Co-design of Complex Systems: From Embodied Intelligence to Mobility Systems

Gioele Zardini

Institute for Dynamic Systems and Control
ETH Zürich

ETH zürich

Swiss National Science Foundation

NCCR Automation

gzardini@ethz.ch - http://gioele.science

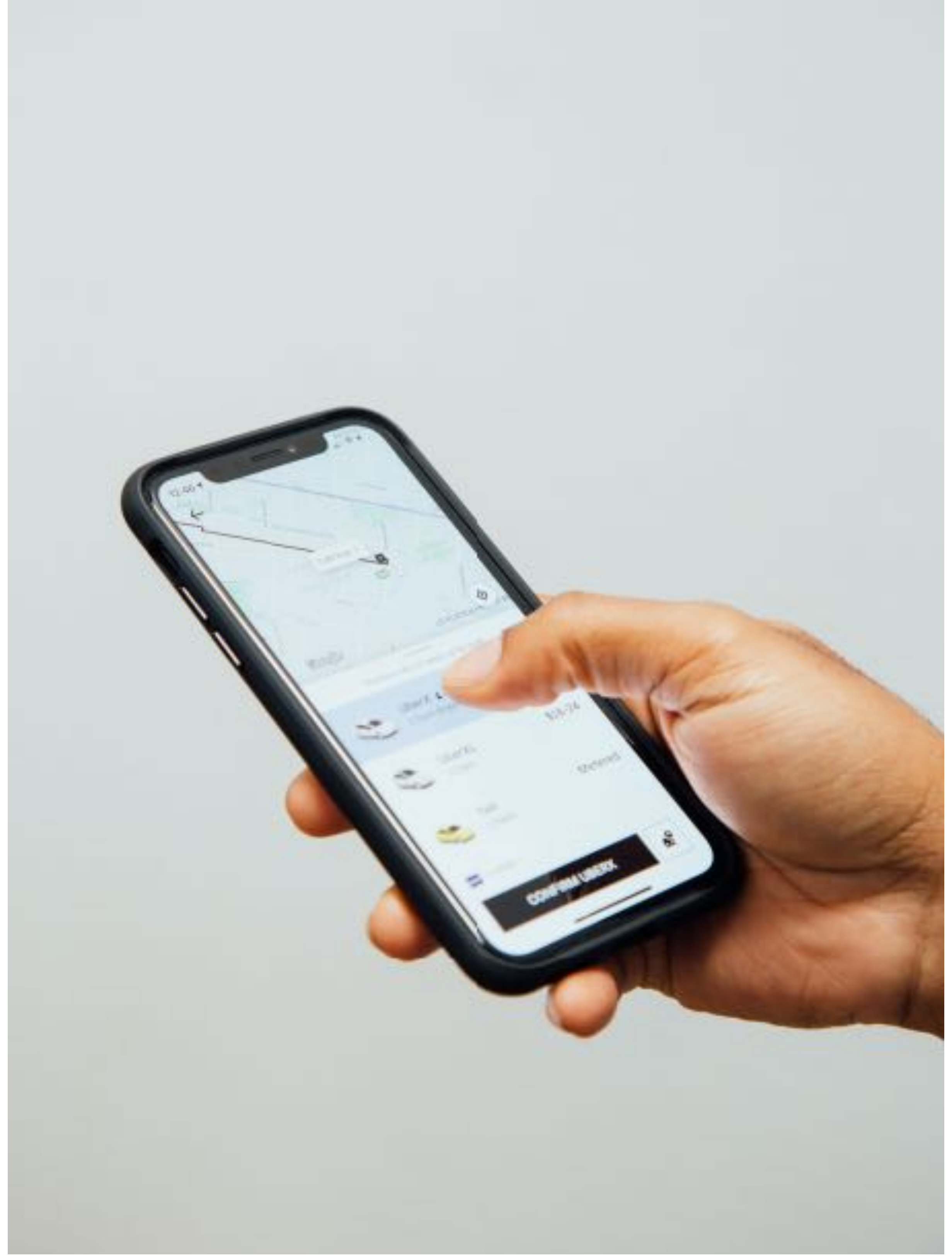
Introductions

- Ph.D. Candidate at ETH Zürich
- Studied at ETH Zurich, spent time in U.S. (California and Massachusetts)
- Lived and worked in Singapore
- Research efforts:

Co-design ***Game theory*** ***Applied Category Theory***

Mobility ***Embodied intelligence***

Mobility systems are under pressure



Travel demand is increasing and **travel needs** are changing

55% of the population resides in cities. By 2050, the proportion is expected to reach 68%

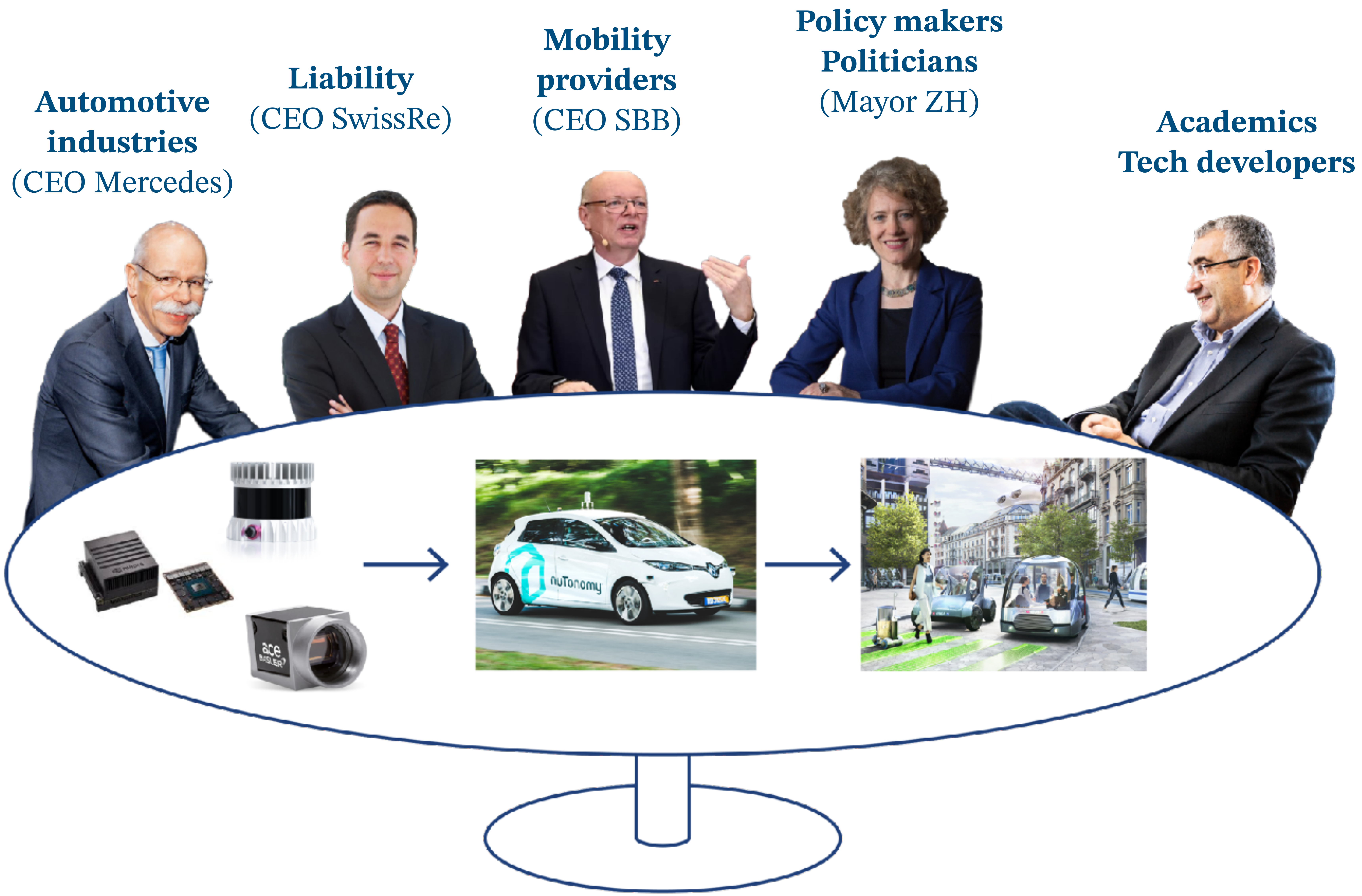
The rise of **private mobility service providers** calls for **service design** and new **regulation schemes**

