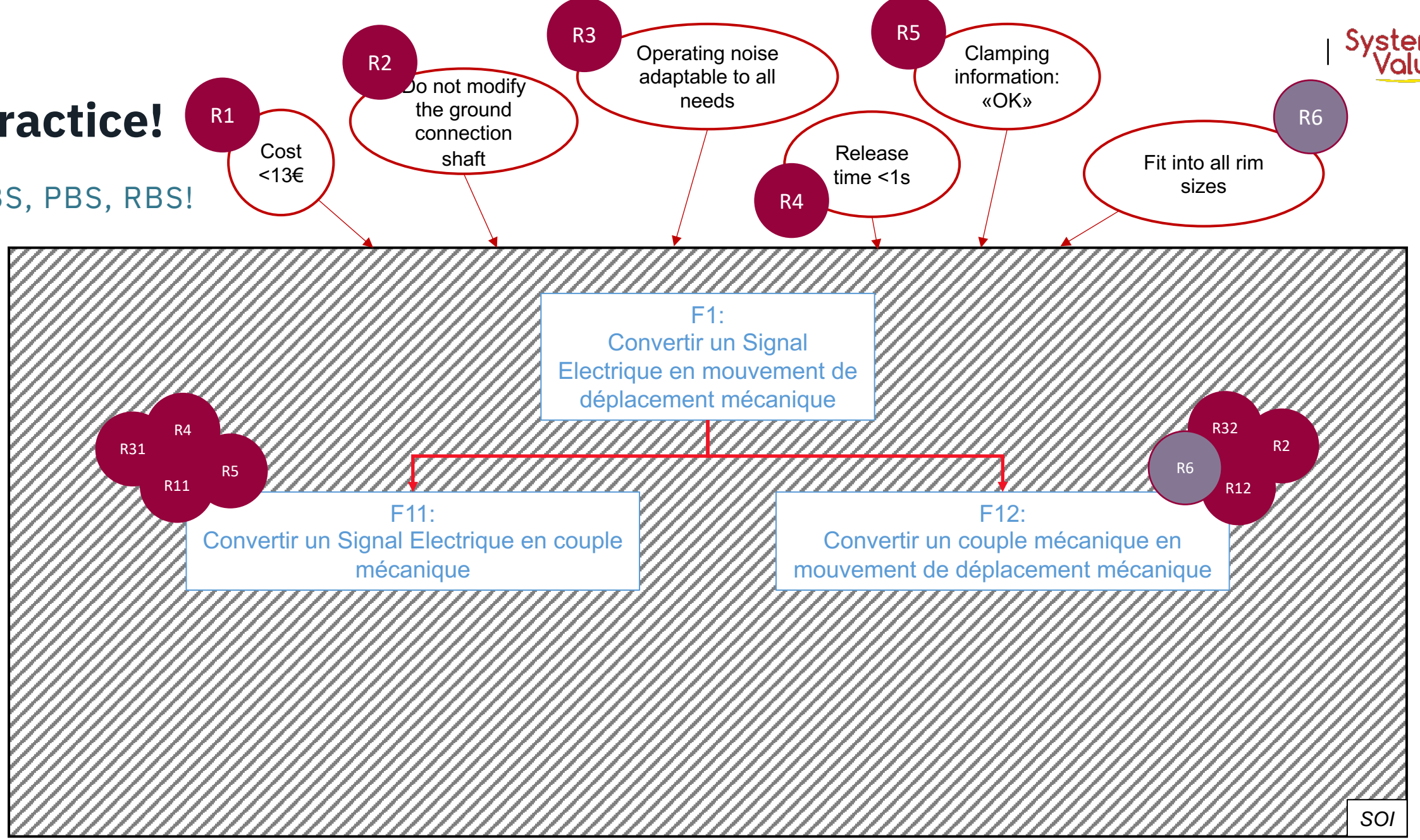
# Practice!

FBS, PBS, RBS!



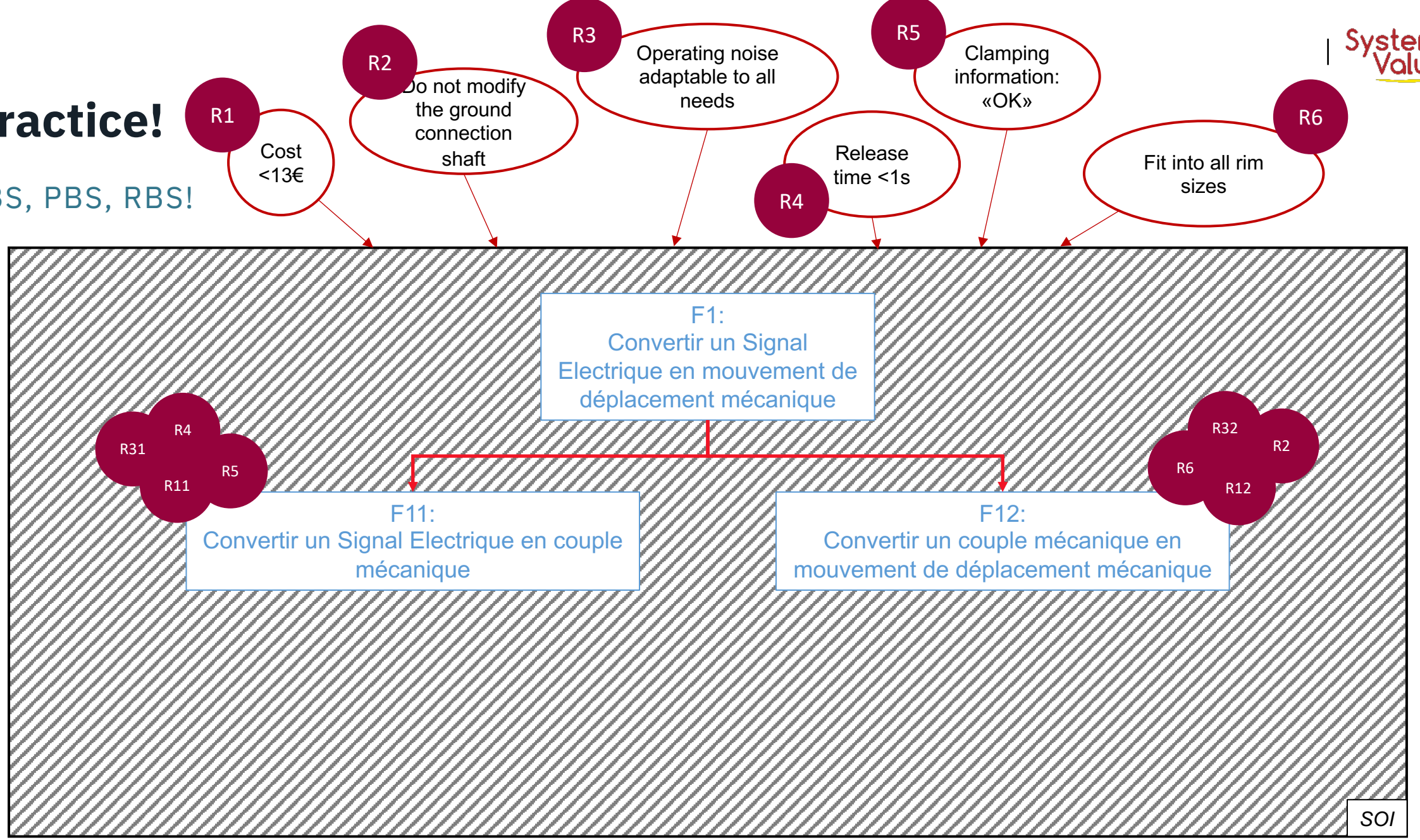
# Practice!

FBS, PBS, RBS!



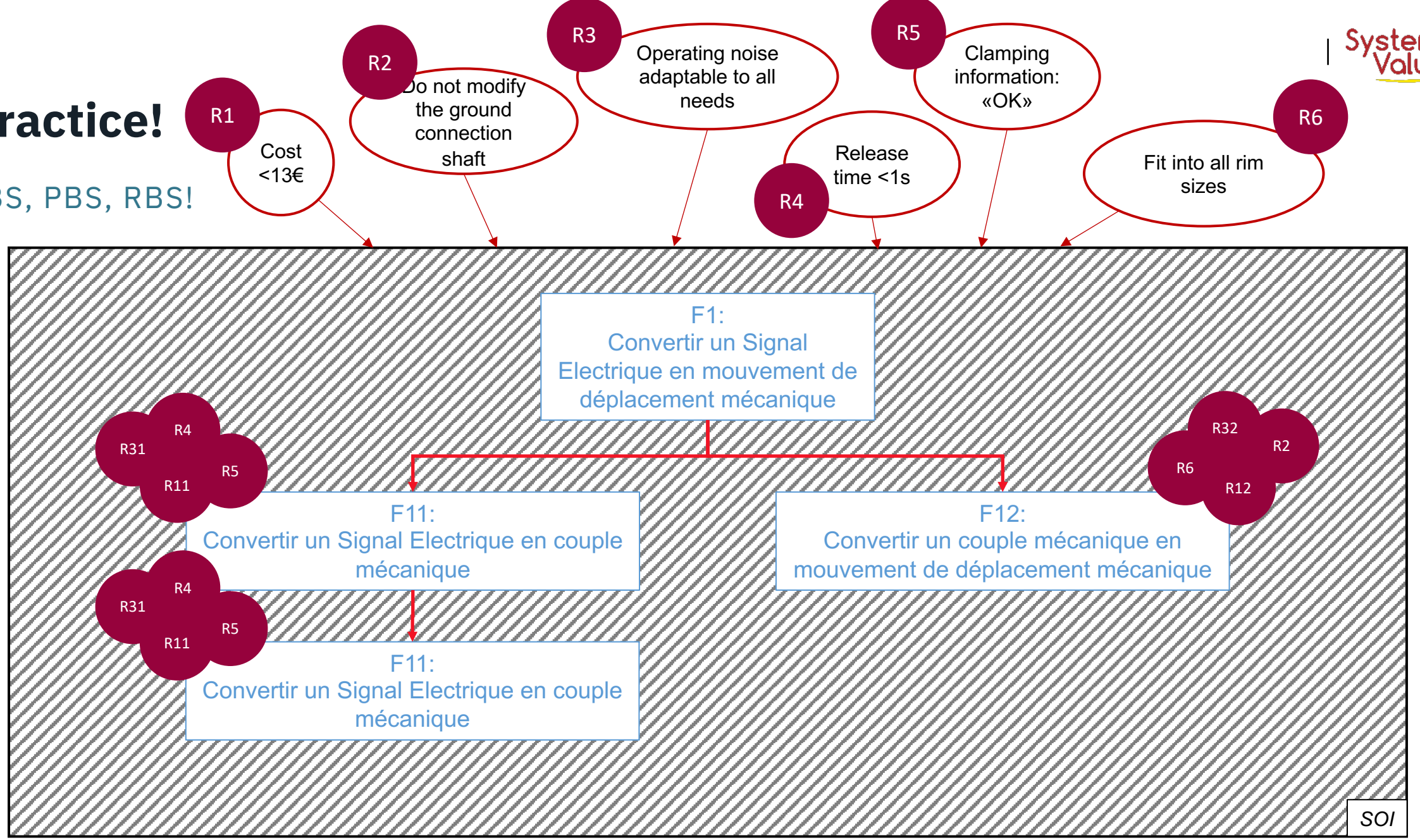
# Practice!

FBS, PBS, RBS!



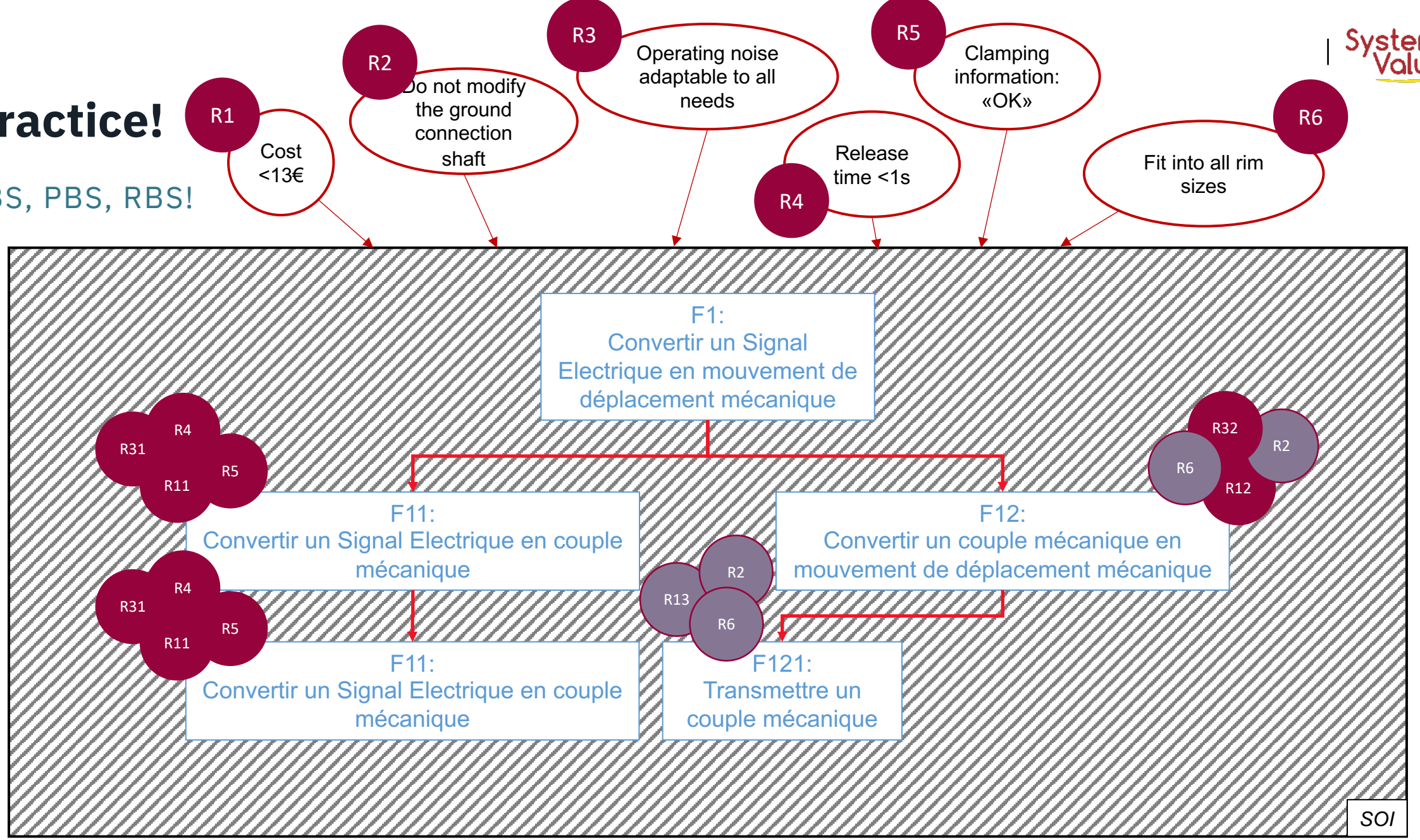
# Practice!

FBS, PBS, RBS!



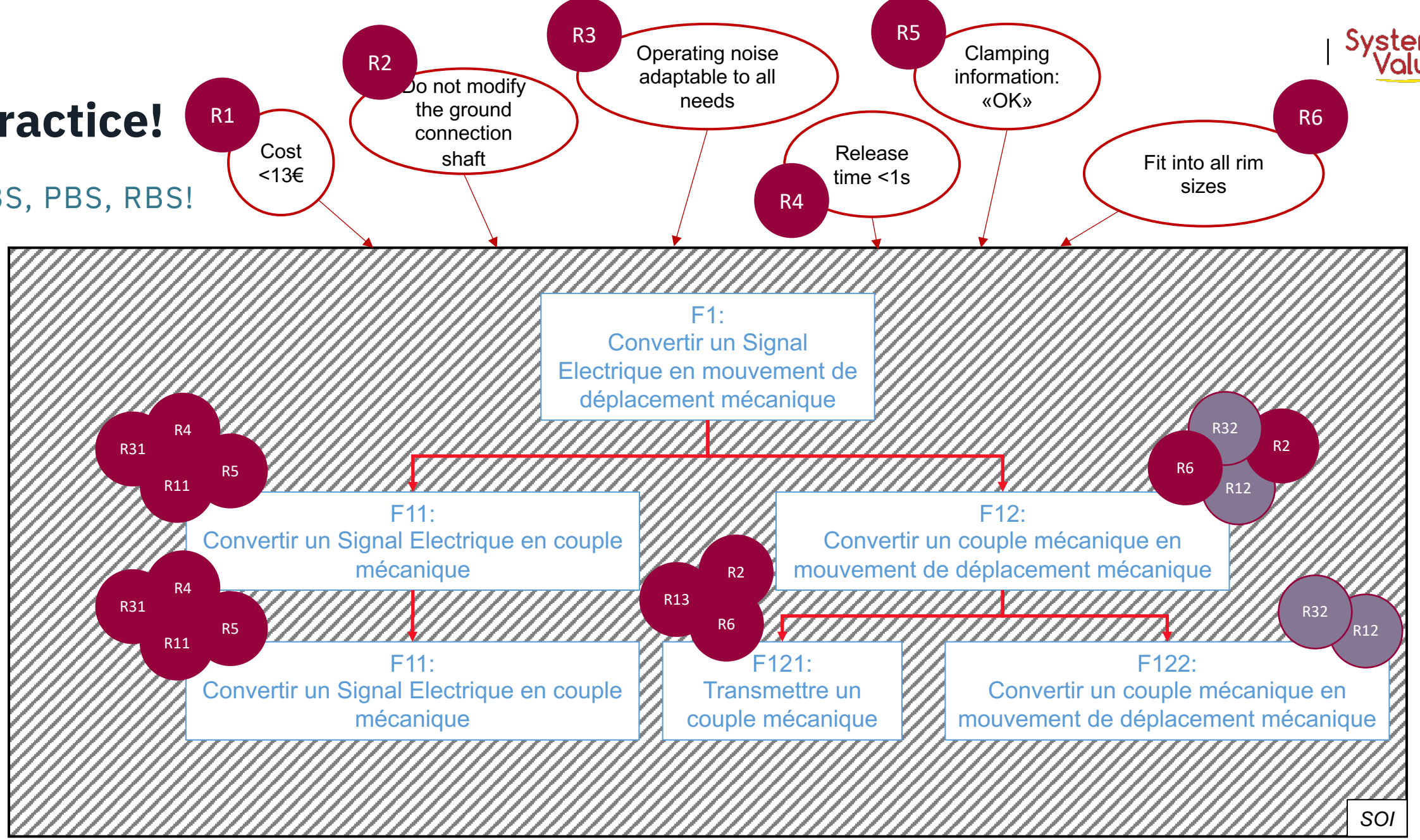
# Practice!

FBS, PBS, RBS!



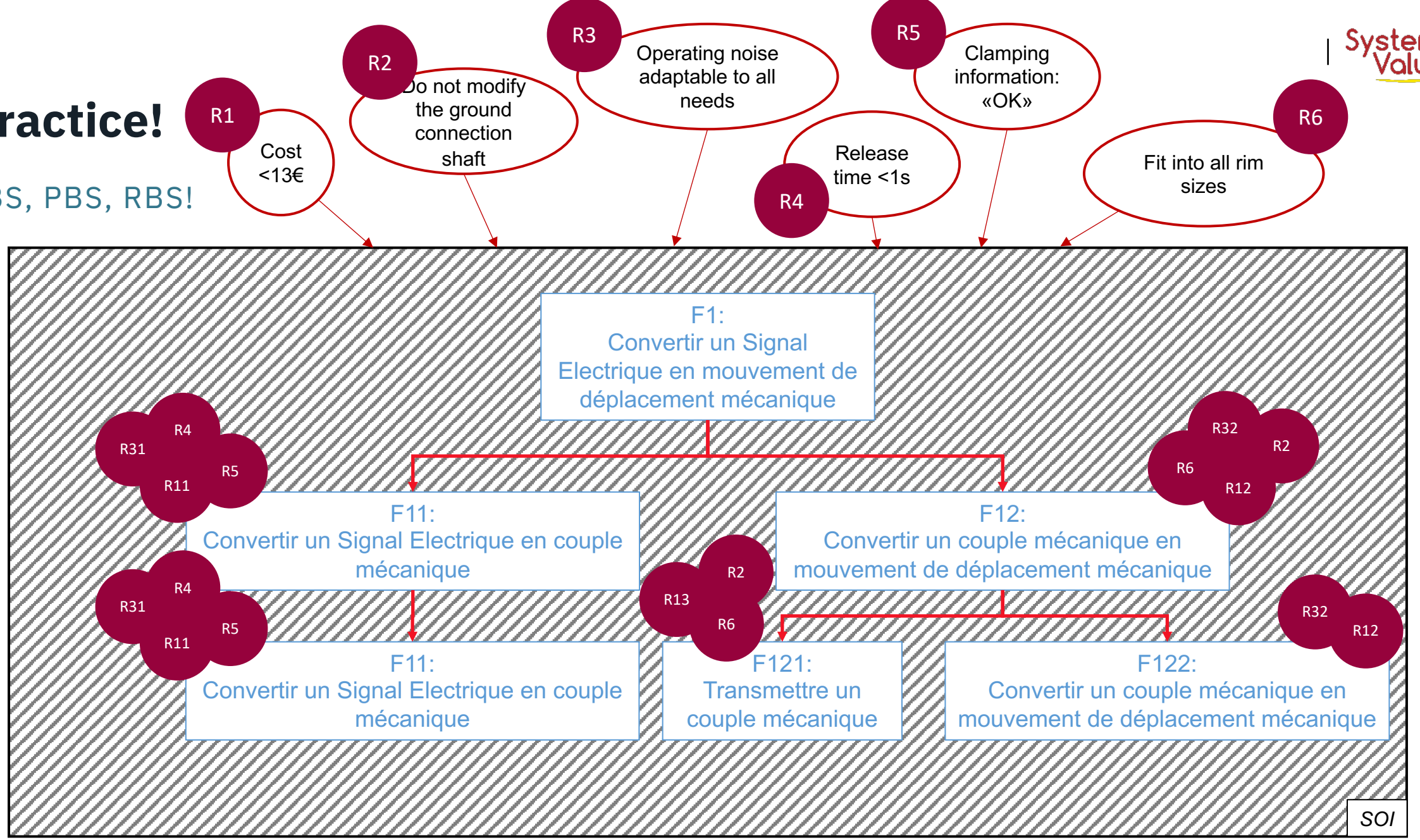
# Practice!

FBS, PBS, RBS!



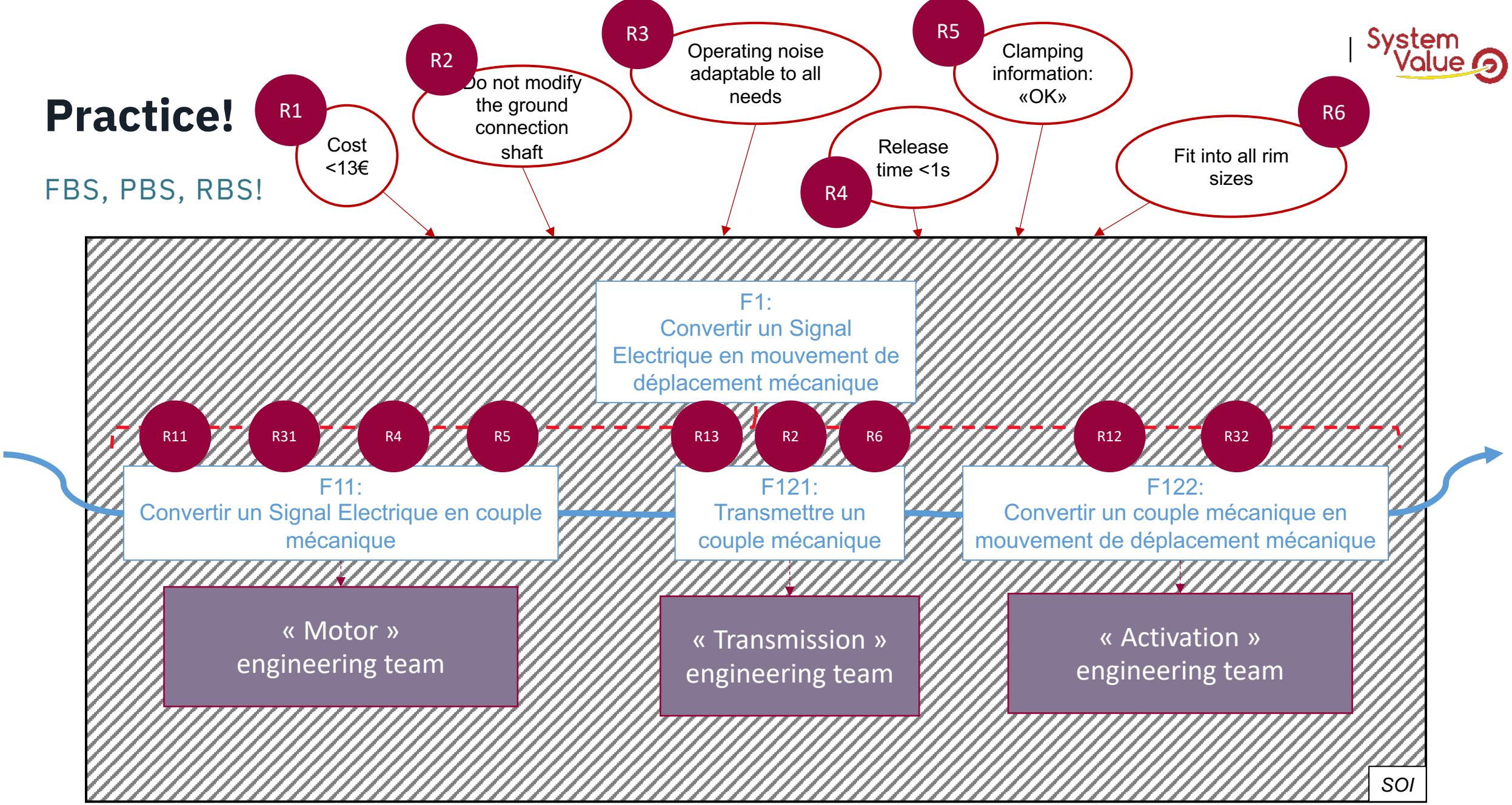
# Practice!

FBS, PBS, RBS!



# Practice!

FBS, PBS, RBS!



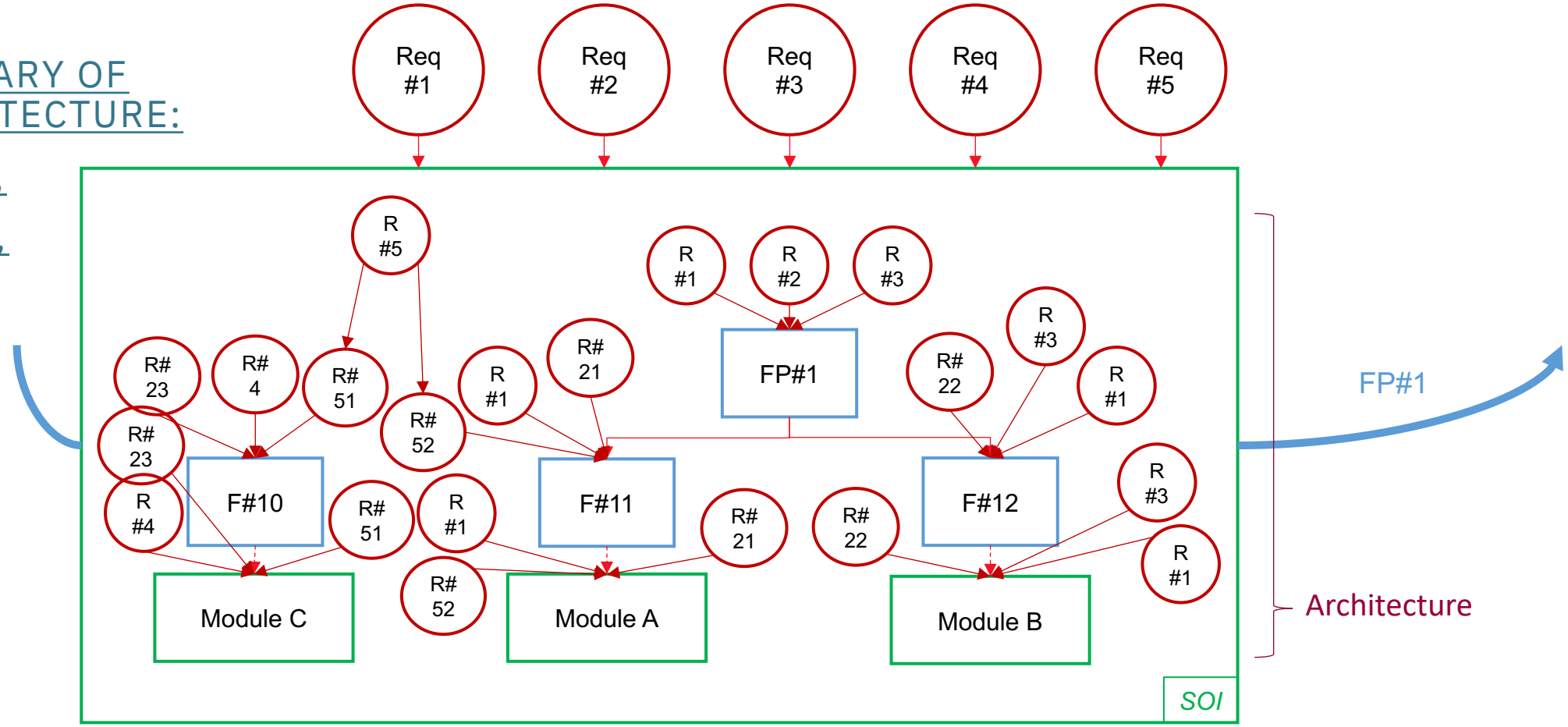
“

# Next step?

# From the black box to the white box: Through your layer

SUMMARY OF ARCHITECTURE:

- ✓ FBS,
- ✓ PBS,
- ✓ RBS

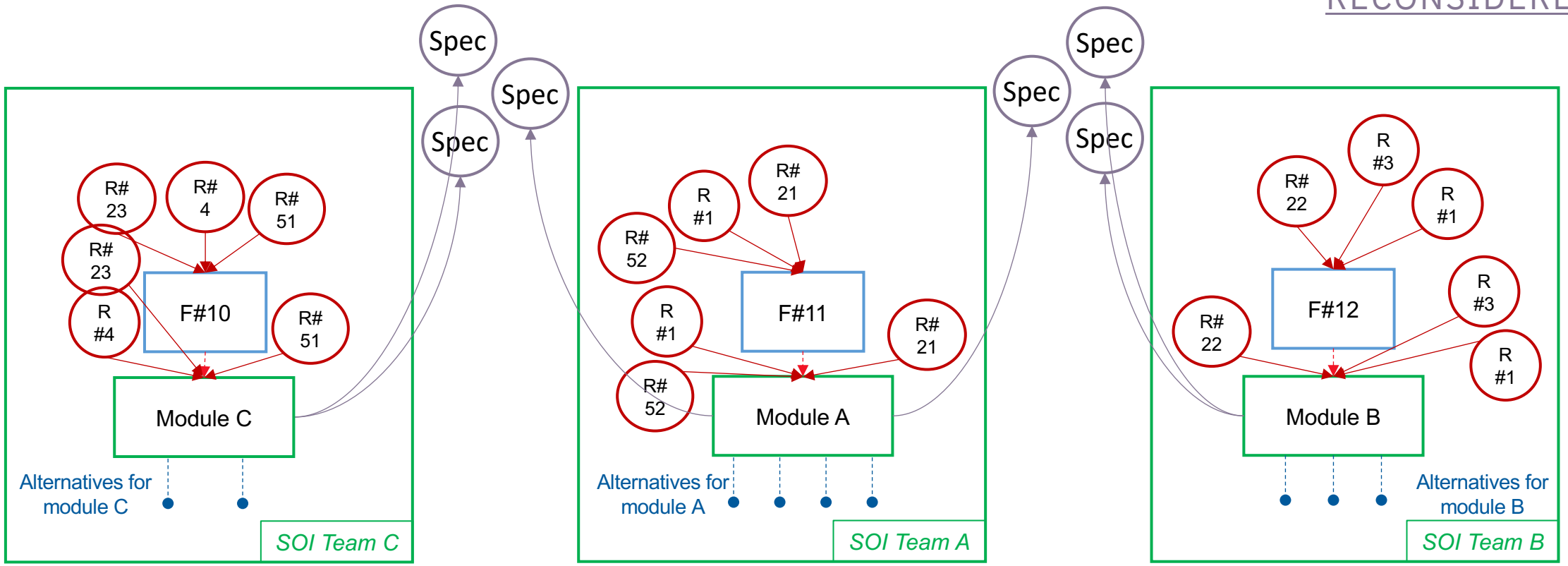


# From the black box to the white box: Down to sub-layer

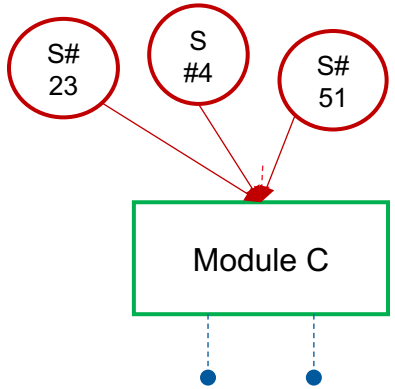
EACH SUB-LAYER'S TEAM WILL DO THE SAME AS YOU DID! DOWN TO ALTERNATIVES PROPOSAL

THEY HAVE TO ANSWER TO YOUR REQUEST BY SPECIFICATIONS, WITH FEW ALTERNATIVES

THESE MUST FULLFIL YOUR ASSUMPTION. IF NOT, ENGINEERING DESIGN MUST BE RECONSIDERED



# From the black box to the white box: Down to sub-layer



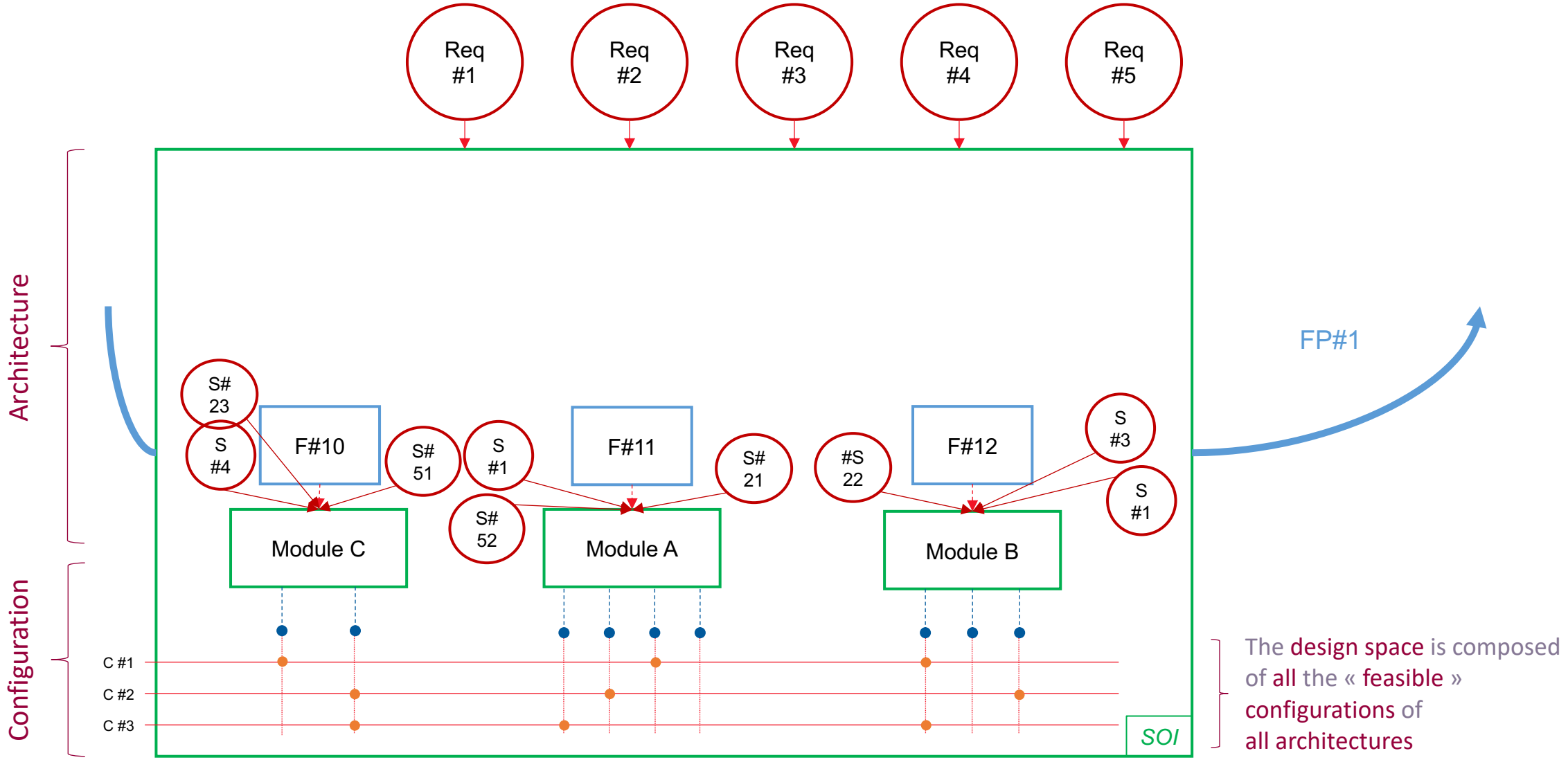

| | Alternative #1 | Alternative #2 |
|---|---|---|
| S#23 | 8 | 10 |
| S#4 | 12 | 15 |
| S#51 | 6 | 4 |

*SOI Team C*

“

# Back to you...

# From the black box to the white box: find best configurations




The design space is composed of all the « feasible » configurations of all architectures

# What « feasible » means?

## IT'S ALL ABOUT PERFORMANCE!

Concepts reaching the expected performance: the black box requirements also call High Level Requirements

**But it might be not enough!**
What's up if a competitor find a better concept (more performing than yours)?

« Feasible concepts » is much more about concepts over-performing than the expected performance!

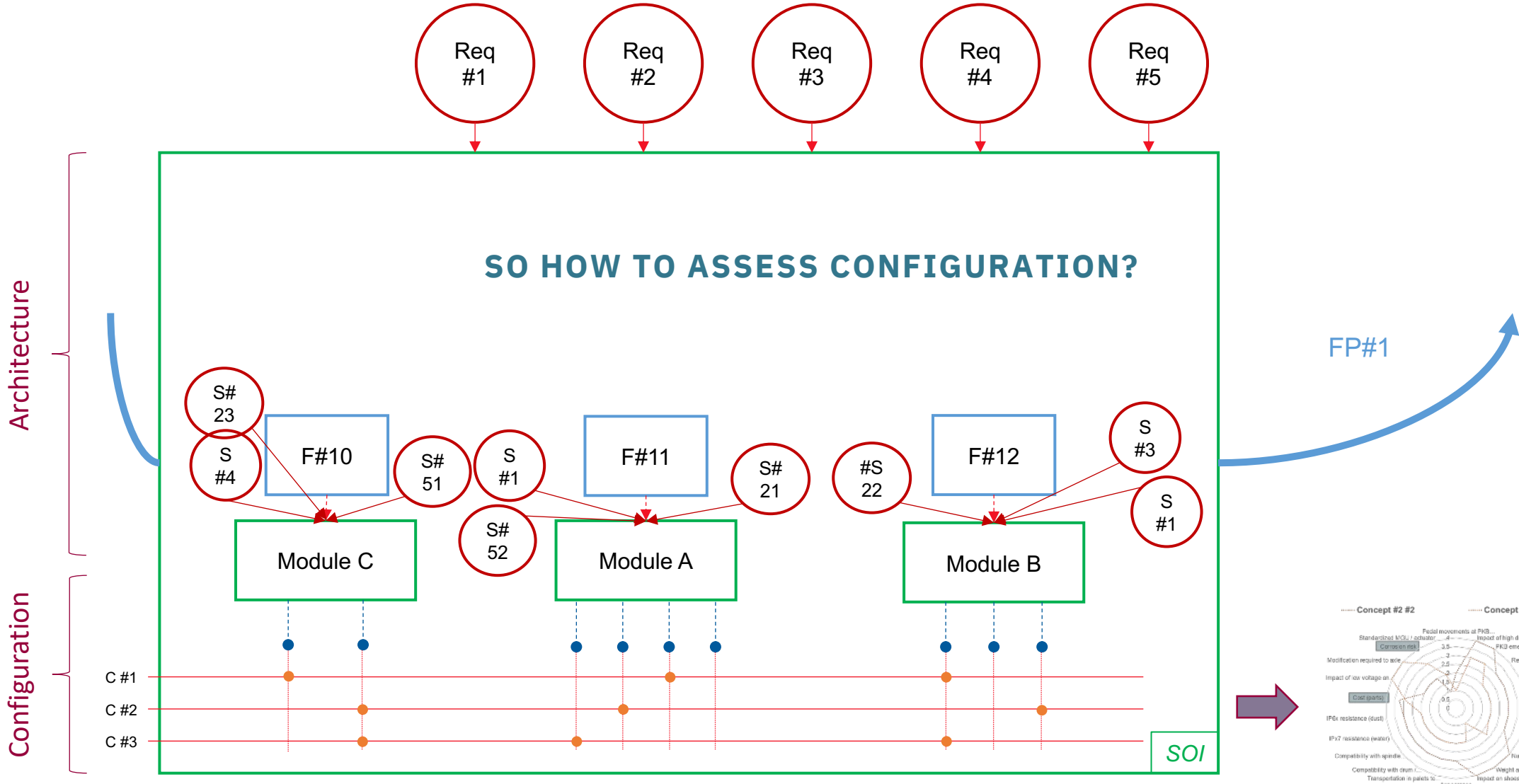
By analogy of « An innovation is not an innovation if we have not market for it »:

**« a feasible concepts is not feasible if it cannot be competitive »**

Today, engineering design trend to be satisfying to a set of requirements and almost never optimal in this set

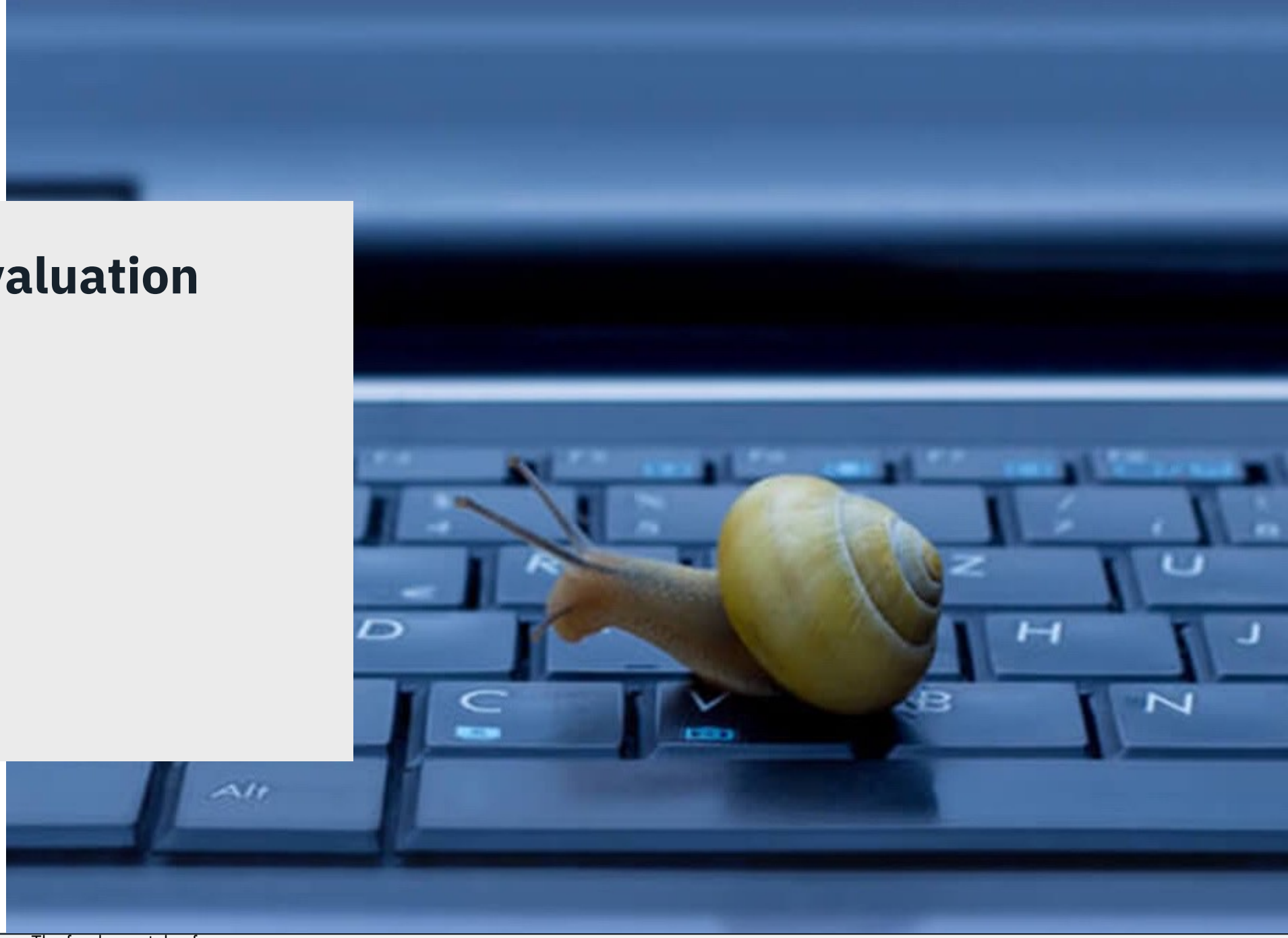
At the end, you MUST make sure you find the best possible architectures and configurations leading to set both the right trade-offs & the best performance