Ride-hailing has increased by 1,000% in NYC from 2012 to 2019

Transportation systems need to meet global **sustainability goals**

Cities are responsible for 60% of greenhouse emissions, 30% of which produced by transportation (in US)

Mobility systems are very complex socio-technical systems



Complexity due to many interconnected components

An autonomous vehicle = hardware, software, behavior, coordination, actuation, sensing, localization, planning, social acceptance, interaction, control, computation, perception, mapping, learning, liability, energetics, communication, regulations

So many **components** (hardware, software, ...), so many **choices** to make!

Nobody can understand the **whole** thing!

We forget why we made some **choices**, and we are afraid to make **changes**...

These "computer" thingies are not helping us that much for design...

anthropomorphization of 21st century engineering malaise →

"My dear, it's simple: you lack a proper theory of **co-design**!"

Co-design across fields and scales

"Your system is just a component in another person's system"

City level

Optimal infrastructure choices

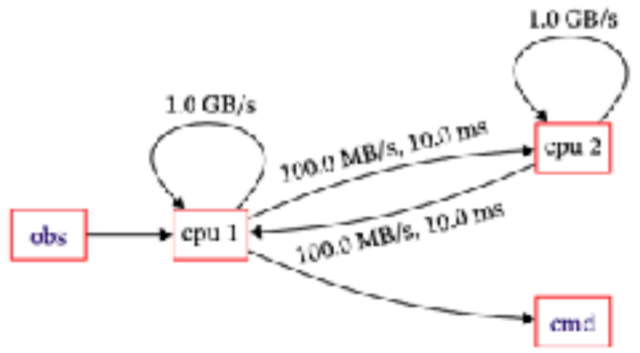
Service level

Optimal deployment

Platform level

Optimal sensor and control choice

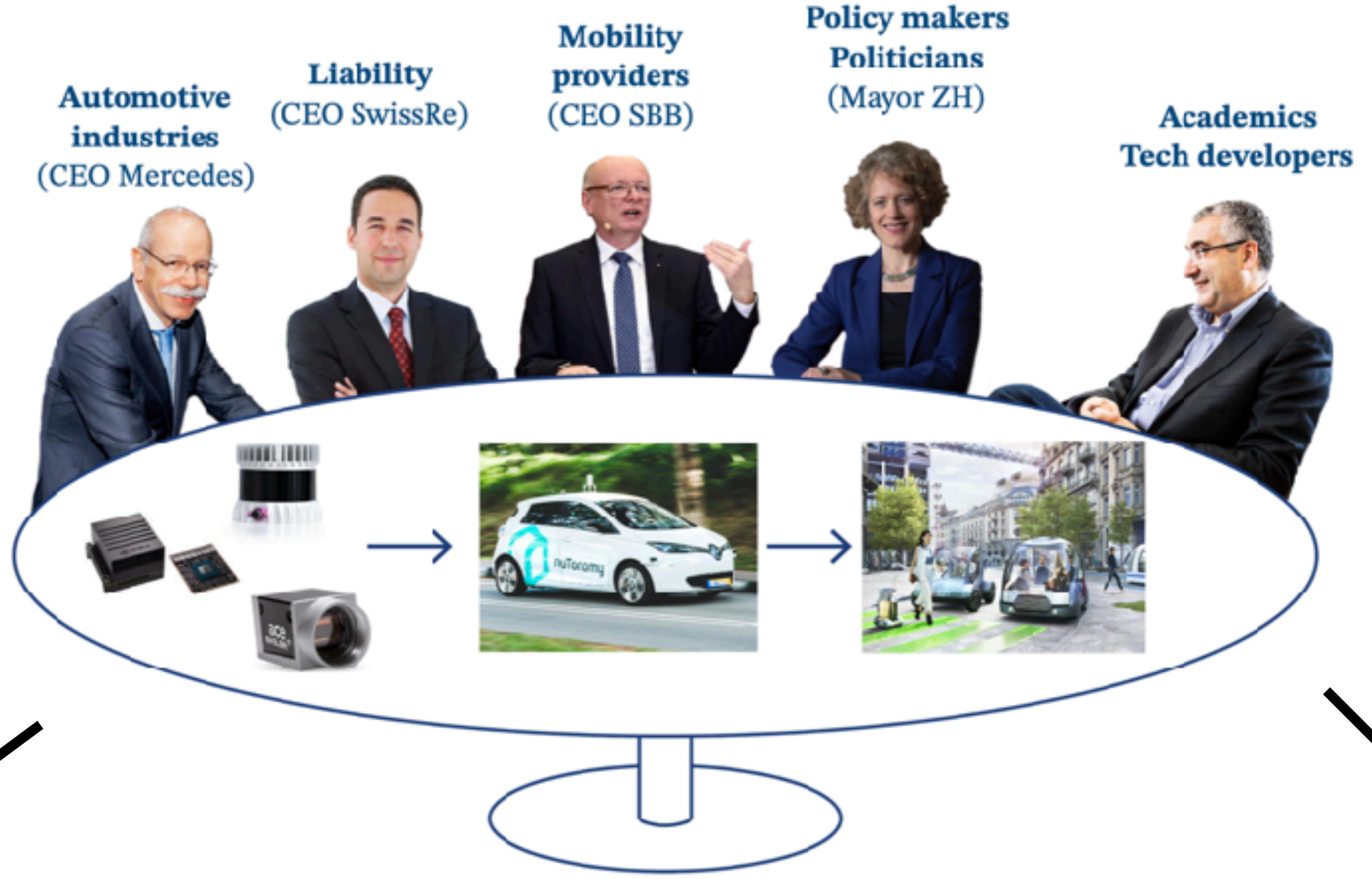
Subsystem level



Optimal resource allocation

We leverage co-design and game theory to solve complex socio-technical problems

Complex socio-technical system



Large interconnected system

↓

Mathematical theory of co-design
applied category theory

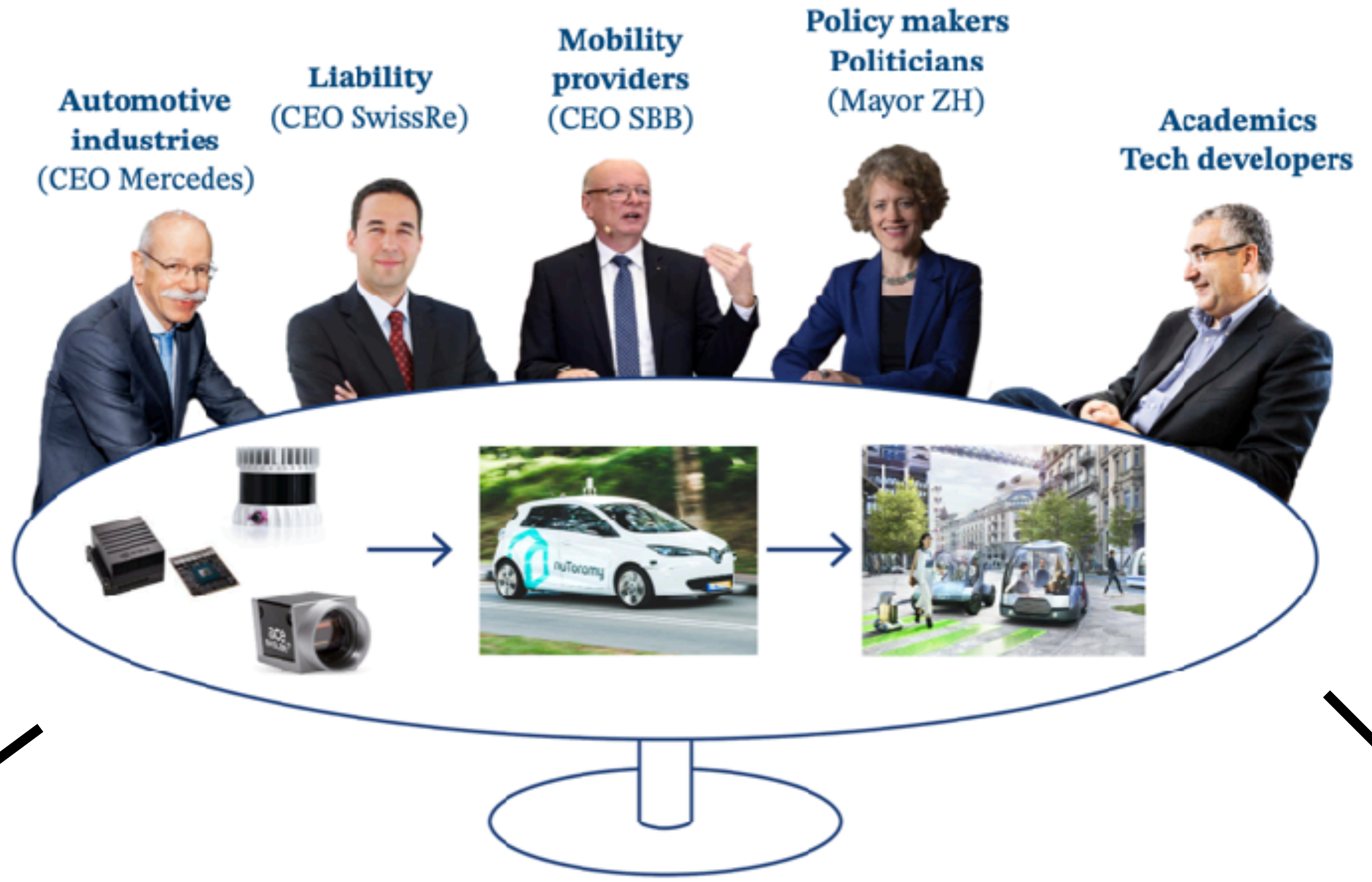
Many agents, many (often conflicting) interests