# From the black box to the white box: evaluate to compare configurations

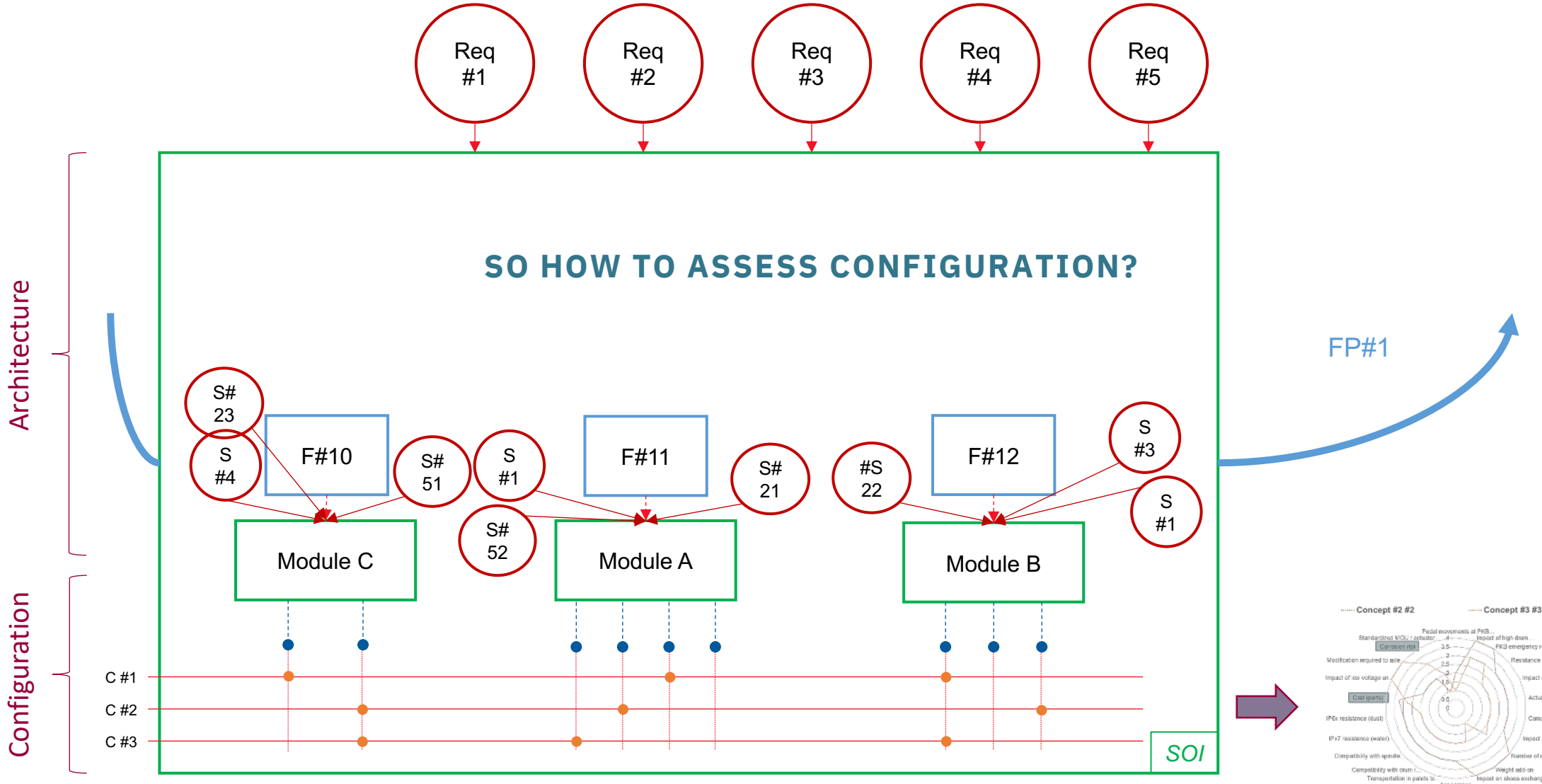


# 03

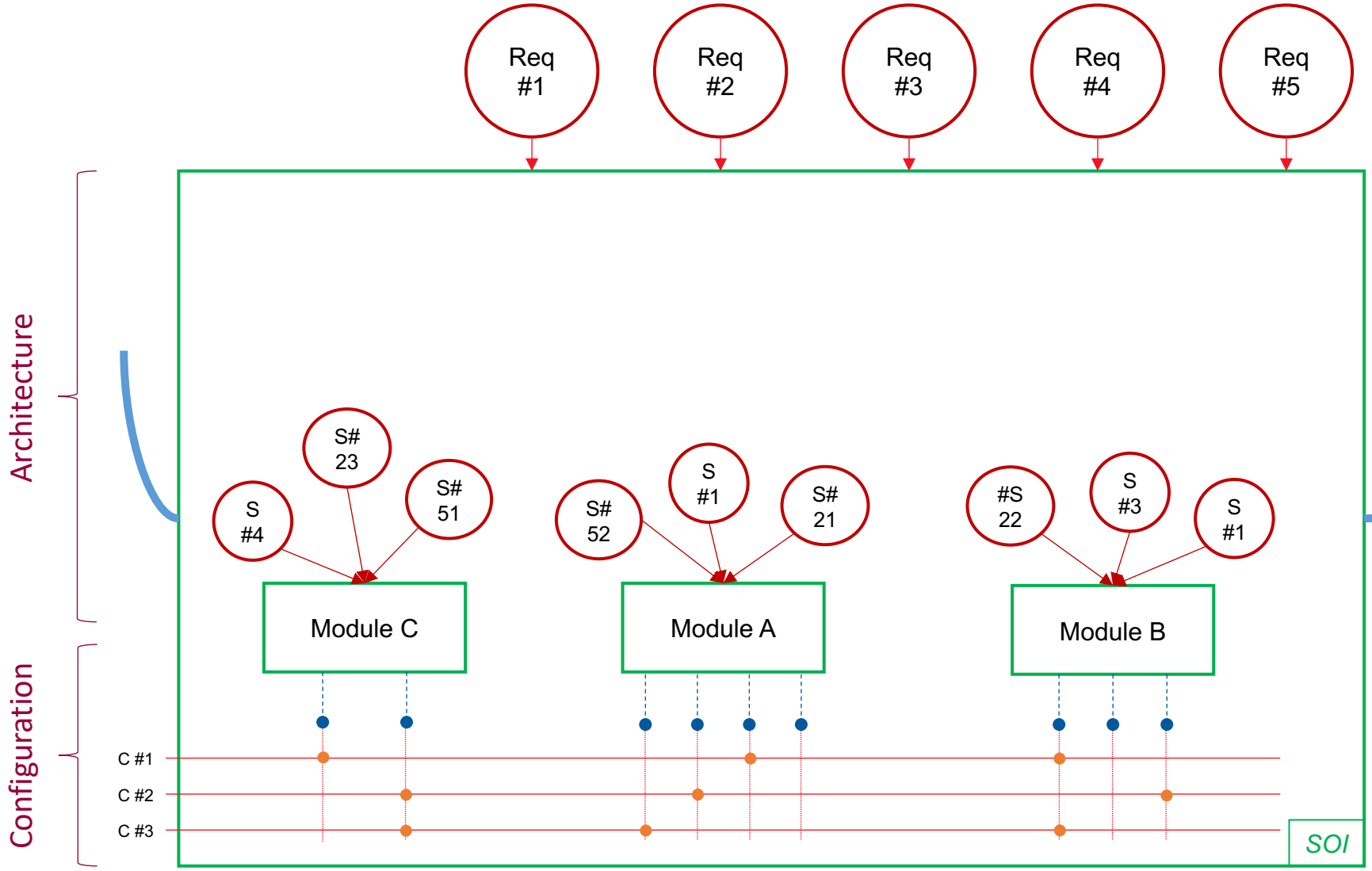
# Architecture Evaluation

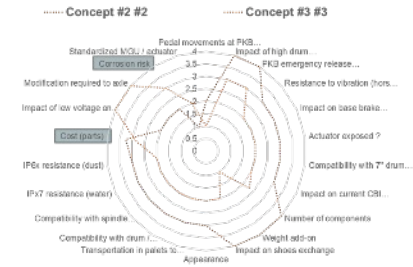


# Architecture evaluation

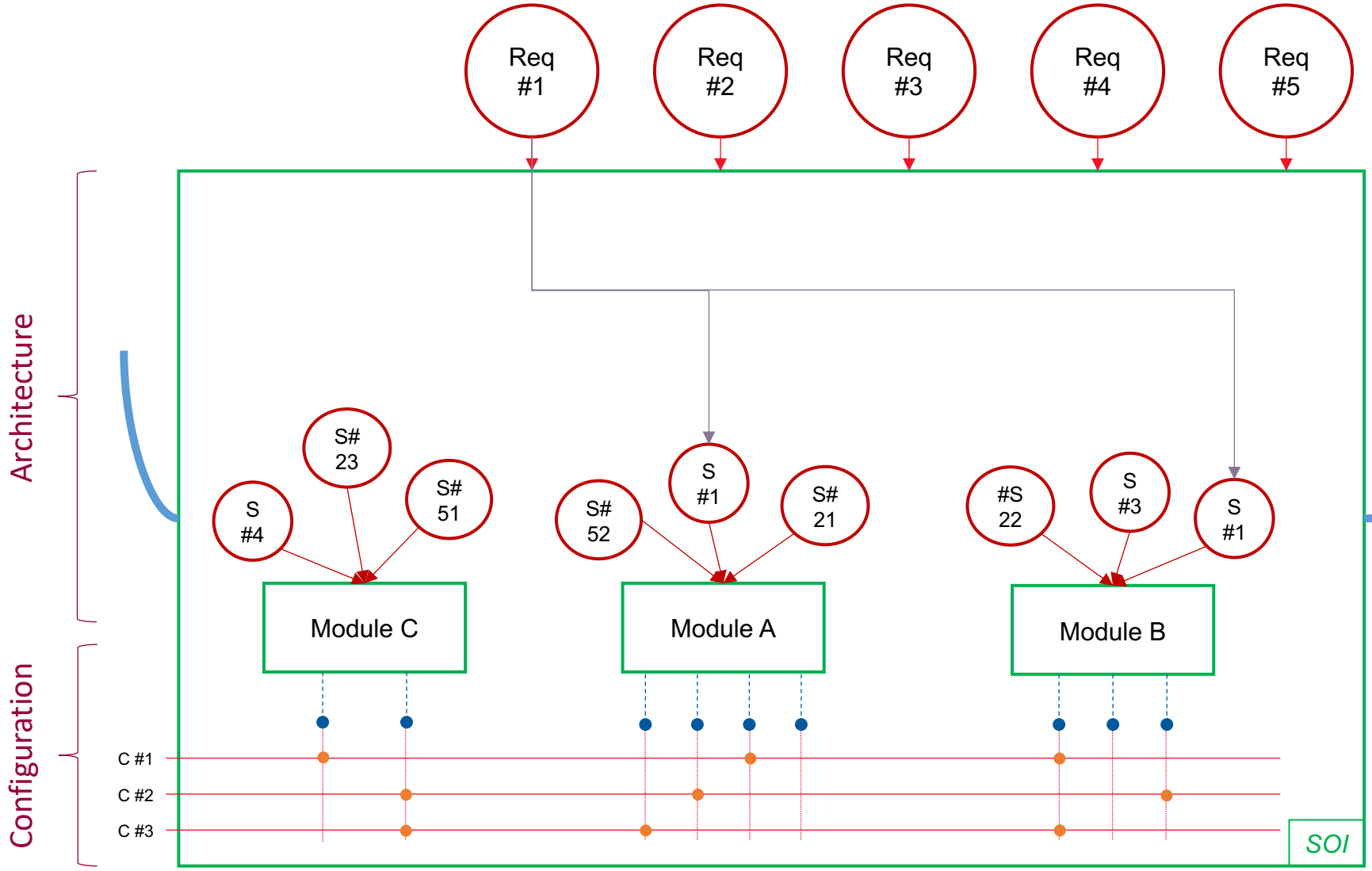


# Architecture evaluation



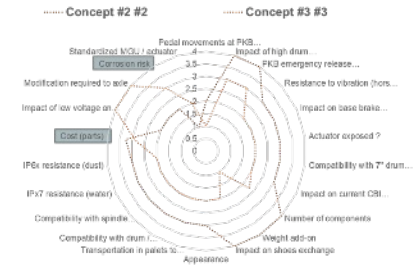



# Architecture evaluation: REQ#1

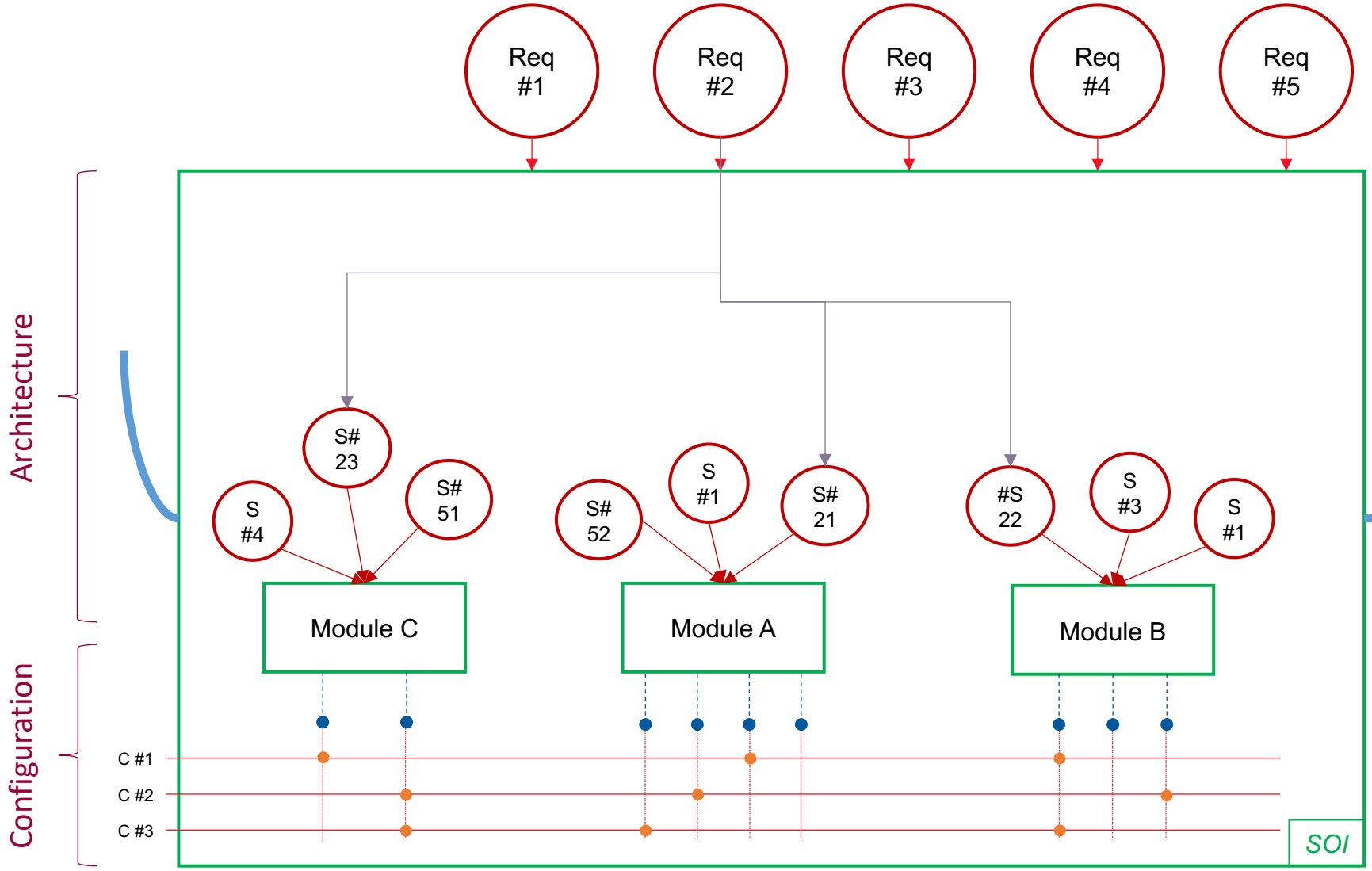



Evaluation rules:

✓ REQ#1=min(S#1)

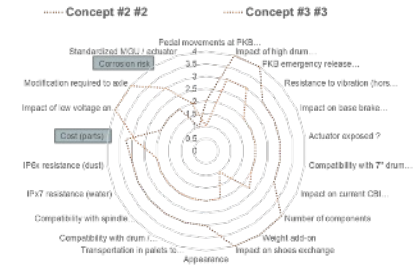


# Architecture evaluation: REQ#2

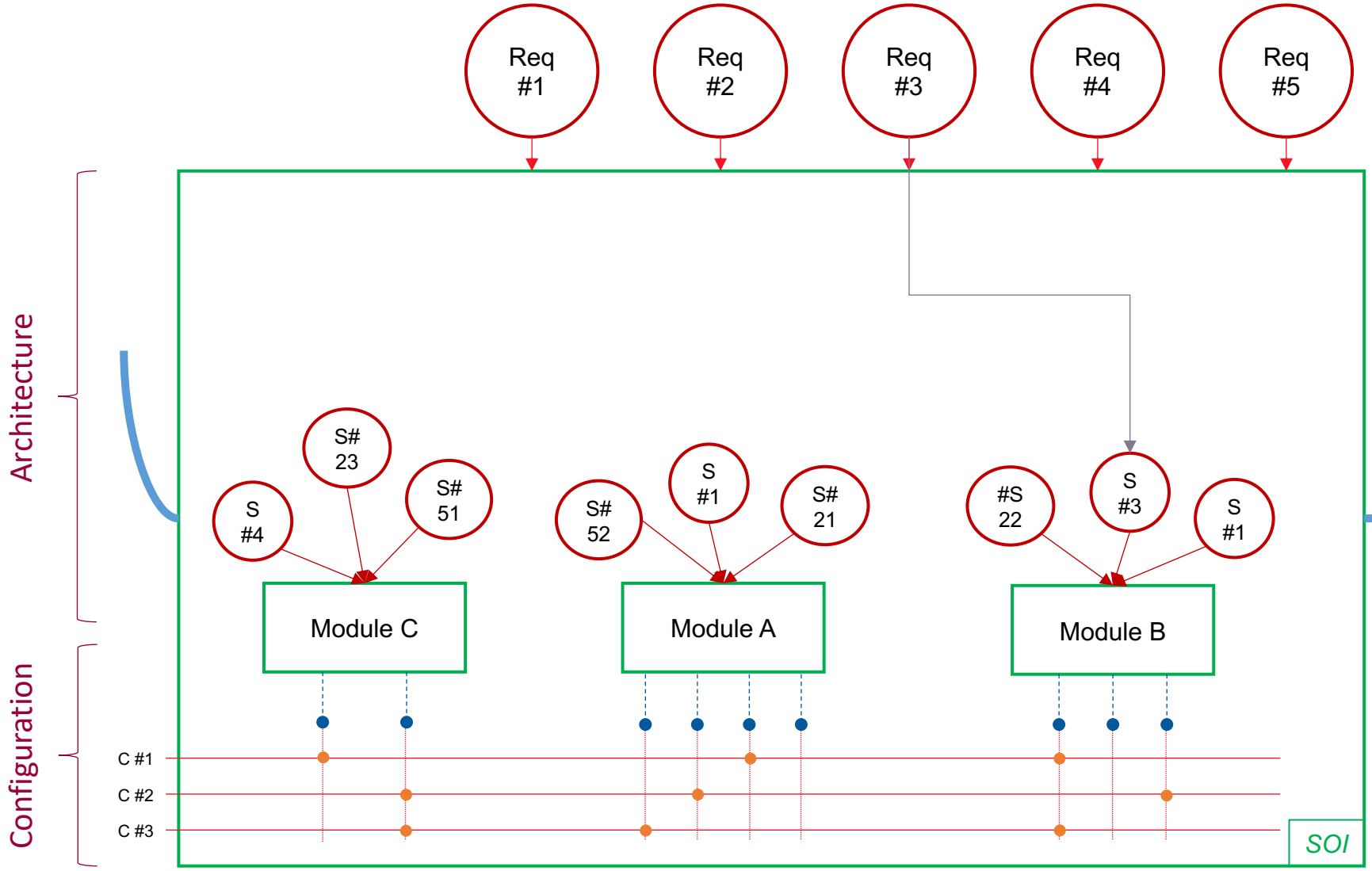



Evaluation rules:

- ✓ REQ#1=min(S#1)
- ✓ REQ#2=S#21+S#22+S#23

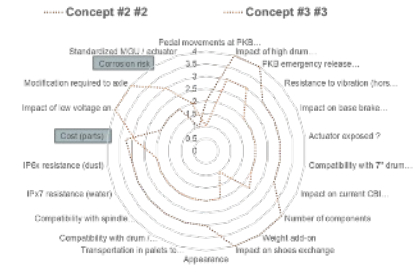


# Architecture evaluation: REQ#3

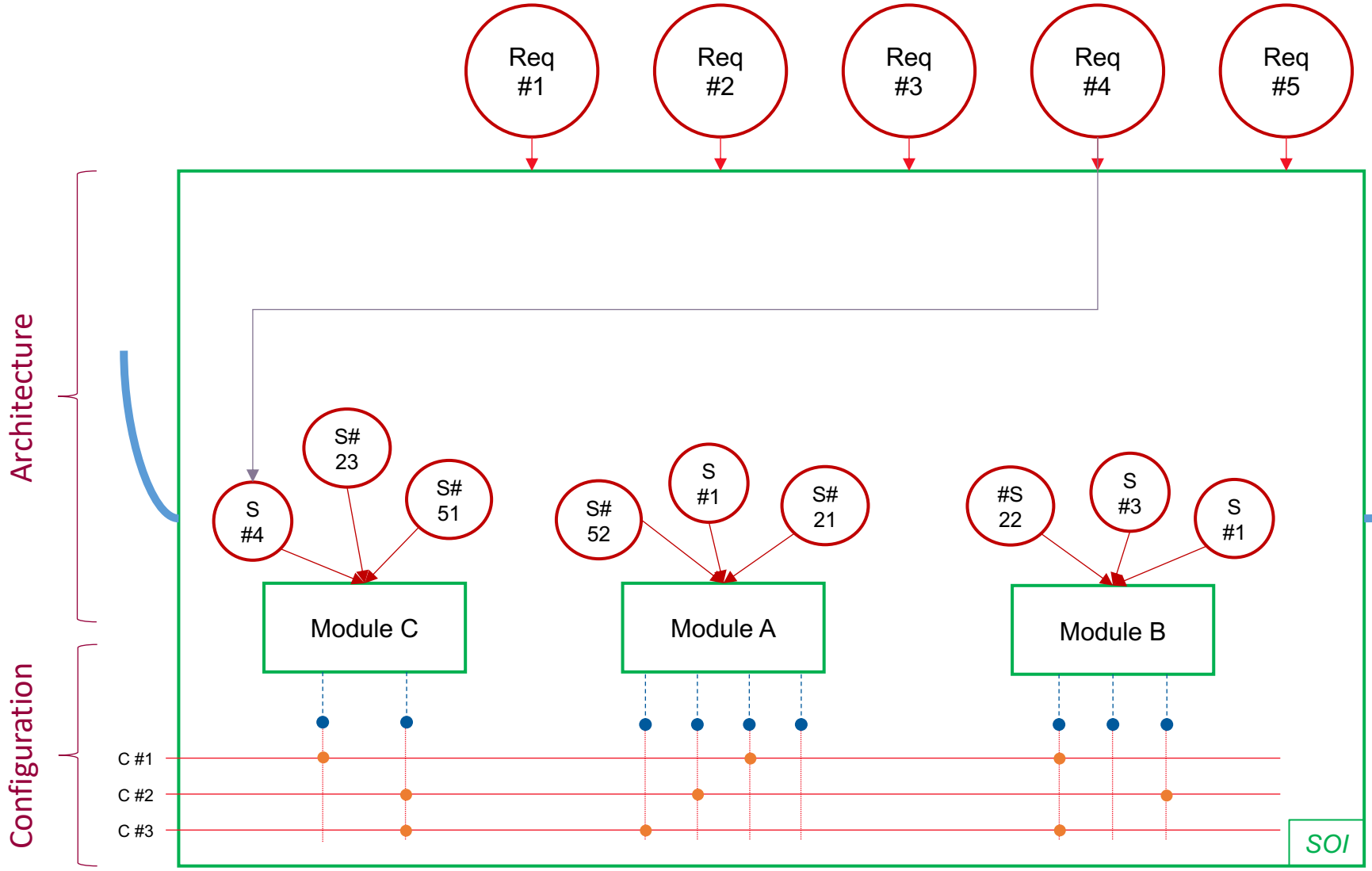



Evaluation rules:

- ✓ REQ#1=min(S#1)
- ✓ REQ#2=S#21+S#22+S#23
- ✓ REQ#3=Module B.S#3

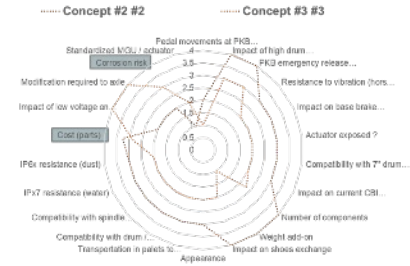


# Architecture evaluation: REQ#4

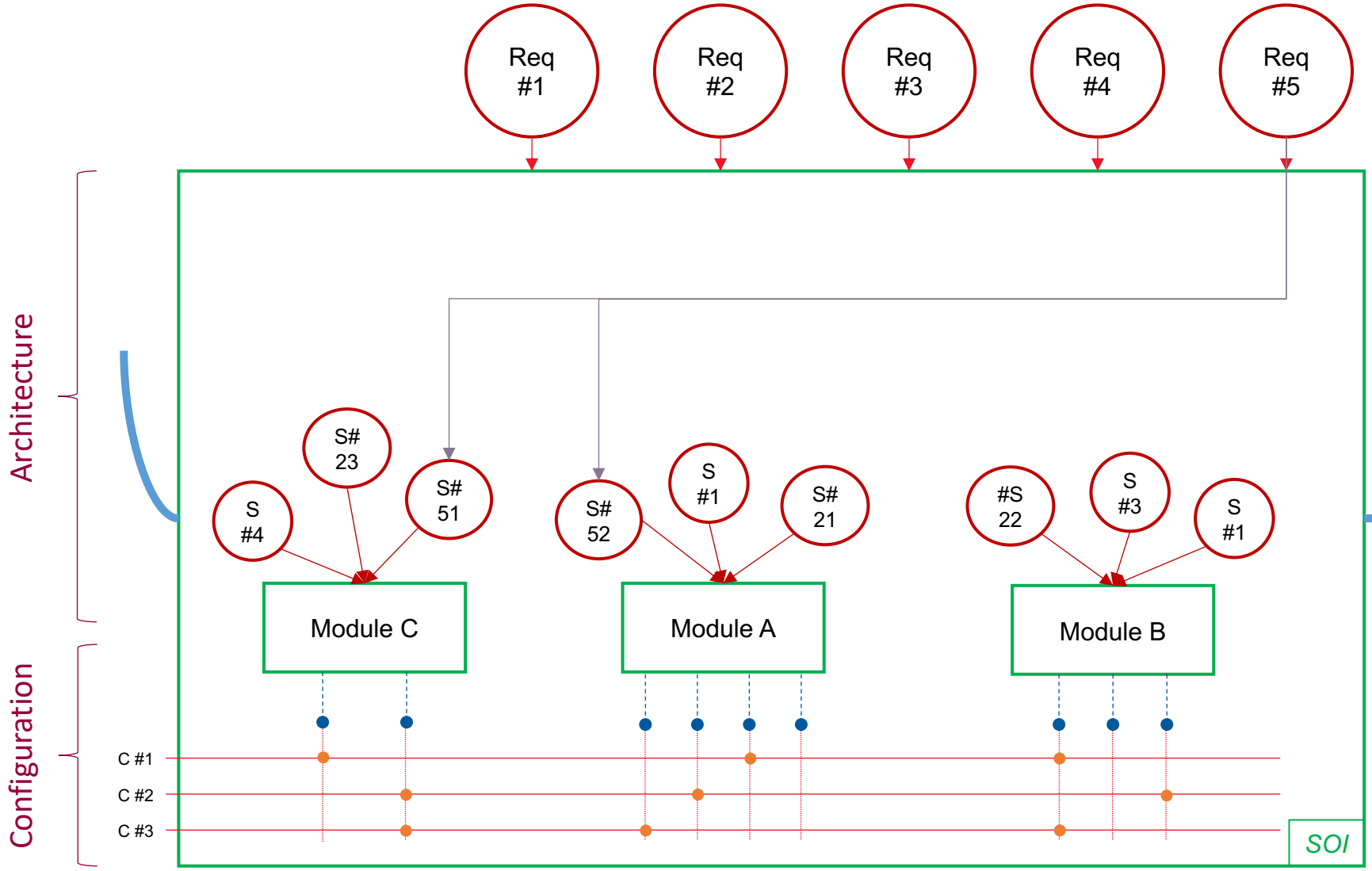



Evaluation rules:

- ✓ REQ#1=min(S#1)
- ✓ REQ#2=S#21+S#22+S#23
- ✓ REQ#3=Module B.S#3
- ✓ REQ#4=Module C.S#4

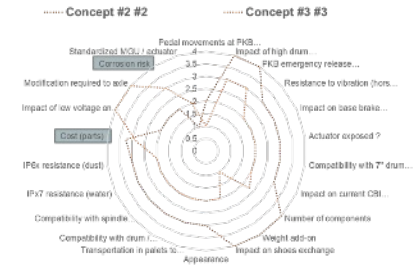


# Architecture evaluation: REQ#5

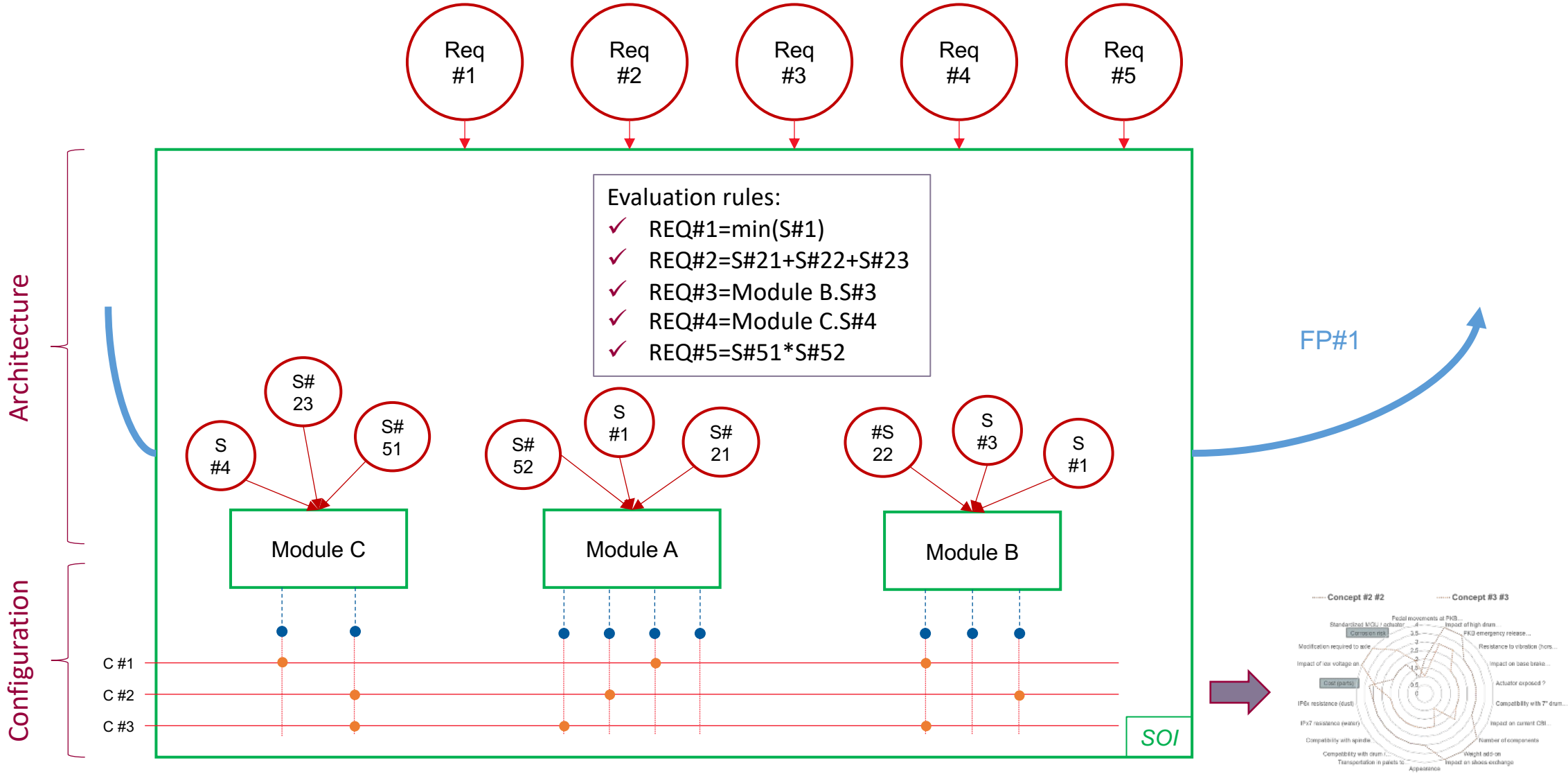



Evaluation rules:

- ✓ REQ#1=min(S#1)
- ✓ REQ#2=S#21+S#22+S#23
- ✓ REQ#3=Module B.S#3
- ✓ REQ#4=Module C.S#4
- ✓ REQ#5=S#51*S#52

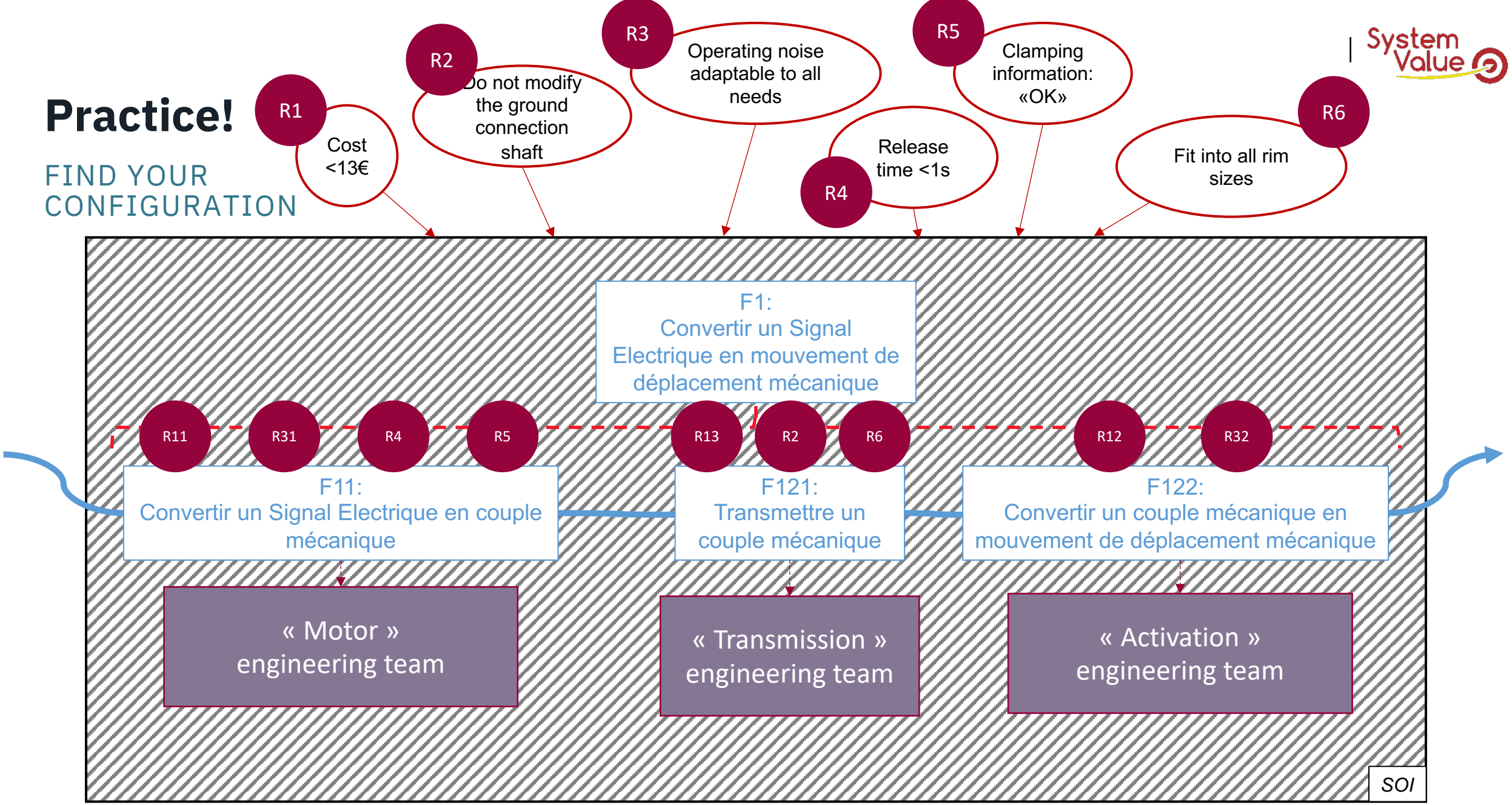


# Mastering architecture evaluation: Setting and applying rules



# Practice!

## FIND YOUR CONFIGURATION



“

# Finally...

# Comparing concepts

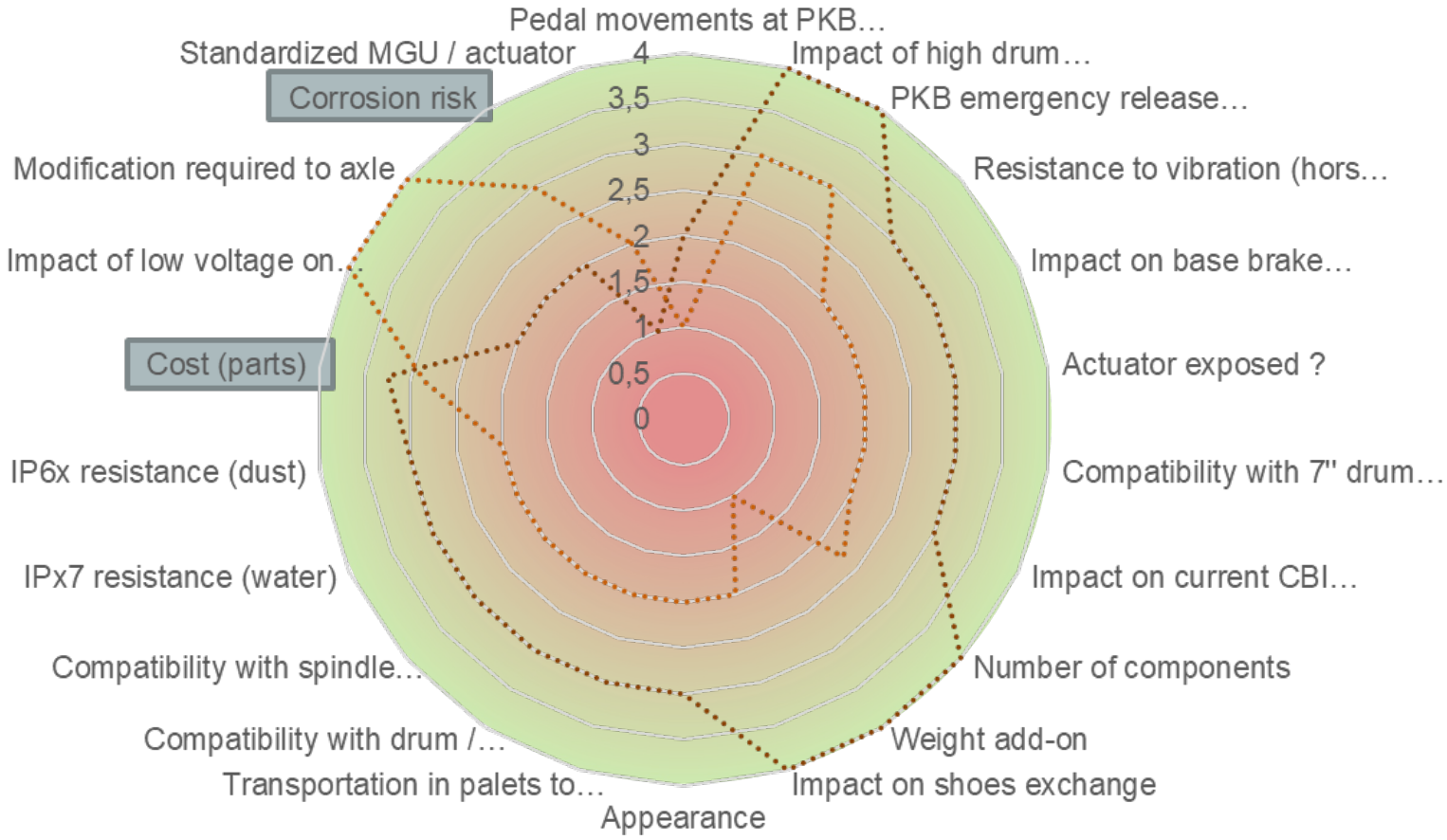
## CONCEPT#1 & CONCEPT#2

*Preparing work:*

- ✓ Reverse performance to minimize
  *To have the expected target outside of the radar graph. For instance, the cheapest cost will be equal to 4; the more expensive at the center*

*Analyse it!*

- ✓ *Any comments?*



# Comparing concepts

## CONCEPT#1 & CONCEPT#2

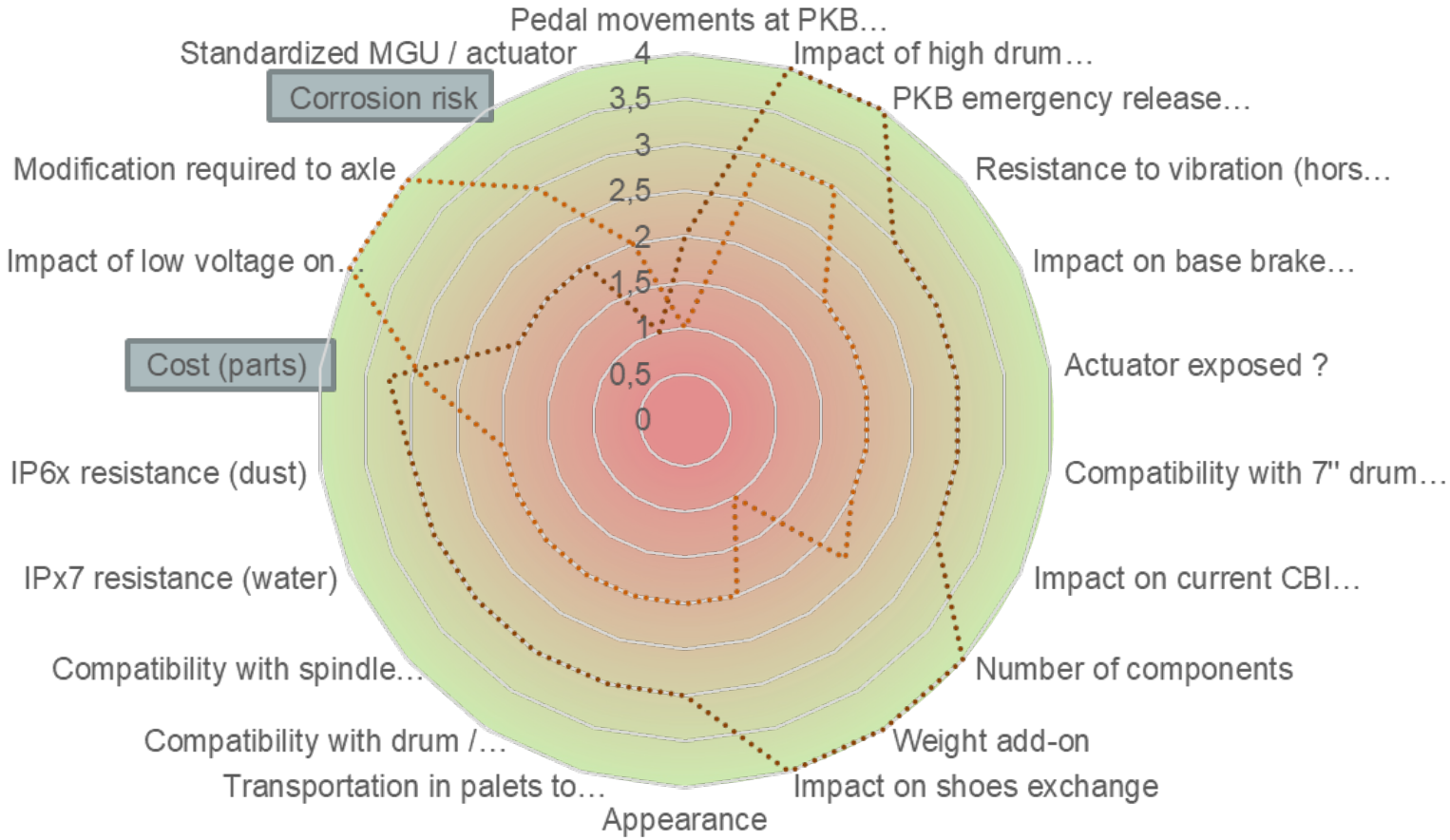
*Analyzing work:*

- ✓ These two concepts are clearly a trade-off,
- ✓ The surface is not a good KPI!