↓

Game theory

We leverage co-design and game theory to solve complex socio-technical problems

Complex socio-technical system



Large interconnected system

↓

Mathematical theory of co-design
applied category theory

Many agents, many (often conflicting) interests

↓

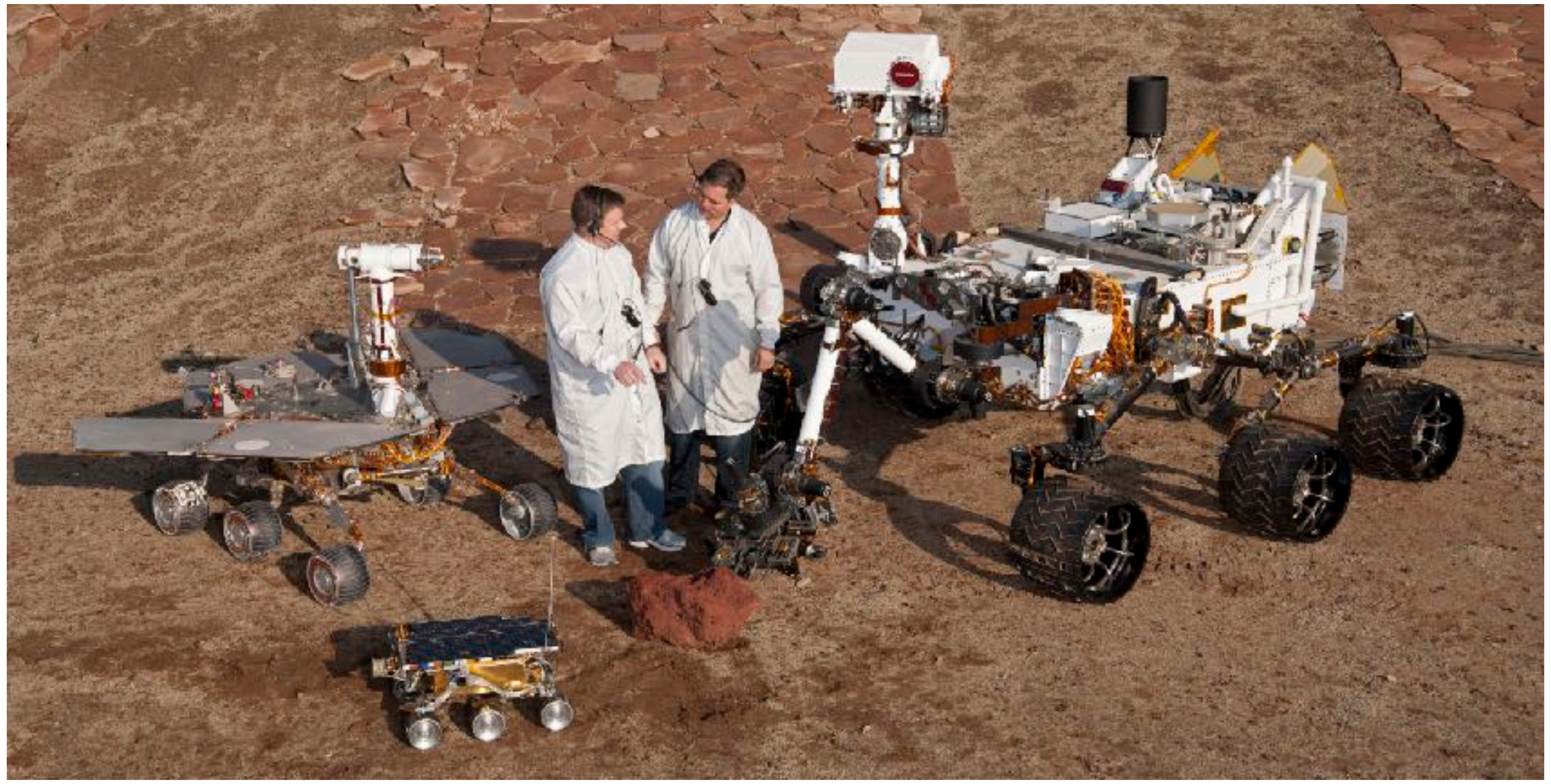
Game theory

What is co-design?

minimize
(resources usage)

subject to
(functionality constraints)

task specification → **expert knowledge** → **catalogue of parts** → "automated designer" → **optimal design(s)**

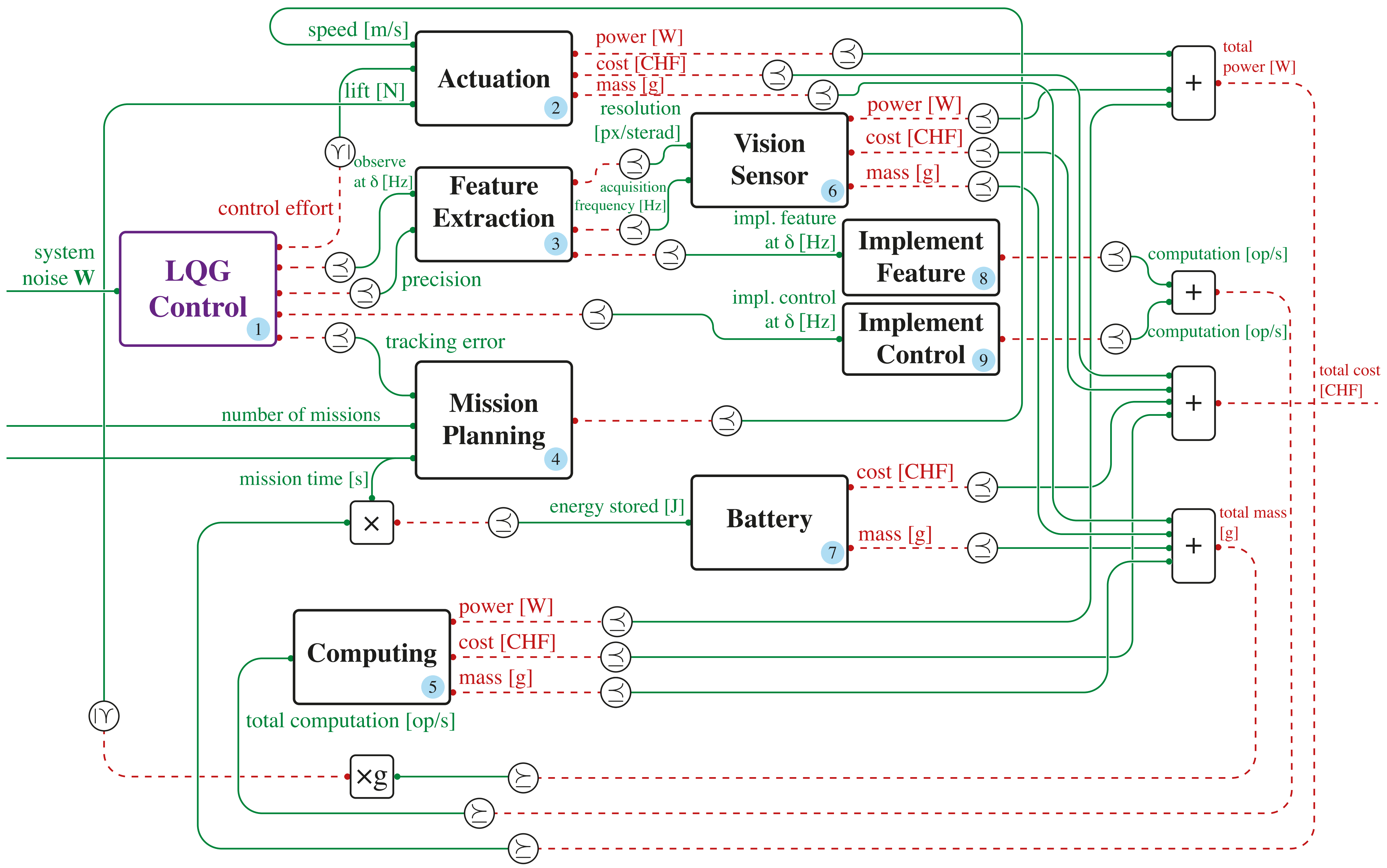


A new approach to “co”-design

- A new approach to <u>**co**</u>**llaborative**, <u>**co**</u>**mputational**, <u>**co**</u>**mpositional**, <u>**co**</u>**ntinuous** design. designed to work **across fields** and **across scales**.

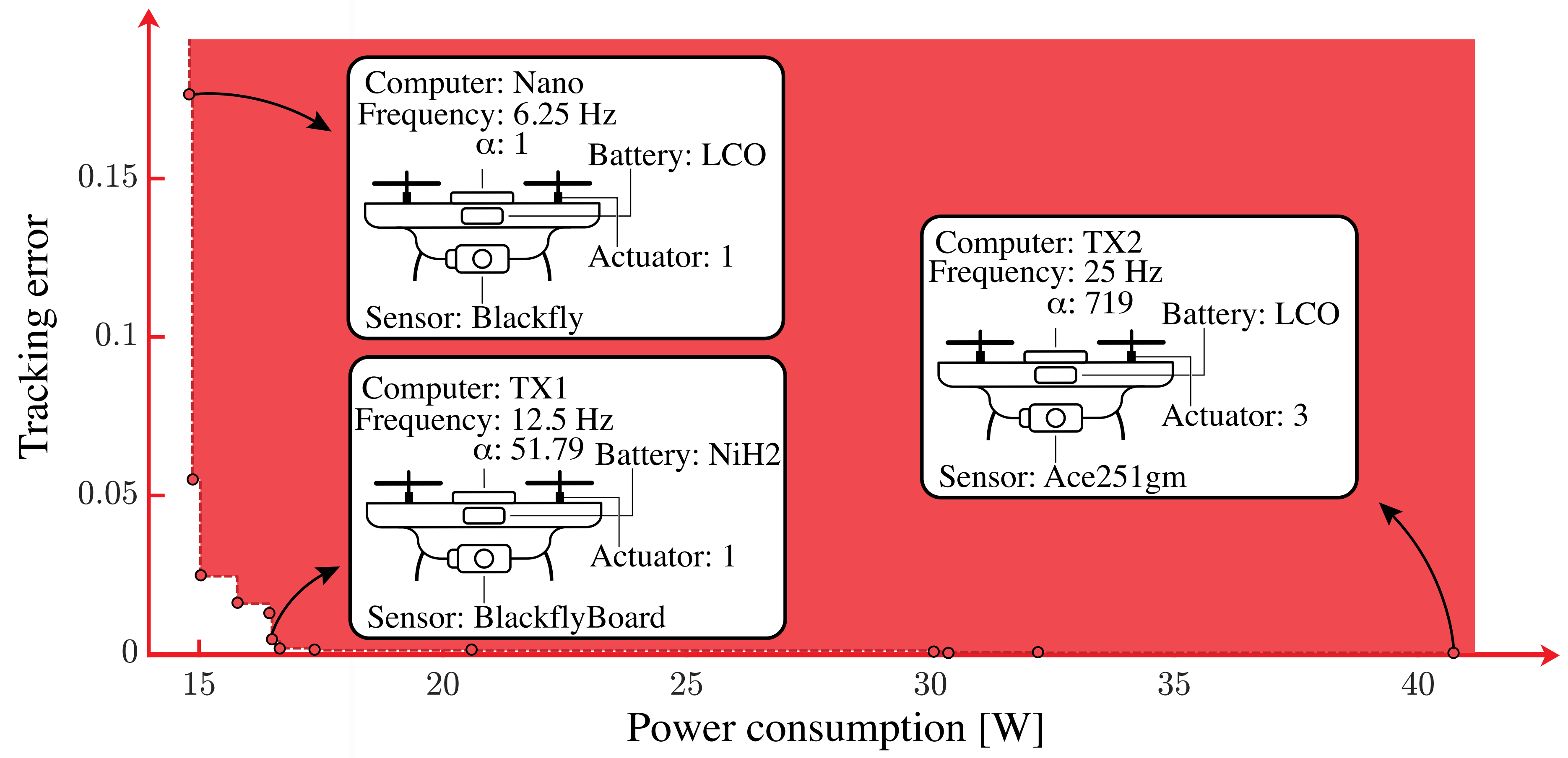
- Intended learning outcomes:
  - Defining **“design problems”** for **components** (“functionality”, “resources”).
  - Modeling **co-design constraints** in a complex **system**.
  - Efficient solution to design queries.

“Co-design diagram”



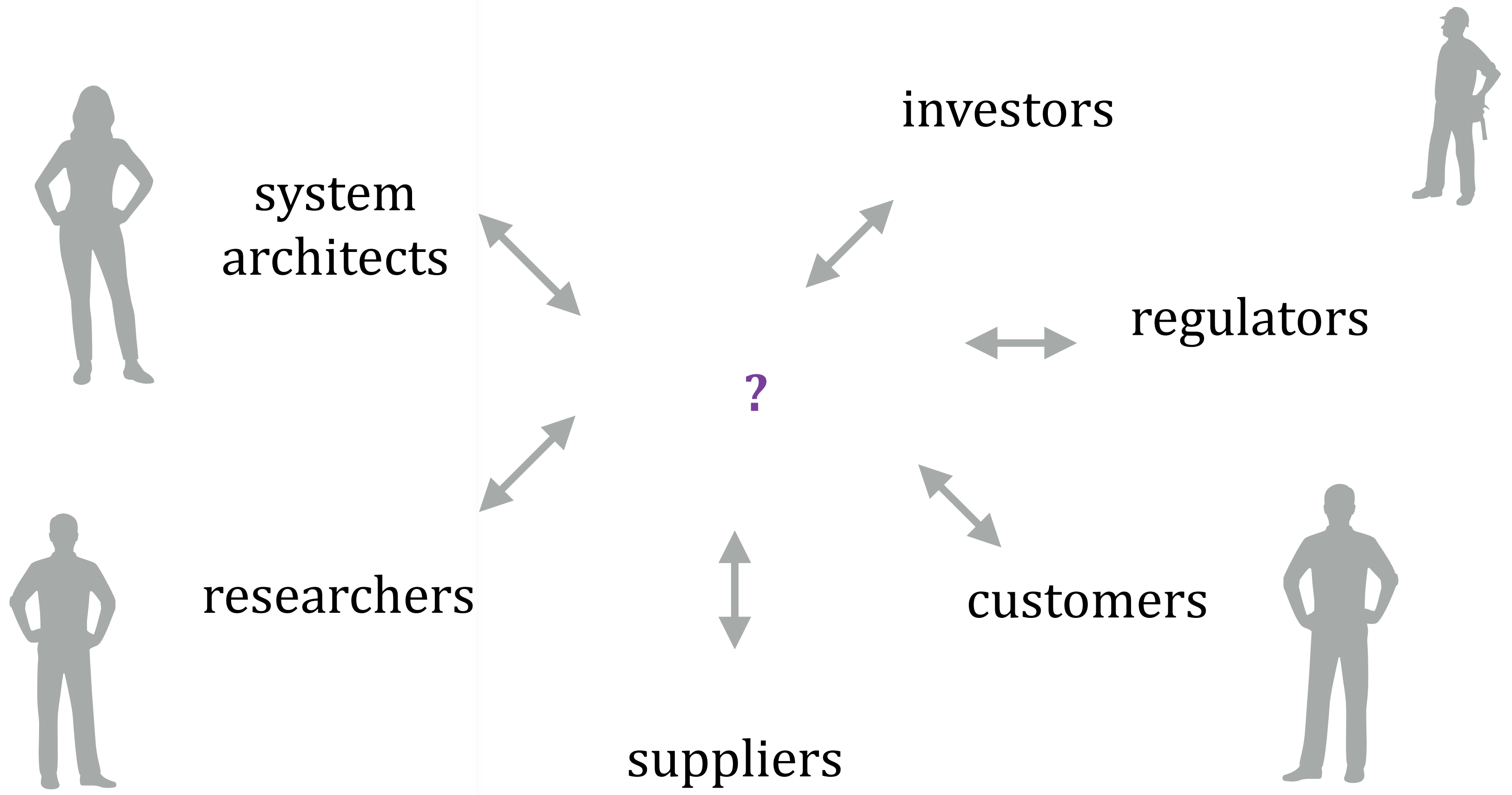
optimization

Pareto front of optimal designs



“Co”-design desiderata

- **Computational**
  - Let the machine help us!