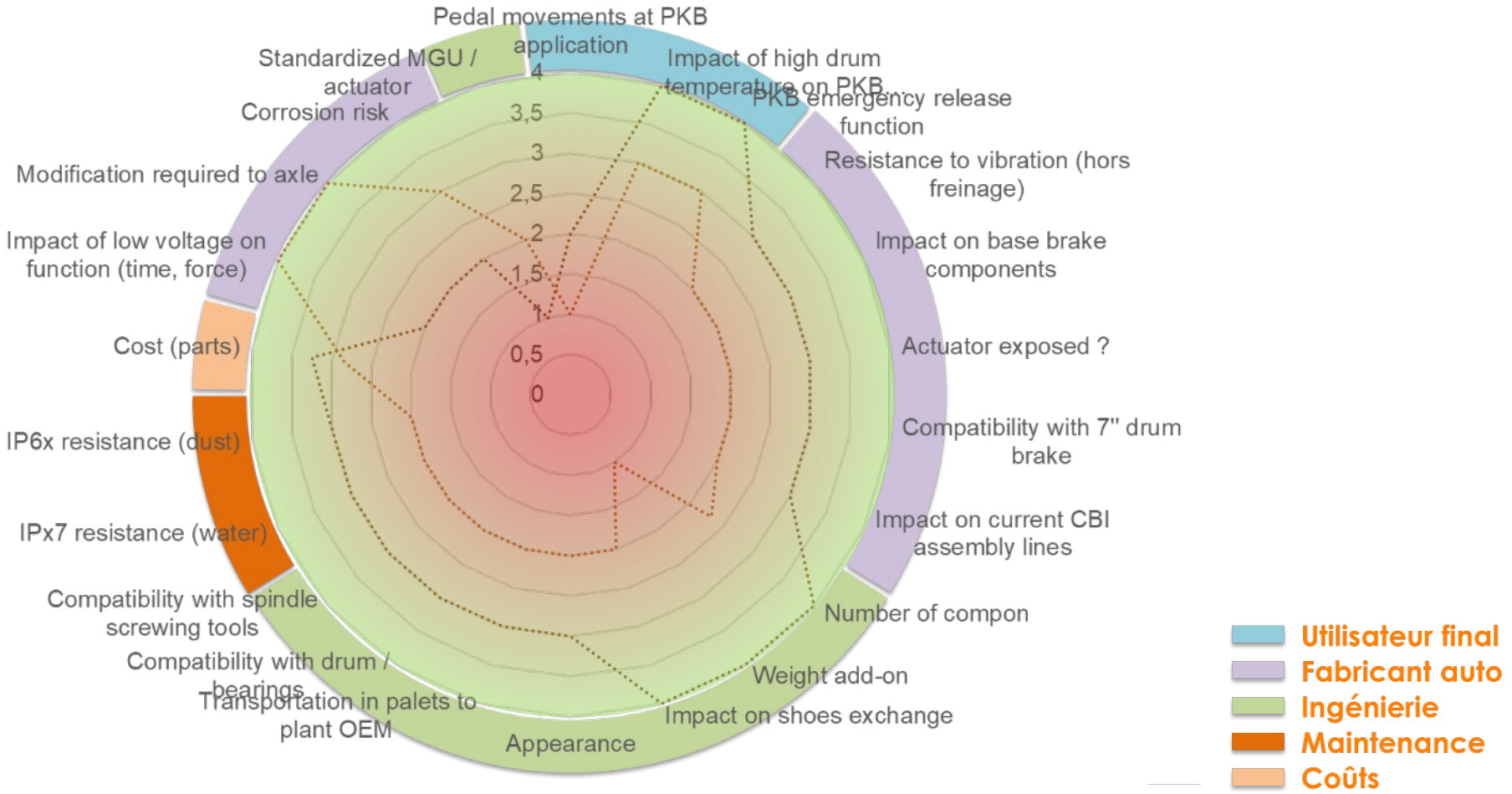
*More data:* concept #1 is the competition!

- ✓ Do we need to be so much over-performing on « weight add-on »?
  - ✓ Recommandation to my team: propose concepts stille better but closer to competition!

Architect role is to manage trade-offs!



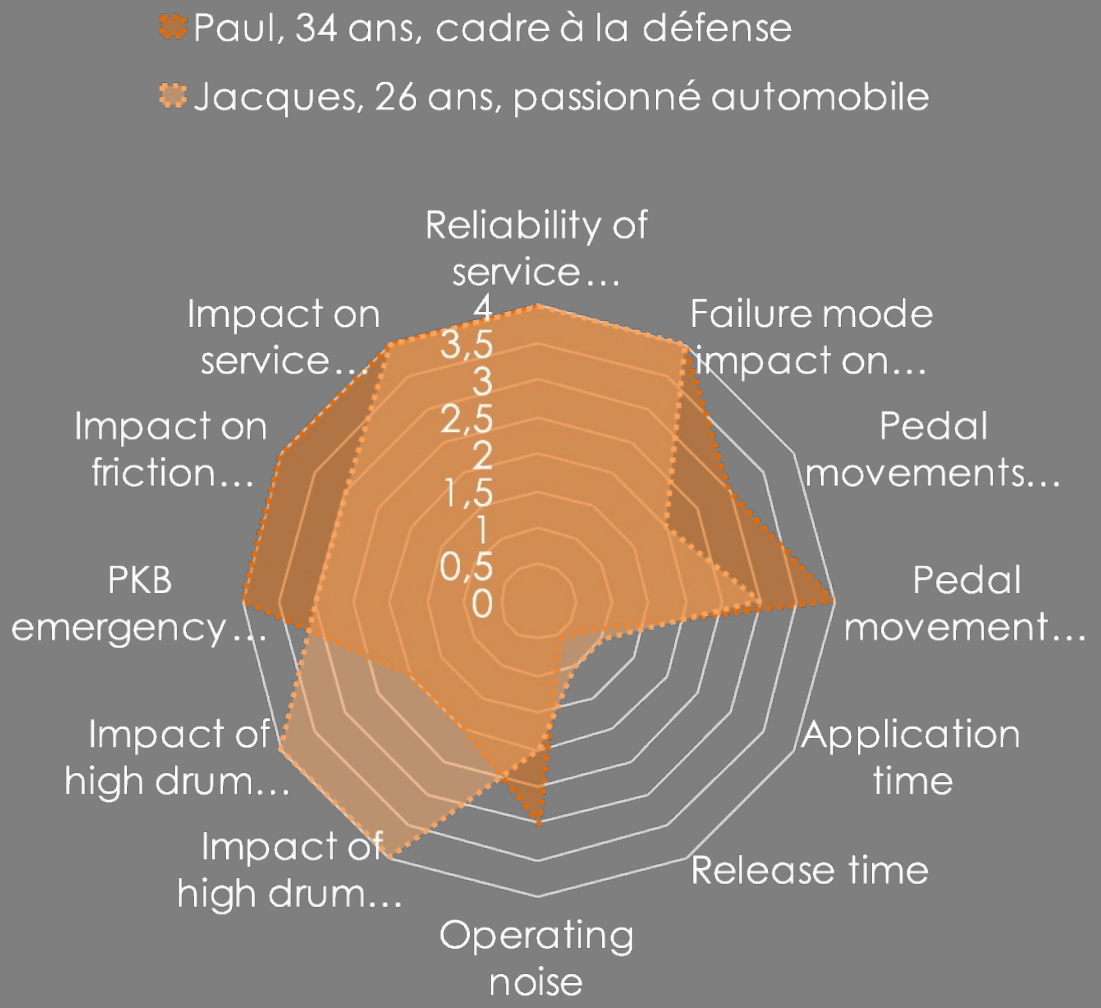
# Comparing concepts with an highlight...

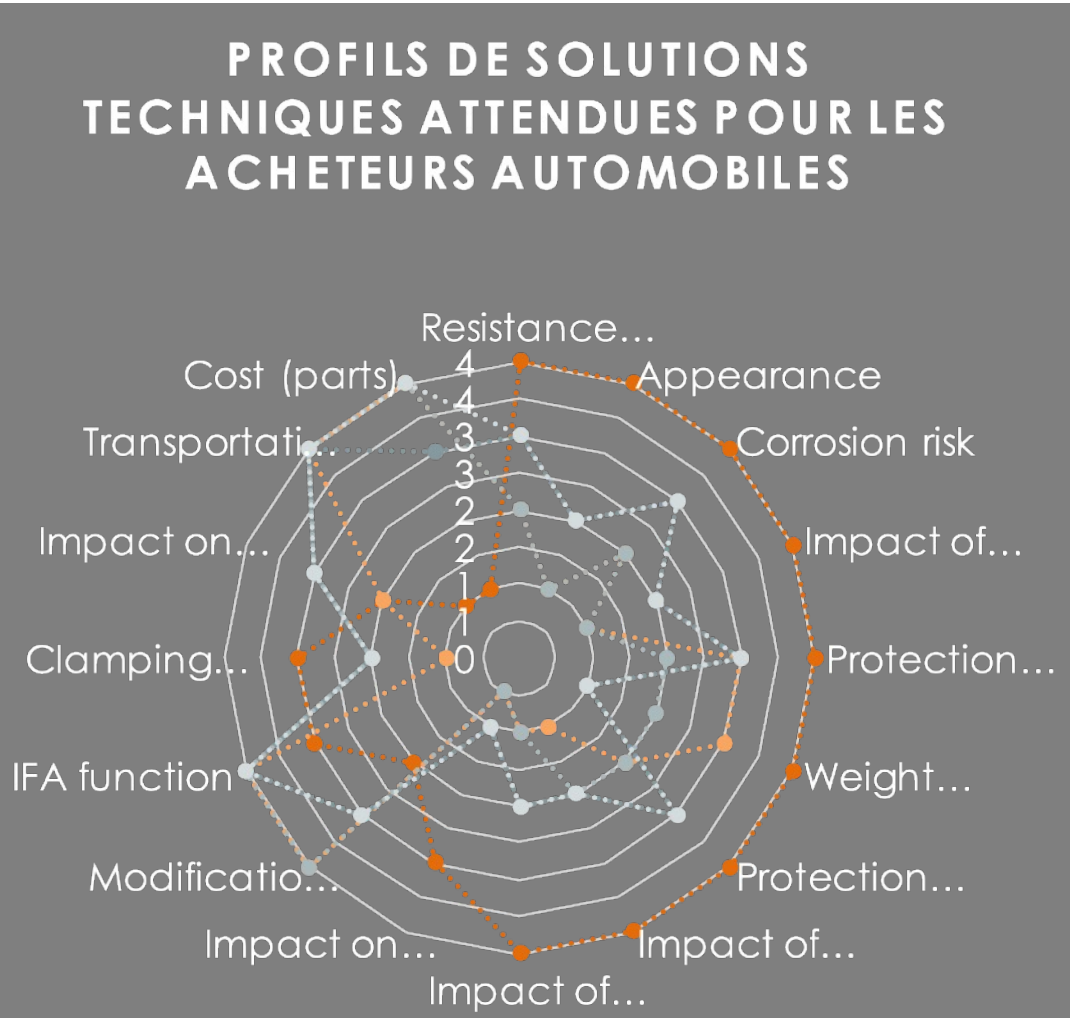


# 04

## How to take decisions?

# Step back: Stakeholder's Point-of-Views



“

# Introducing « classical » decision approach

# Weighted Sum

## WEIGHTED SUM IS ABOUT AGGREGATING KEY PERFORMANCE INDICATORS IN ORDER TO HAVE AN OVERALL BIG "QUOTATION" OF YOUR SOI

For instance, considering that the "price" is twice time much more important than the "mass", a decision criteria could be to maximise the following:

"Overall Decision Criteria" = 2 * "SOI's Price" + 1 * "SOI's Mass"

Bullshit!

So:

| | *Initial data* | | *Scale change* | |
|---|---|---|---|---|
| | **SOI #1** | **SOI #2** | **SOI #1** | **SOI #2** |
| Price (€) | 1000 | 1500 | 0,6 | 0,4 |
| Mass (kg} | 3 | 2 | 0,4 | 0,6 |
| **ODC** | | | **1,6** | **1,4** |

*Final result (select the best value)*

“

If you want to analyze system quintescence using weighted sum, stop to be an engineer!

# Why Weighted Sum is over ?

## AN EXAMPLE

Considering that the “price” is twice time much more important than the “mass” can be true in a validity domain (so with limits!)

For instance, if I’m a business man, “price” can have a importance over a certain value (>=2000€) but can have no more importance below this value…

So:

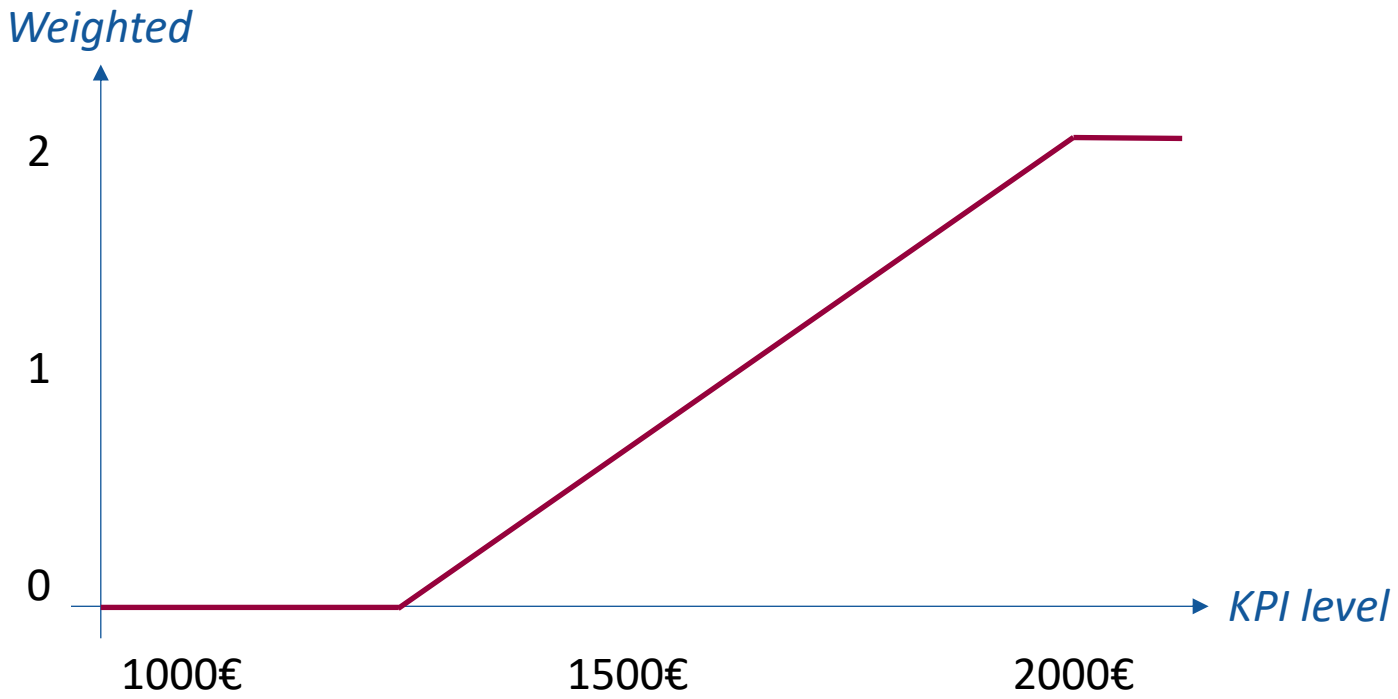
| | *Original data* | | *Scale change* | |
|---|---|---|---|---|
| | **SOI #1** | **SOI #2** | **SOI #1** | **SOI #2** |
| Price (€) | 1000 | 1500 | 0 | 0 |
| Mass (kg} | 3 | 2 | 0,4 | 0,6 |
| **ODC** | | | **0,4** | **0,6** |

*Final result (select the best value)*

# Introduction to “Dynamic Weighted Sum”

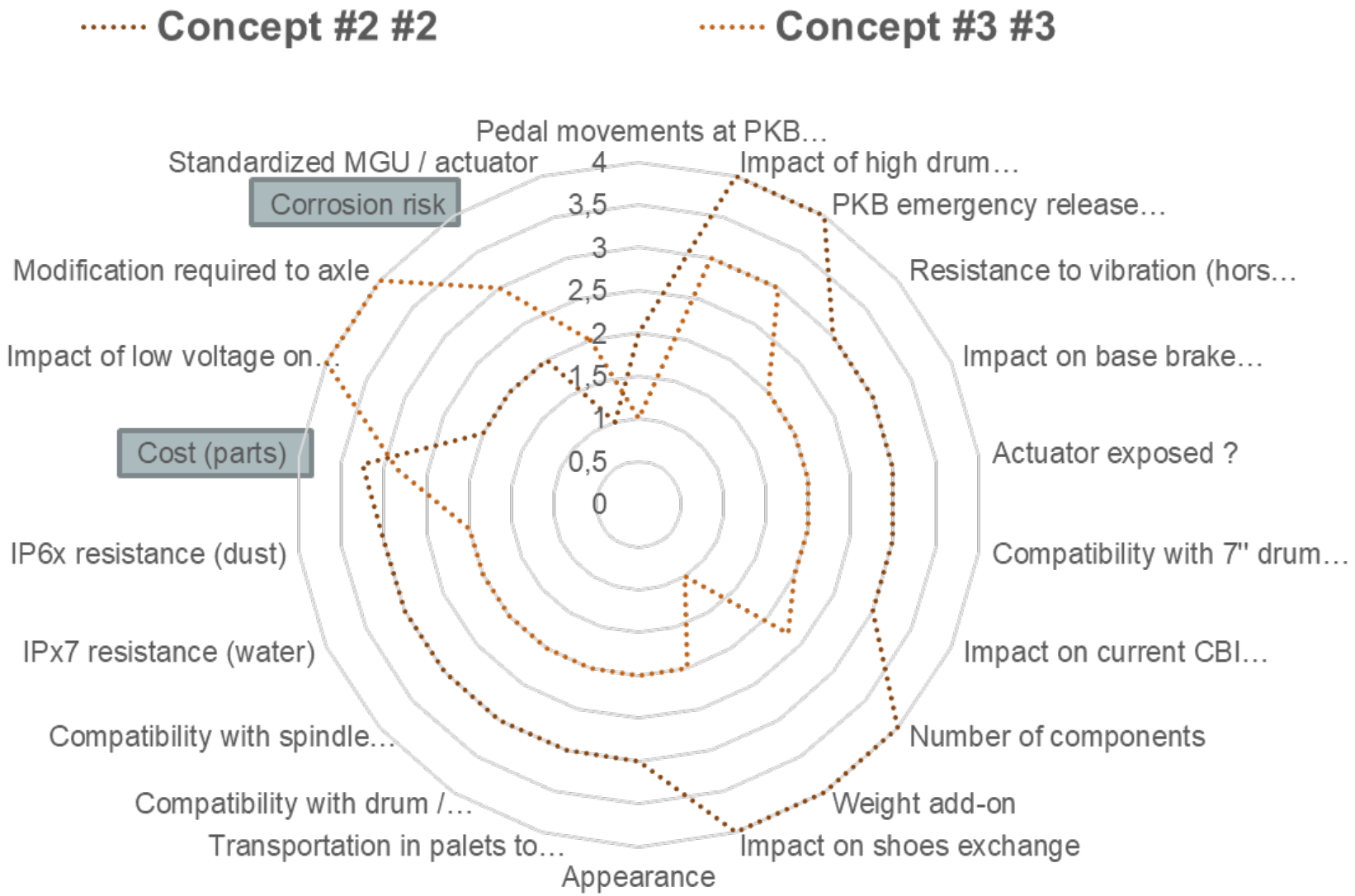
## DYNAMIC WEIGHTED SUM IS LOOKING TO SOLVE THE PROBLEM OF STATIC WEIGHTED

Considering that the weighted might change with KPI level it’s essential to capitalize the user preference in term of “shape”:

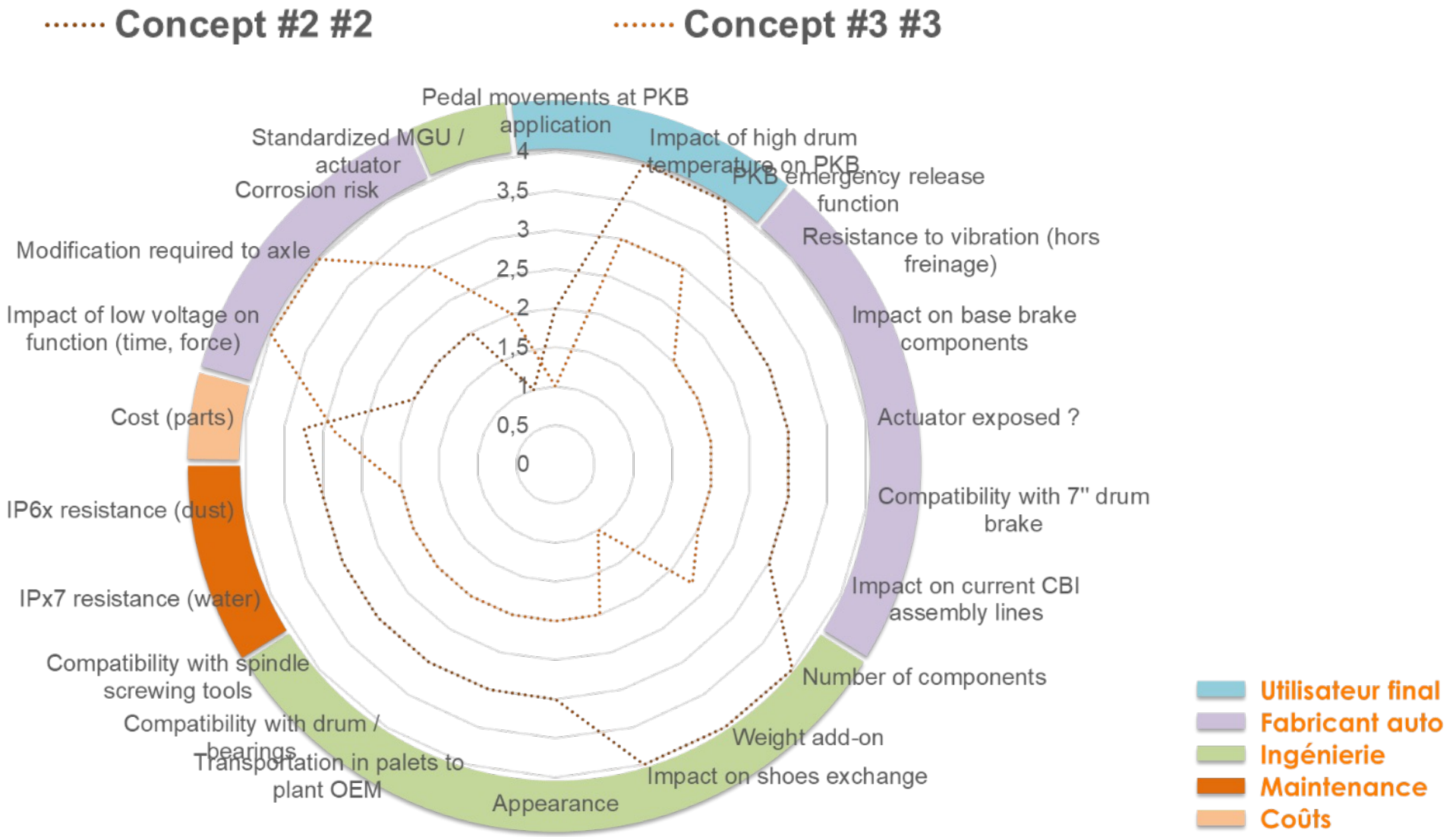



Weighted = If(”Price”)<1250 then 0
elseif(“Price”>2000 then 2
else XXX

# Objectives: performance assessment



# Objectives: performance assessment

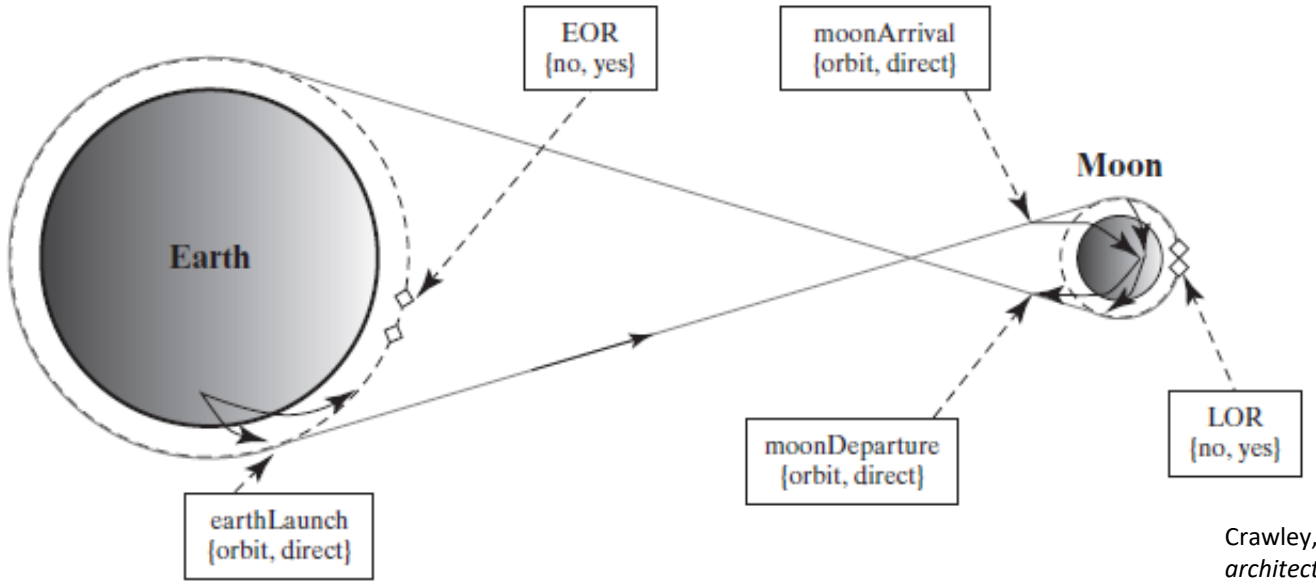


“

# Introducing Pareto Front

# Systems architecting as a decision-making process

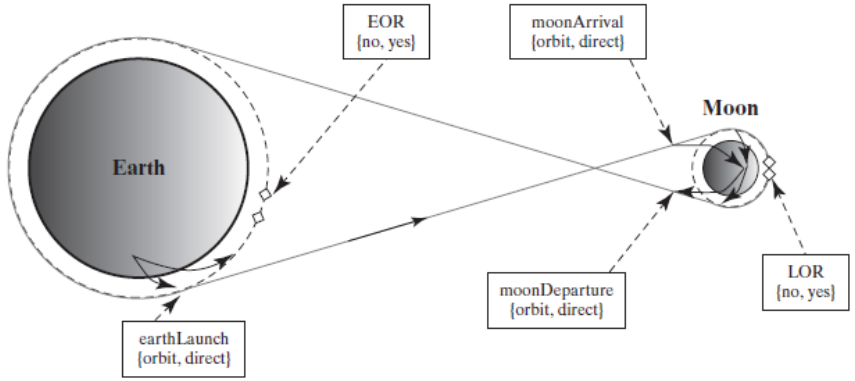
| shortID | Decision | units | alt A | alt B | alt C | alt D |
|---|---|---|---|---|---|---|
| EOR | Earth Orbit Rendezvous | none | no | yes | | |
| earthLaunch | Earth Launch Type | none | orbit | direct | | |
| LOR | Lunar Orbit Rendezvous | none | no | yes | | |
| moonArrival | Arrival At Moon | none | orbit | direct | | |
| moonDeparture | Departure From Moon | none | orbit | direct | | |
| cmCrew | Command Module Crew | people | 2 | 3 | | |
| lmCrew | Lunar Module Crew | people | 0 | 1 | 2 | 3 |
| smFuel | Service Module Fuel | none | cryogenic | storable | | |
| lmFuel | Lunar Module Fuel | none | NA | cryogenic | storable | |

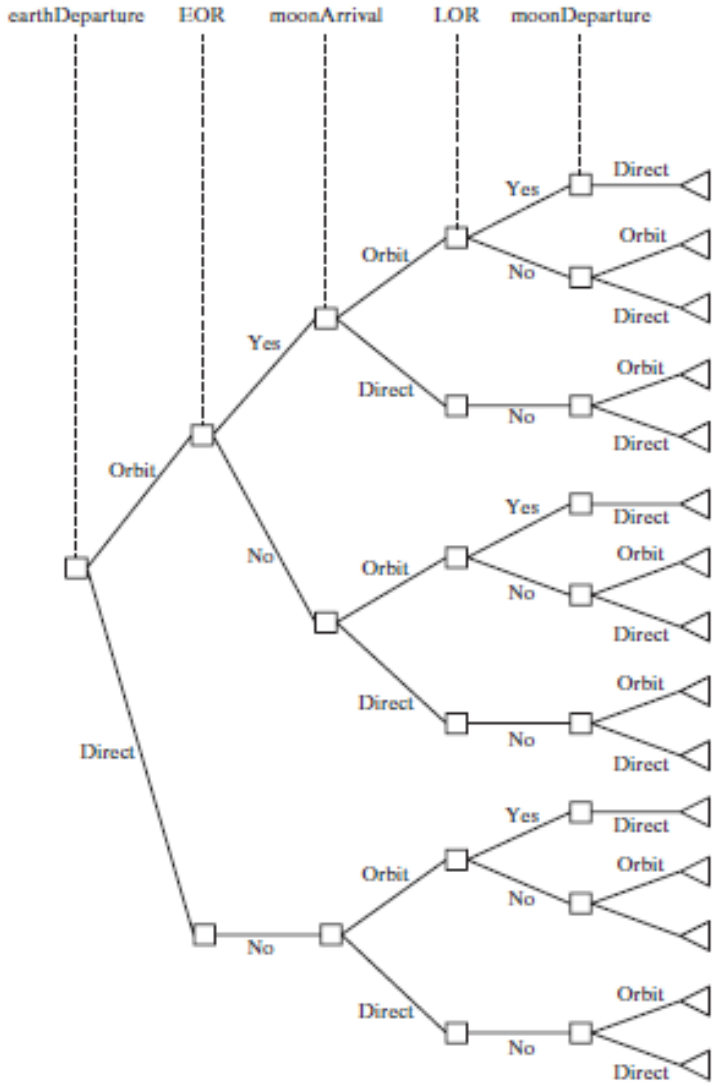



Crawley, E., Cameron, B., & Selva, D. (2015). *System architecture: strategy and product development for complex systems*. Prentice Hall Press.

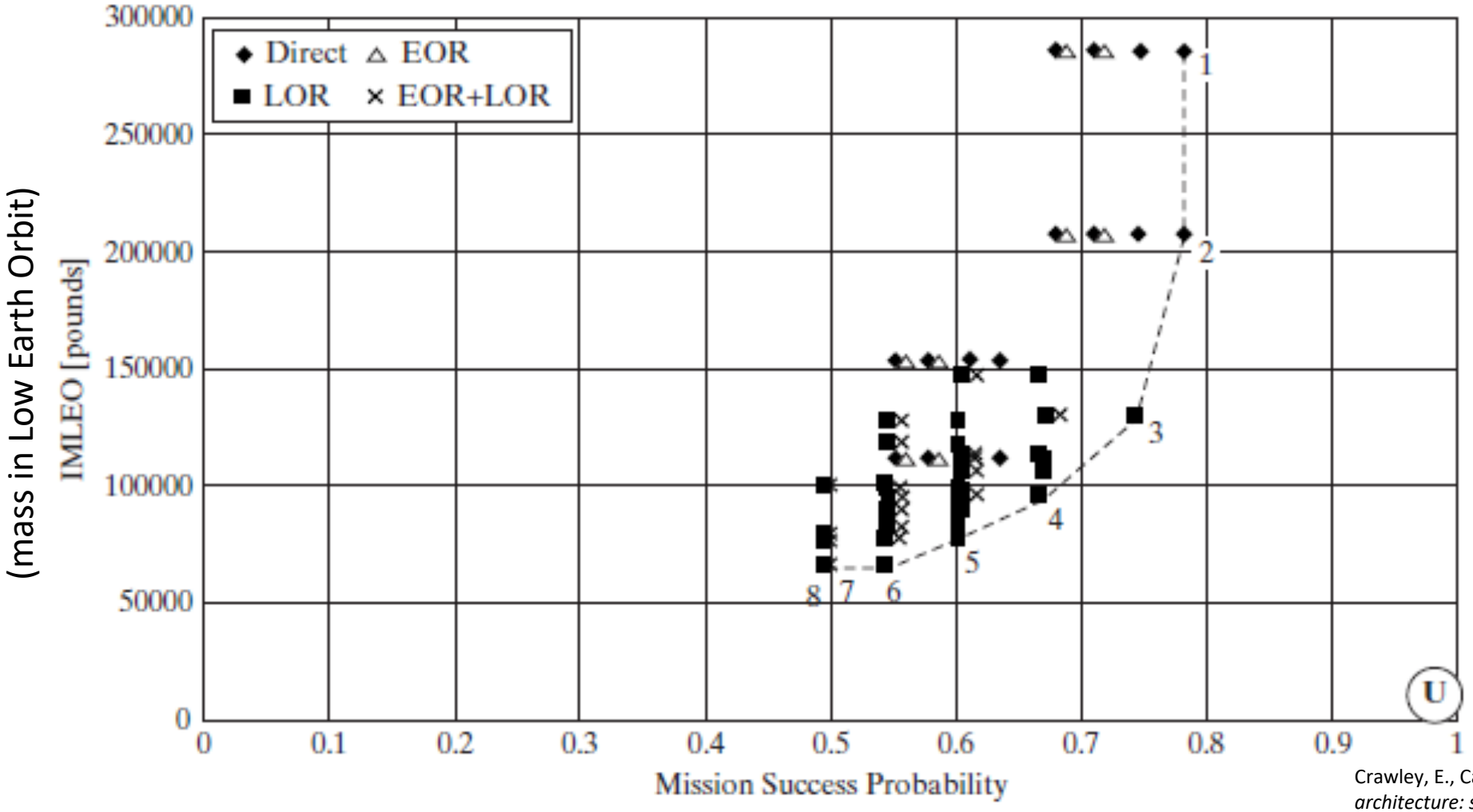
# Systems architecting as a decision-making process

| shortID | Decision | units | alt A | alt B | alt C | alt D |
|---|---|---|---|---|---|---|
| EOR | Earth Orbit Rendezvous | none | no | yes | | |
| earthLaunch | Earth Launch Type | none | orbit | direct | | |
| LOR | Lunar Orbit Rendezvous | none | no | yes | | |
| moonArrival | Arrival At Moon | none | orbit | direct | | |
| moonDeparture | Departure From Moon | none | orbit | direct | | |
| cmCrew | Command Module Crew | people | 2 | 3 | | |
| lmCrew | Lunar Module Crew | people | 0 | 1 | 2 | 3 |
| smFuel | Service Module Fuel | none | cryogenic | storable | | |
| lmFuel | Lunar Module Fuel | none | NA | cryogenic | storable | |



# Systems architecting as a decision-making process




Crawley, E., Cameron, B., & Selva, D. (2015). *System architecture: strategy and product development for complex systems*. Prentice Hall Press.

# Non-dominated solutions – The Pareto Frontier

## PARETO DOMINANCE AND FRONTIER

1. A vector function $\vec{f}(\vec{x}) = [f_1(\vec{x}), \ldots, f_k(\vec{x})]$ and
2. A feasible solution space $\Omega$

The MOP consists in to find a vector $\vec{x} \in \Omega$ that optimizes the vector function $\vec{f}(\vec{x})$ .

**Definition 2.** Pareto dominance. A vector $\vec{x}$ dominates $\vec{x}'$ (denoted by $\vec{x} \prec \vec{x}'$):

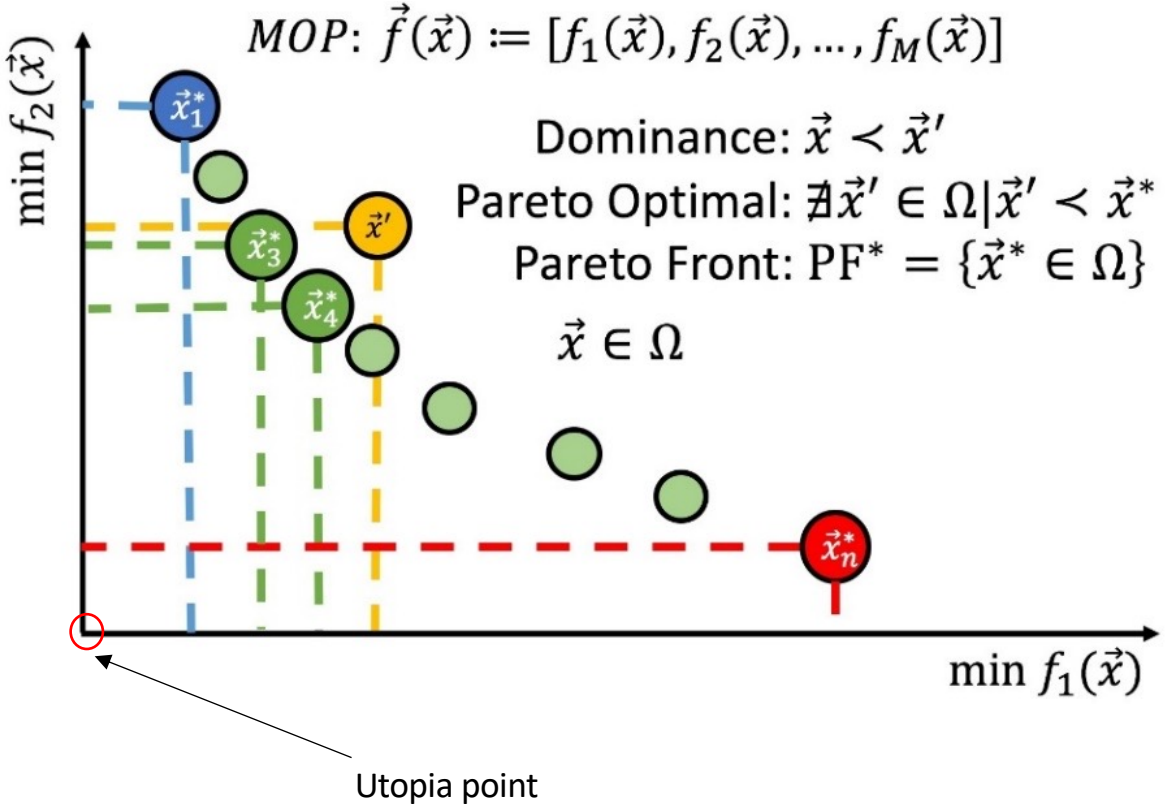
1. If $f_i \leq f_i(\vec{x}')$ for all $i$ functions in $\vec{f}$, and
2. There is at least one $i$ such that $f_i(\vec{x}) < f_i(\vec{x}')$ .

**Definition 3.** Pareto optimal. A vector $\vec{x}^*$ is Pareto optimal if does not exists a vector $\vec{x} \in \Omega$ such that $\vec{x}' \prec \vec{x}^*$ .

**Definition 4.** Pareto optimal set. The Pareto optimal set for a MOP is defined as:

$$P^* = \{\vec{x}^* \in \Omega\} \ .$$

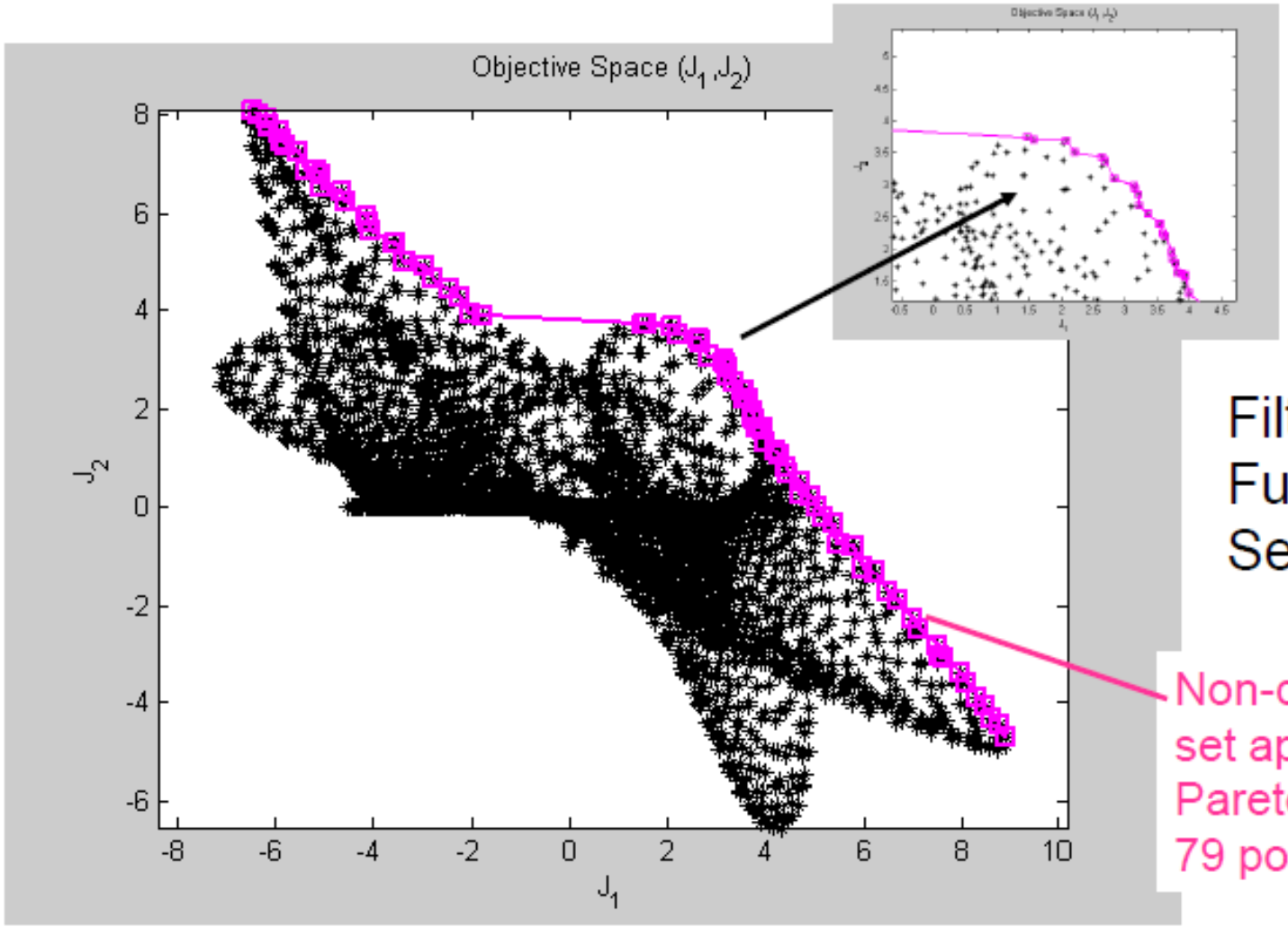
**Definition 5.** Pareto front. Given a MOP and its Pareto optimal set, the Pareto front is defined as $PF^* = \{f(\vec{x}) | \vec{x} \in P\}$ .




The set of dominating solutions and the Pareto optimal set are **not** the same!