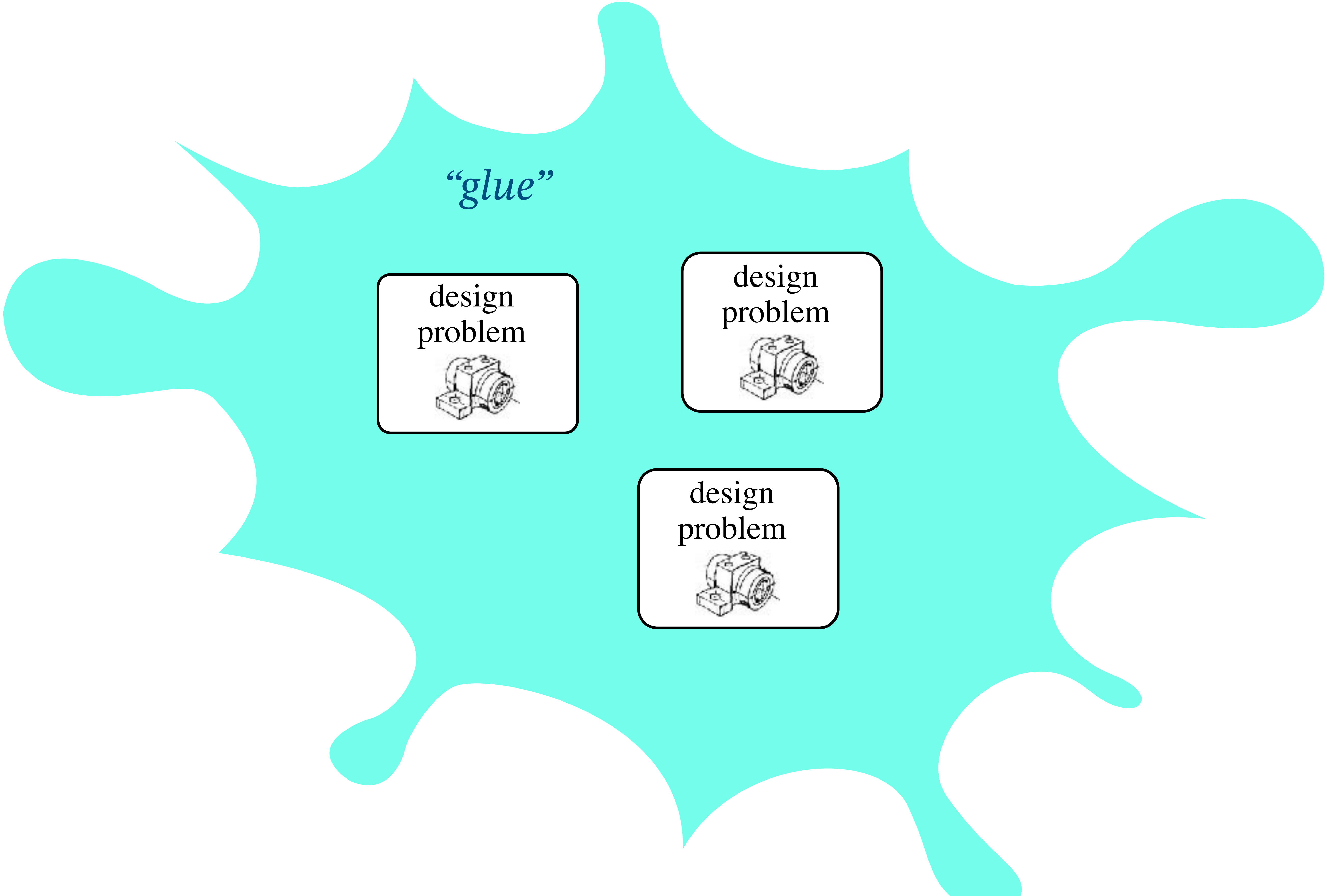


- **Collaborative**
  - Pooling knowledge from experts across fields.
- **Continuous**
  - Design is not static: it should be reactive to changes in goals and contexts.
- **Compositional**
  - My system is a component of somebody else’s system.

The modeling challenge

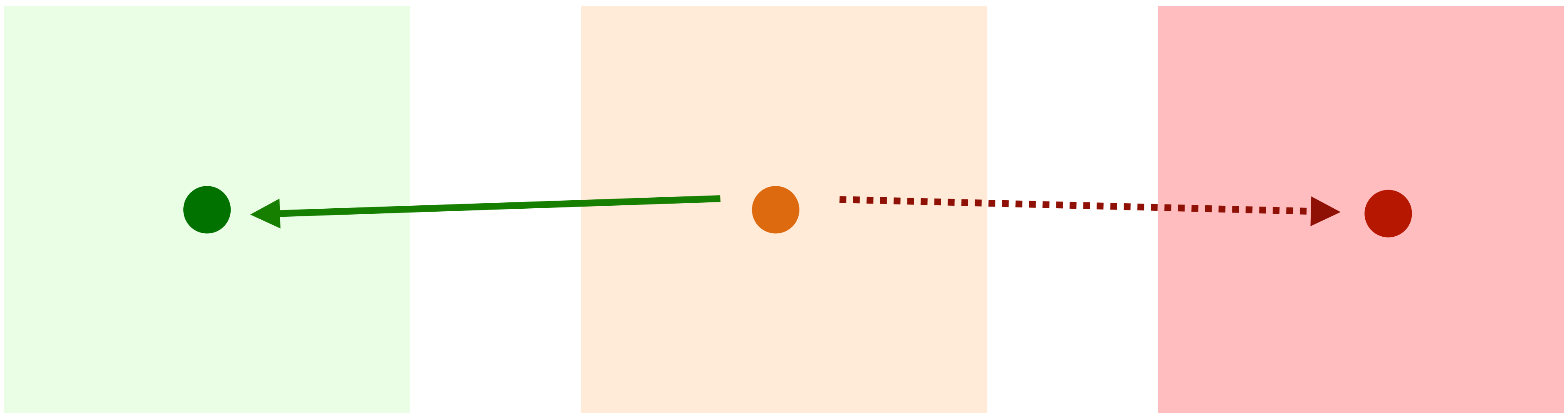
- How to find a **"theory of everything" across fields...**
  ... that is still **computationally** and **intellectually tractable?**

- **Approach: focus on the interactions** (co-design constraints).



An abstract view of design problems

- Across fields, design or synthesis problems are defined with 3 spaces:
  - **implementation space**: the options we can choose from;
  - **functionality space**: what we need to provide/achieve;
  - **requirements/costs space**: the resources we need to have available;



functionality
(provided)

desired behavior

specifications

objectives

guarantees

"function"

implementations

choices

plans

blueprints

decision variables

"form"

costs, resources
(required)

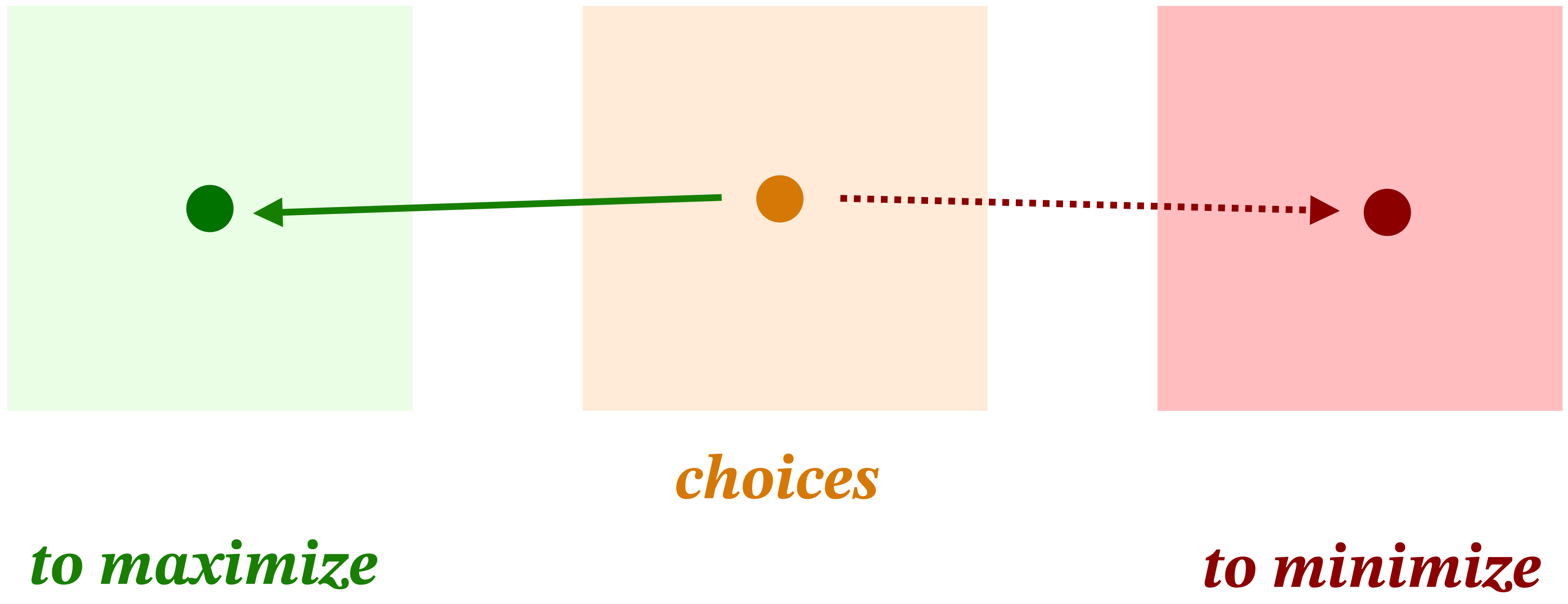
requirements

dependencies

"function"

An abstract view of design problems

- Across fields, design or synthesis problems are defined with 3 spaces:
  - **implementation space**: the options we can choose from;
  - **functionality space**: what we need to provide/achieve;
  - **requirements/costs space**: the resources we need to have available;



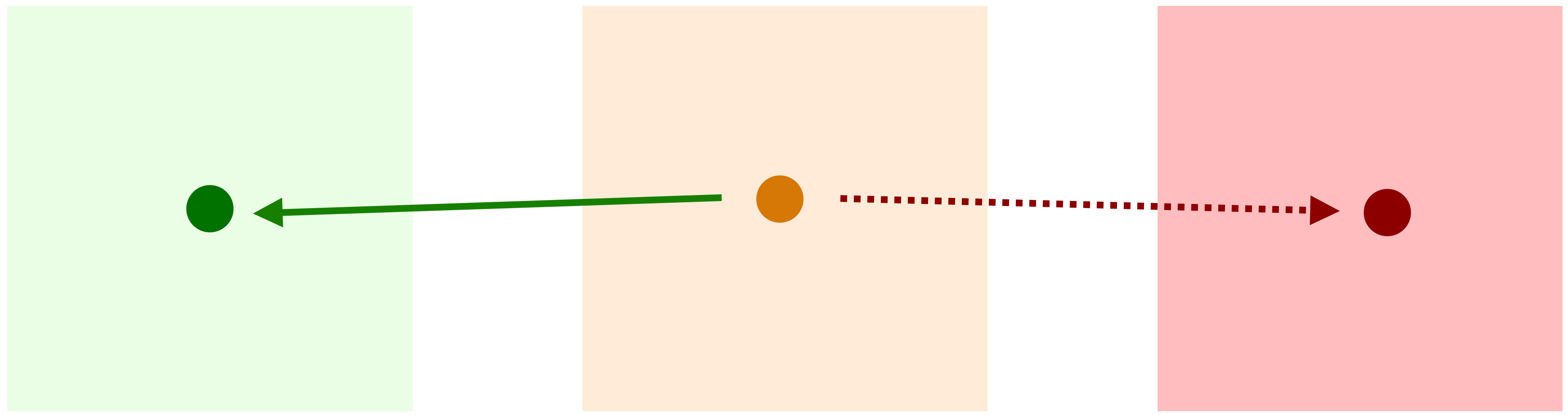
Partially ordered sets with a bottom

$\langle \mathbf{F}, \preceq_{\mathbf{F}} \rangle$

$\langle \mathbf{R}, \preceq_{\mathbf{R}} \rangle$

An abstract view of design problems

- Across fields, design or synthesis problems are defined with 3 spaces:
  - **implementation space**: the options we can choose from;
  - **functionality space**: what we need to provide/achieve;
  - **requirements/costs space**: the resources we need to have available;