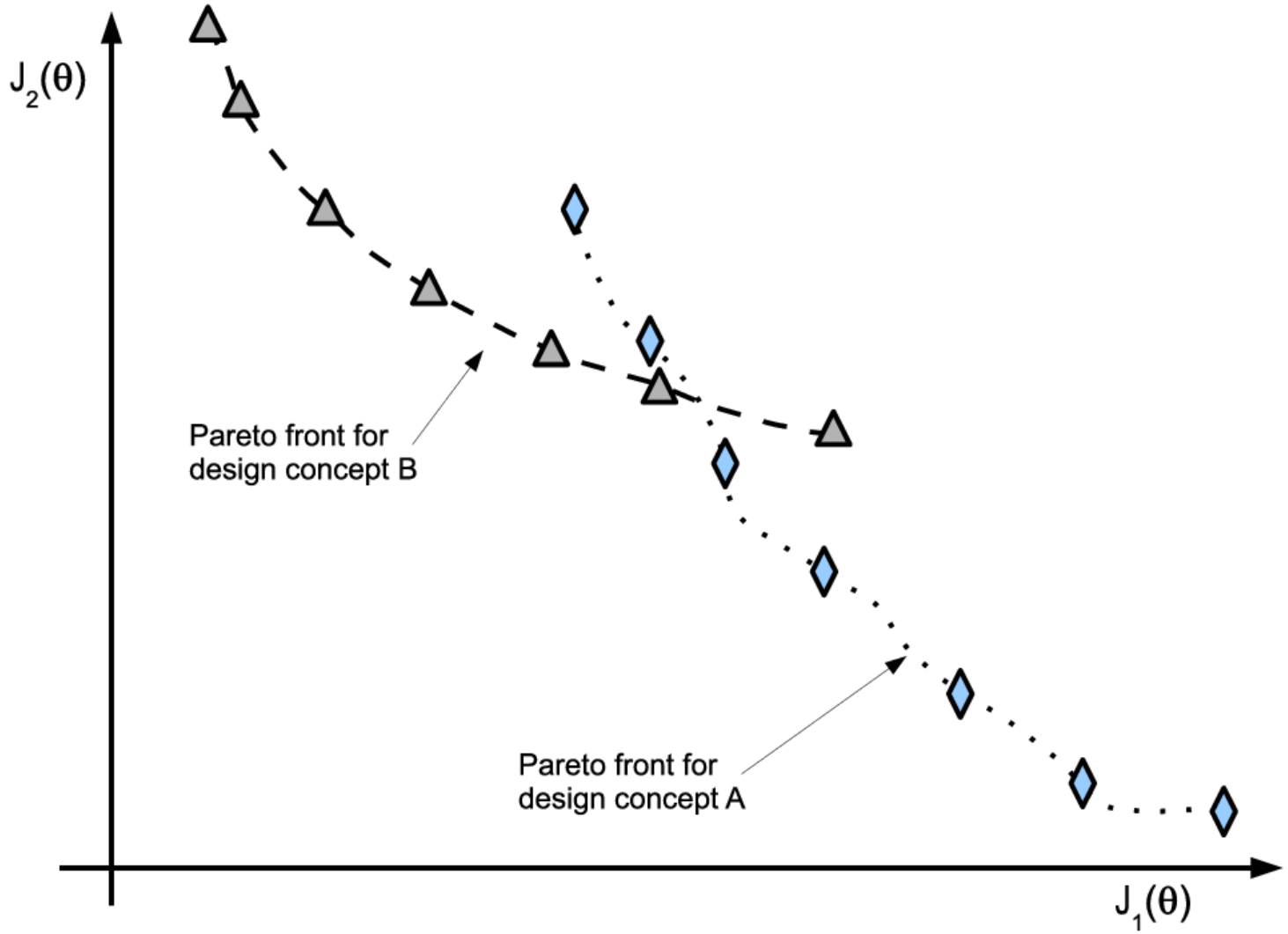
# Why use the Pareto frontier?

- Vast number of solutions → quickly focus on the „best“ solution
- Solutions on the Pareto frontier only small subset of all solutions



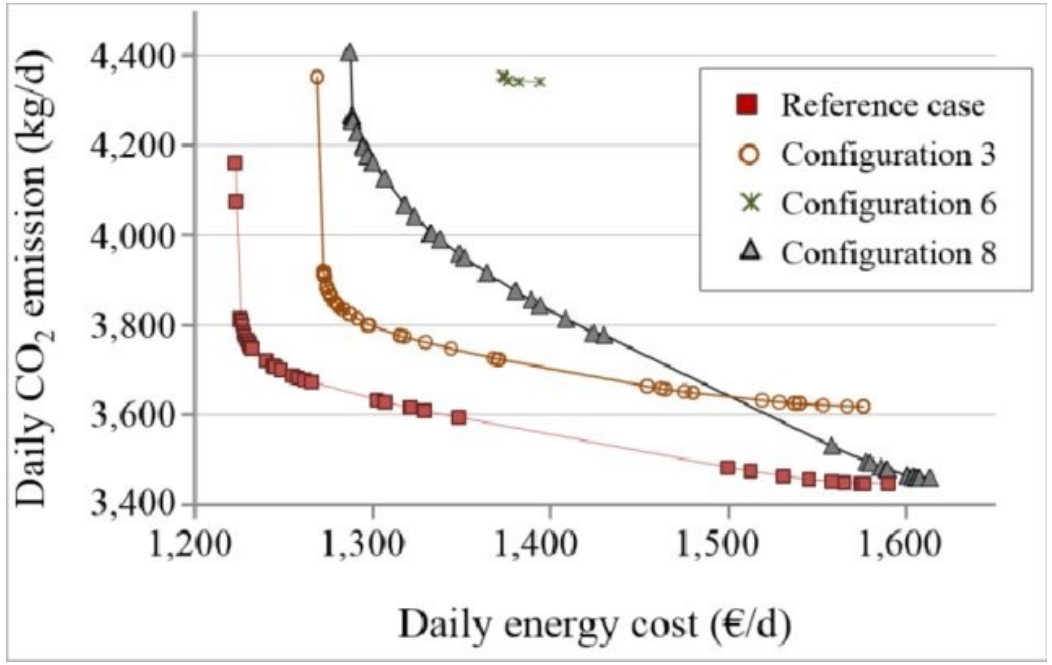
# What happens with multiple concepts? (Laptop vs. PC)

- Different concepts have different design parameter sets



# What happens with multiple concepts? (Energy system)

- Energy system example: Different configurations




Where is the Pareto frontier?
- For all configurations
- For configuration 3 and 8

Di Somma, M. (2016). OPTIMAL OPERATION PLANNING OF DISTRIBUTED ENERGY SYSTEMS THROUGH MULTI-OBJECTIVE APPROACH: A NEW SUSTAINABILITY-ORIENTED PATHWAY.

# Exercise: Determine a non-dominated solution

| | |
|---|---|
| max{range} | [km] |
| min{cost} | [$/km] |
| max{passengers} | [-] |
| max{speed} | [km/h] |

Multiobjective
Aircraft Design

| #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 |
|---|---|---|---|---|---|---|---|
| 7587 | 6695 | 3788 | 8108 | 5652 | 6777 | 5812 | 7432 |
| 321 | 211 | 308 | 278 | 223 | 355 | 401 | 208 |
| 112 | 345 | 450 | 88 | 212 | 90 | 185 | 208 |
| 950 | 820 | 750 | 999 | 812 | 901 | 788 | 790 |

⇨ Which designs are non-dominated ? (5 min)

# Method: Pairwise coparison

| #1 | | #2 | Score #1 | Score #2 |
|---|---|---|---|---|
| 7587 | > | 6695 | 1 | 0 |
| 321 | > | 211 | 0 | 1 |
| 112 | < | 345 | 0 | 1 |
| 950 | > | 820 | 1 | 0 |
| | | | 2 vs | 2 |

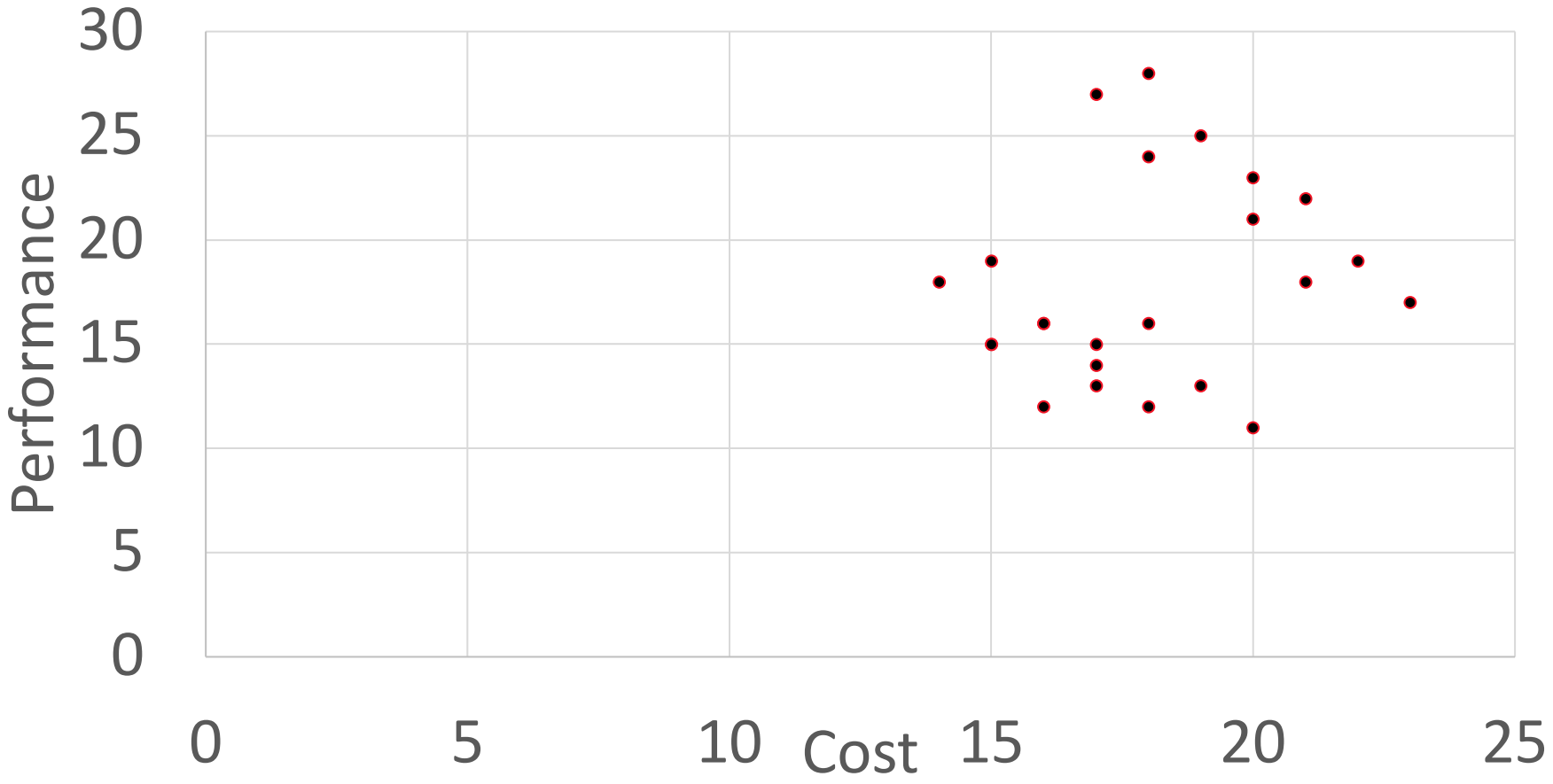
*Neither #1 nor #2 dominate each other*

| #1 | | #6 | Score #1 | Score #6 |
|---|---|---|---|---|
| 7587 | > | 6777 | 1 | 0 |
| 321 | < | 355 | 1 | 0 |
| 112 | > | 90 | 1 | 0 |
| 950 | > | 901 | 1 | 0 |
| | | | 4 vs | 0 |

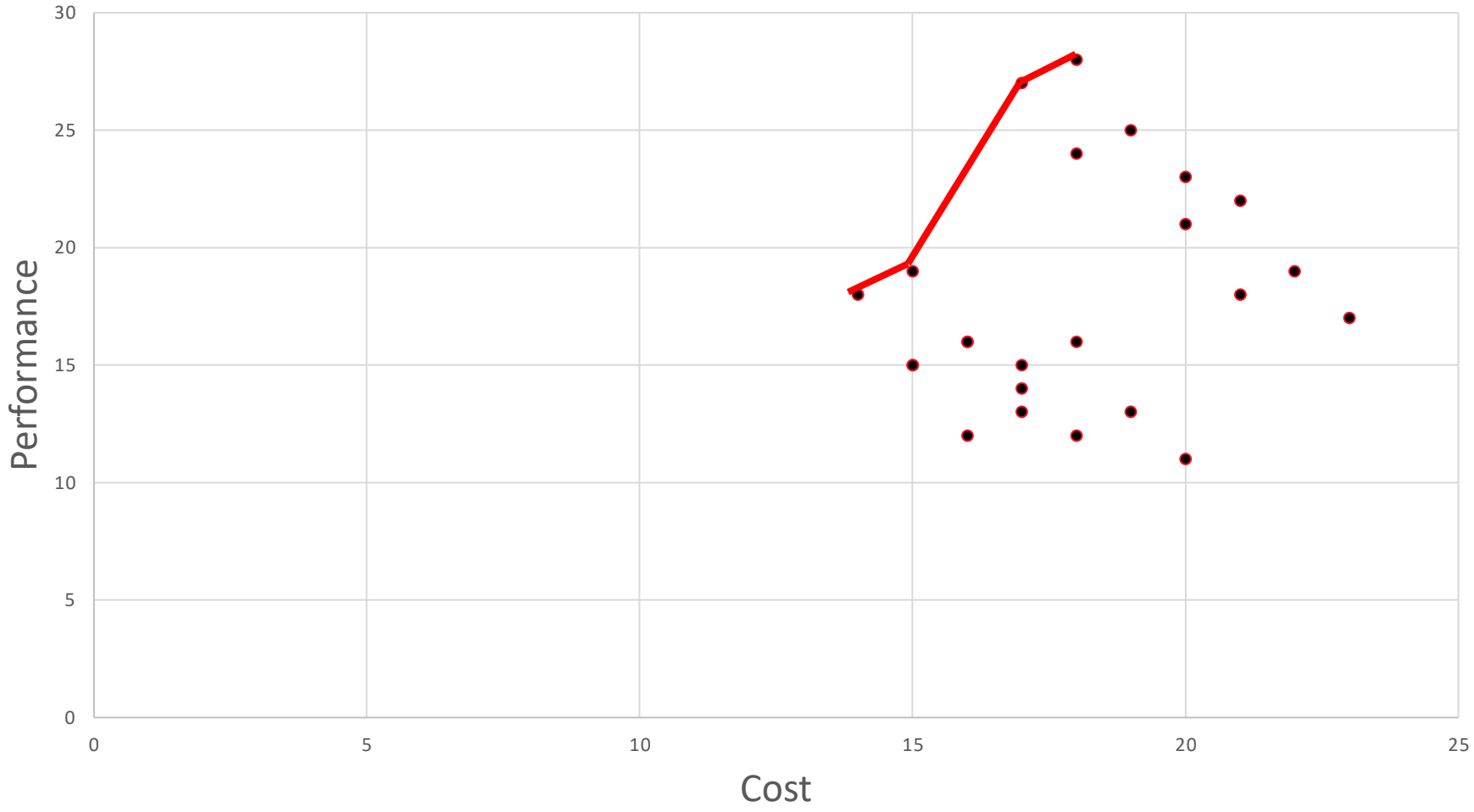
*Solution #1 dominates solution #6*

In order to be dominated a solution must have a"score" of 0 in pairwise comparison

# Exercise: Which of these solutions is Pareto optimal?



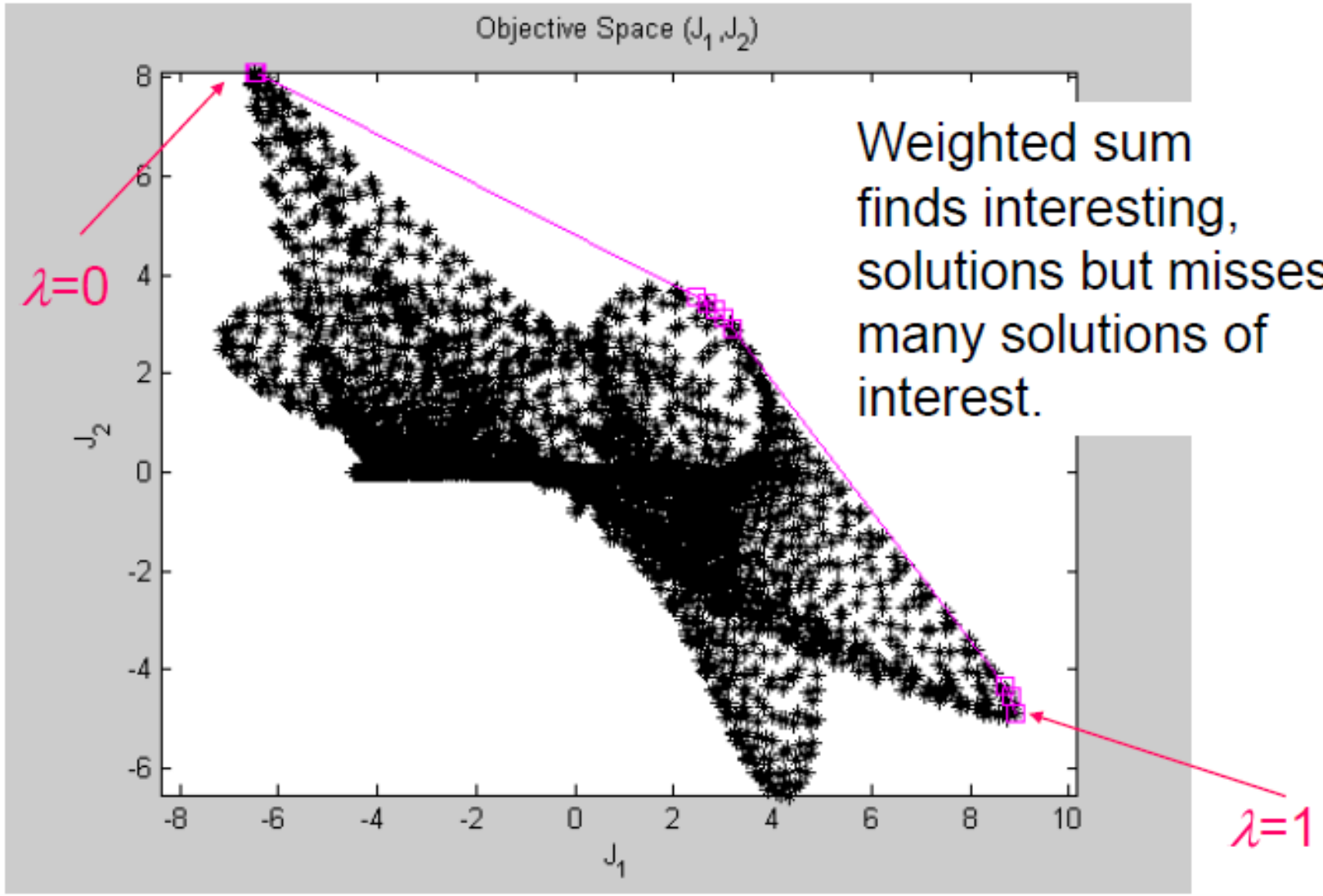
# Need to learn how to READ!



“

# What about decision methods?

# Pareto frontier vs. Weighted sum




- The weights are varied between 0 and 1. You can see that only a few Pareto optimal solutions are „discovered“.
- For weight 0, it finds the best solution for just one criteria (here: J2). For 1, it finds the best solution for J1.

# 05

## Product line (platforming)

# Platforming: what is about?

HOW MANY DIFFERENT CAMERA I NEED TO TAKE ALL THESE PICTURES?

**PLATFORMING IS ALL ABOUT MINIMISING THE DIVERSITY OF SOLUTIONS TO COVER SEVERAL NEEDS**


I WOULD LOVE TO USE ONLY ONE BUT…
I DON'T WANT TO HAVE A BAD IMAGE QUALITY

# A set of solutions to cover the whole needs

## EX: THE CANON FAMILY

EOS family lead to different trade-off, as represented, for instance, in the beside graph.

The variety of solutions is then obtain by joining a body with a lens.

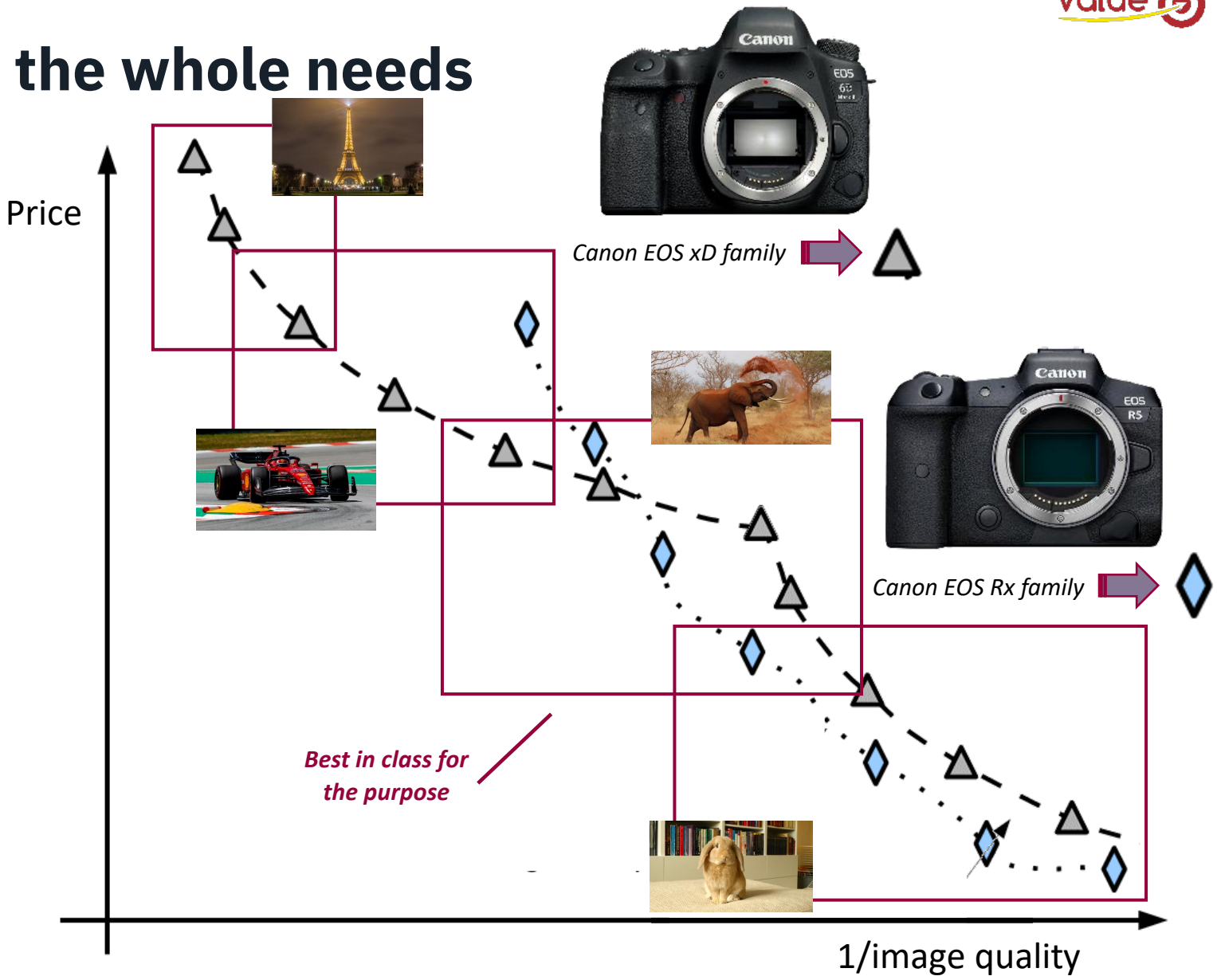
# A set of solutions to cover the whole needs

## EX: THE CANON FAMILY

EOS family lead to different trade-off, as represented, for instance, in the beside graph.

The variety of solutions is then obtain by joining a body with a lens.

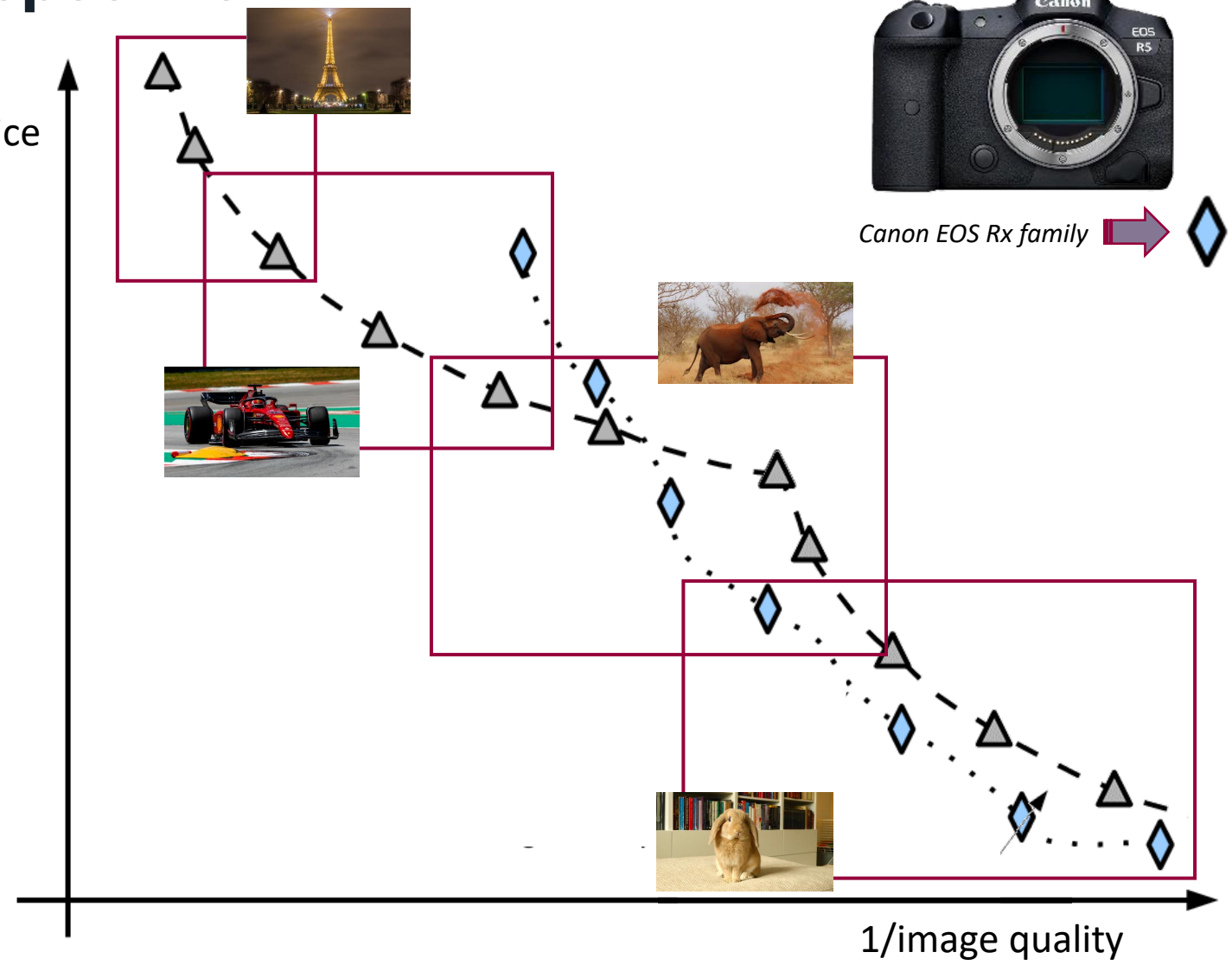
Following the chosen body and lens, capacity of the solution may or may not be able to cover expectations



# Platforming, a decision perspective

## EX: THE CANON FAMILY

If you choose the **EOS Rx family**, you'll be able to do, according to the beside graph, very nice photos of rabbit and elephant, poor photos of Formula 1 car and not photo of the Eiffel tower by night.



# Platforming, a decision perspective

## EX: THE CANON FAMILY

If you choose the **EOS xD family**, you'll be able to do, according to the beside graph, very nice photos of the Eiffel tower by night and Formula 1 car, as well as good photos of elephant and rabbit.


*Canon EOS xD family*




As conclusion:

| | *Best in case* |
|---|---|
| · To take "by night" picture | EOS xD |
| · To take "race car" picture | EOS xD |
| · To take "elephant" picture | EOS Rx |
| · To take "home animal" picture | EOS Rx |

A trade must be done…

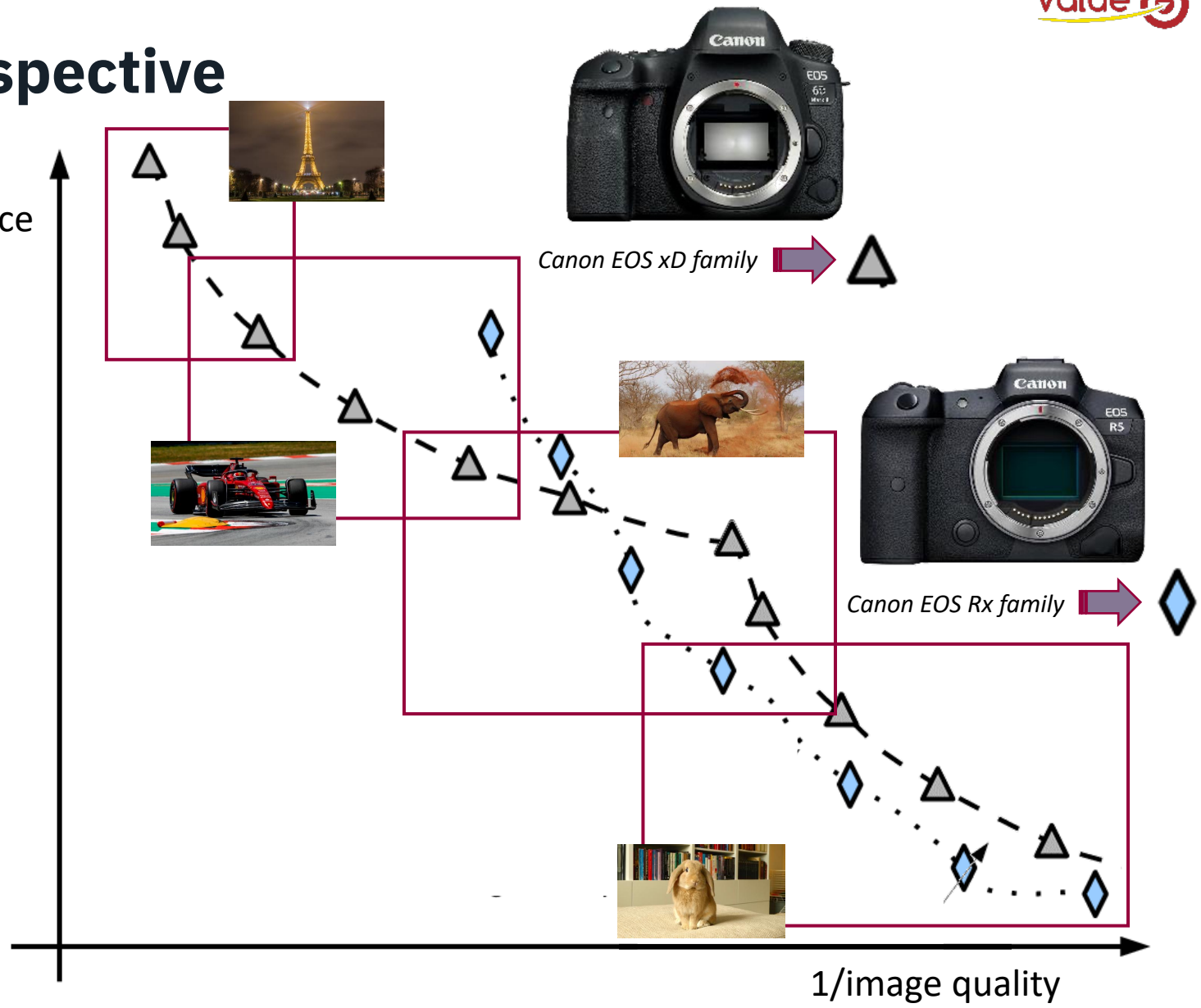
# Platforming, a decision perspective

## EX: THE CANON FAMILY

So how to rationalize the decision to take?

Introducing the ”DOL”: Degree of Liberty

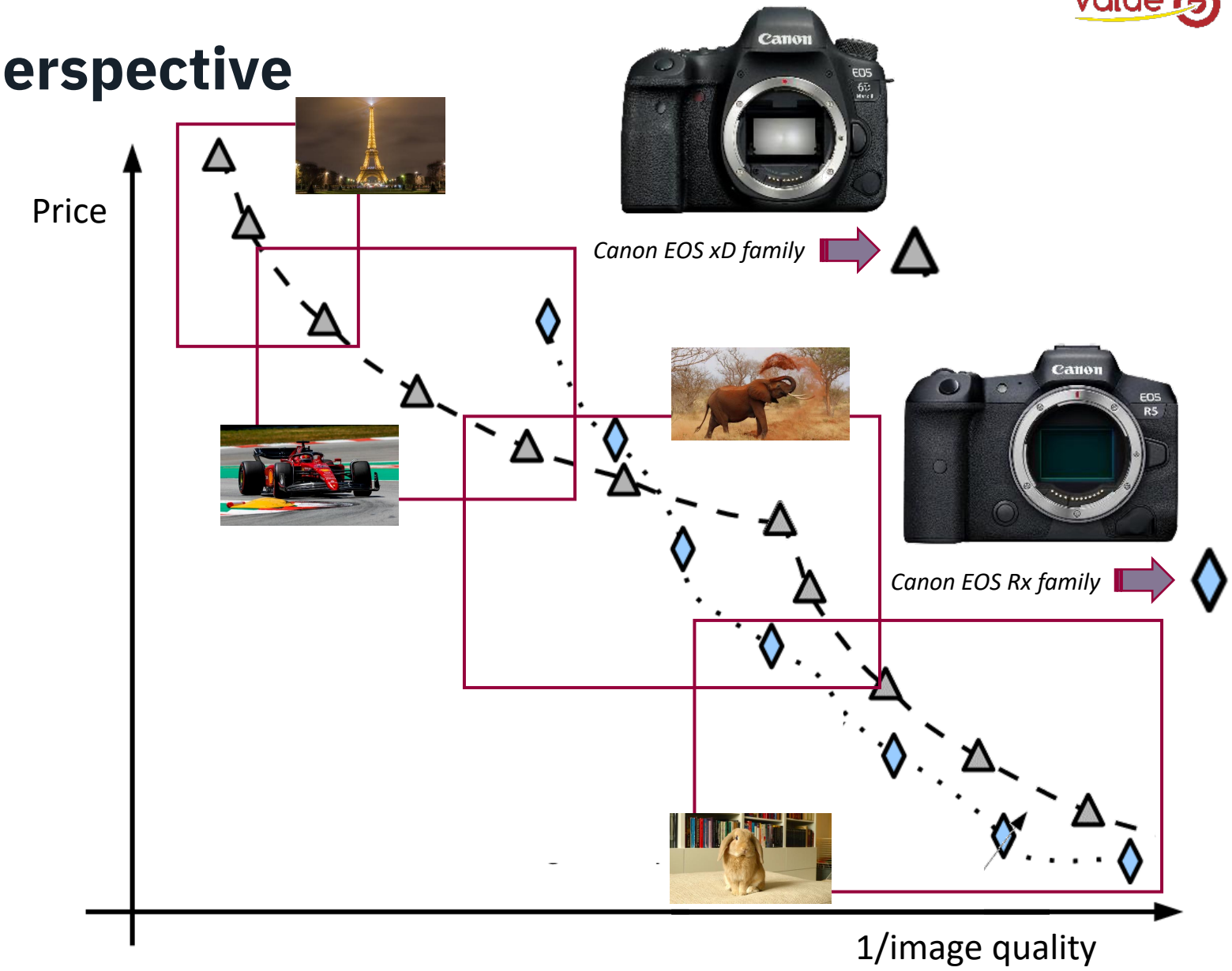
- The DOL is the liberty you might let, on sub-system(s), to find the best solutions. For instance:
  - DOL = 0 means you chose the best combo of one body and one lens to fulfil all needs. So, for sure, you will not be optimal in each expectation
  - DOL = 1 means you chose the best combo of combo : you chose what ever two body and one lens or two lens and one body to maximize the satisfaction whatever the need you need to cover



# Platforming, a decision perspective

## EX: THE CANON FAMILY

*Kind reminder: each solution into both curves are pareto solutions for each architecture.*



# Platforming, a decision perspective

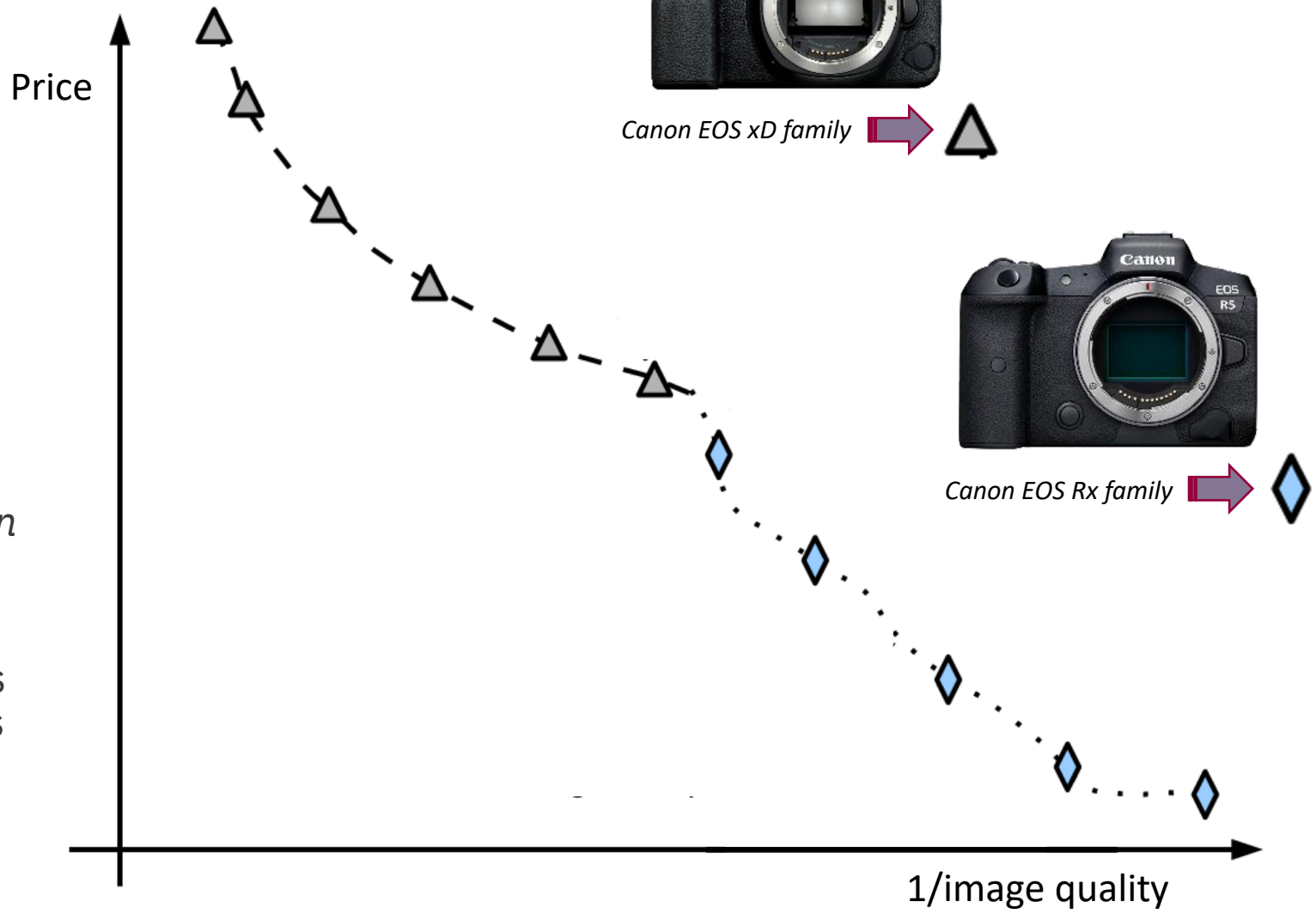
## EX: THE CANON FAMILY

*Kind reminder: each solution into both curves are pareto solutions for each architecture.*

The retained pareto curve is as that (see beside).

*Thanks to a systematic exploration: we can say that it's impossible to find better solutions than these ones.*

Considering that each remaining concepts are optimal, each of them can be noted as 100%



# Platforming, a decision perspective

## EX: THE CANON FAMILY

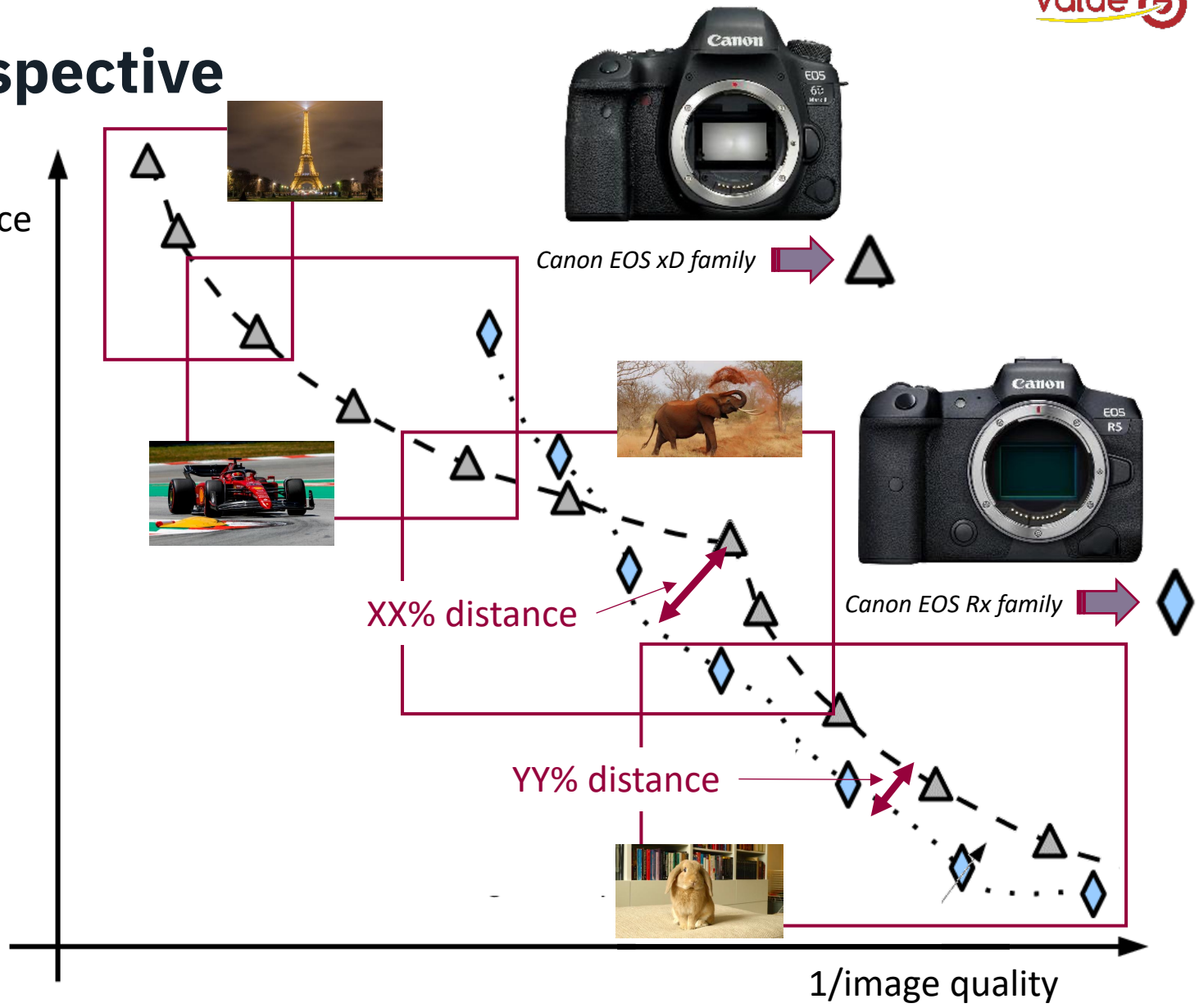
Back to our initial Design space and considering our decision scope has to chose only the body (DOLmax = 1 (2 body alternatives - 1 than must be, in any case chosen)

For DOL = 1, we can be, every time, optimal

For DOL = 0, the EOS xD is the only one that lead to cover the whole need:

- 100% performance to take “by night” picture
- 100% performance to take “race car” picture
- XX%<100% performance take “elephant” picture
- YY% <100% performance to take “home animal” picture

*XX% and YY% are calculated by analyzing the distance between the best solution find using the combo of combo and the optimal solution (at 100%)*



27/03/2023

# Thank you for you attention

Any questions?



System Value