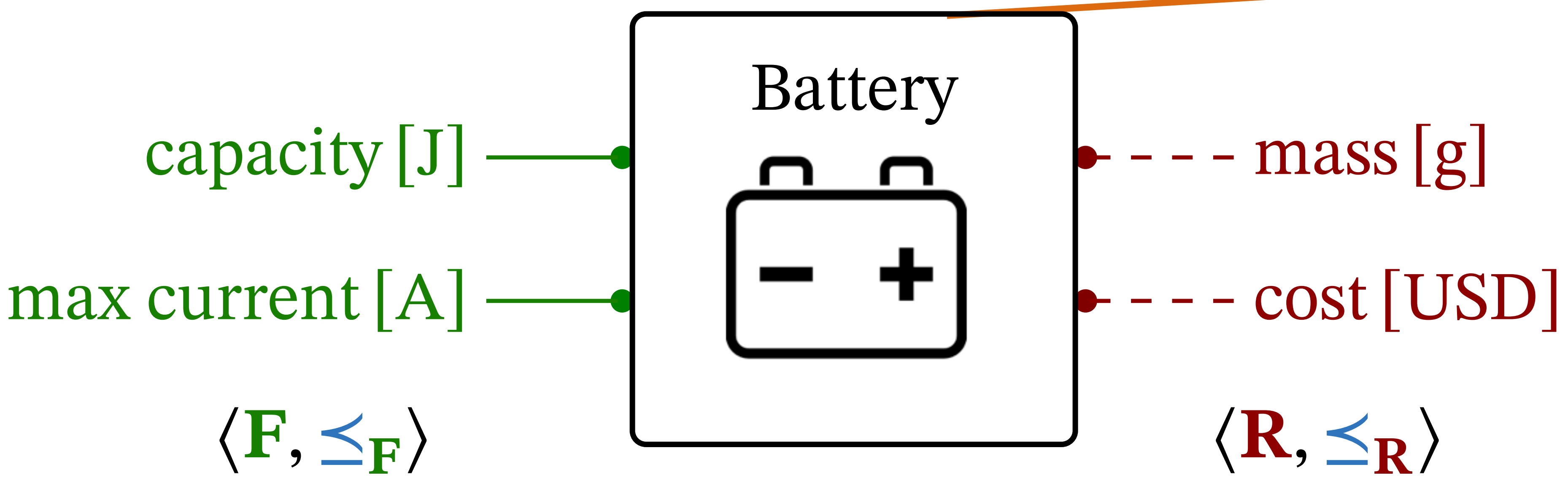
choices

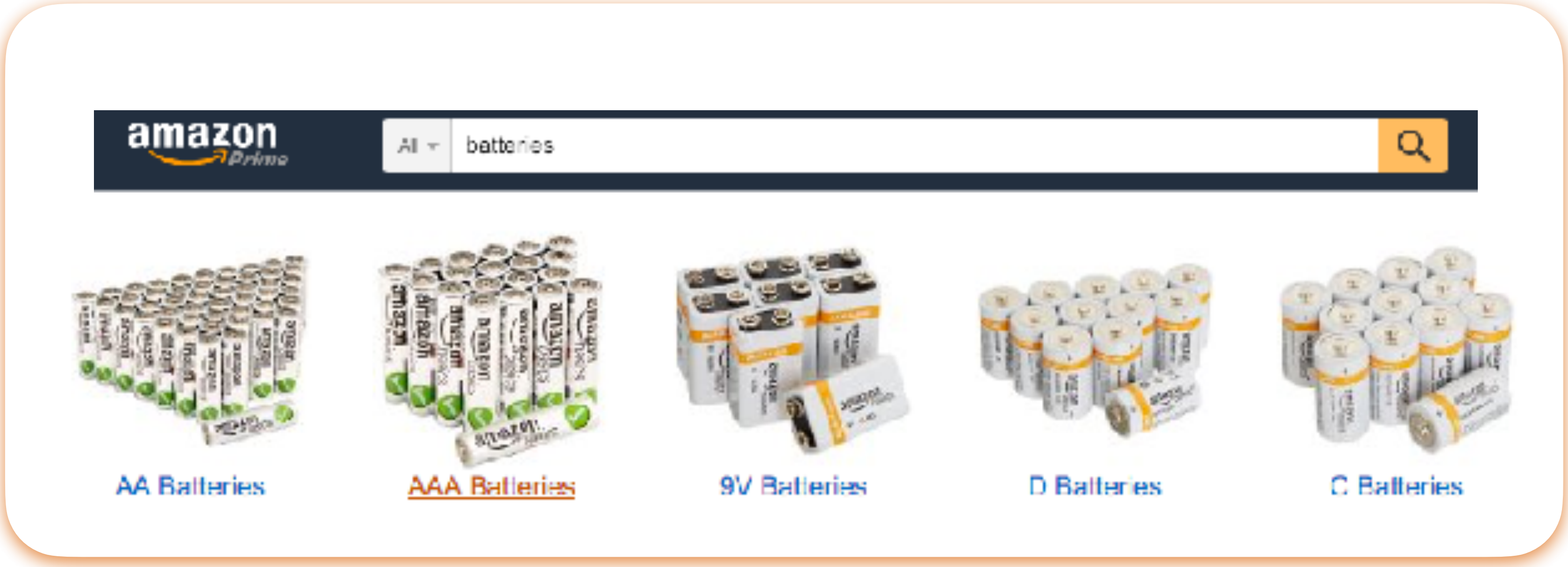
Partially ordered sets with a bottom

to maximize $\langle \mathbf{F}, \preceq_{\mathbf{F}} \rangle$

to minimize $\langle \mathbf{R}, \preceq_{\mathbf{R}} \rangle$

implementations





Posets model trade-offs

$\langle \mathbb{R}_{\geq 0}, \leq \rangle \quad \langle \mathbb{N}, \leq \rangle$

Croque et frites *Quiche lorraine*

American pastries

Cordon bleu in fondue

Fondue

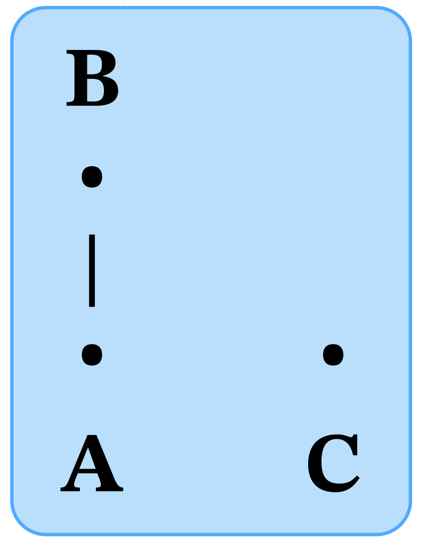
Cordon bleu

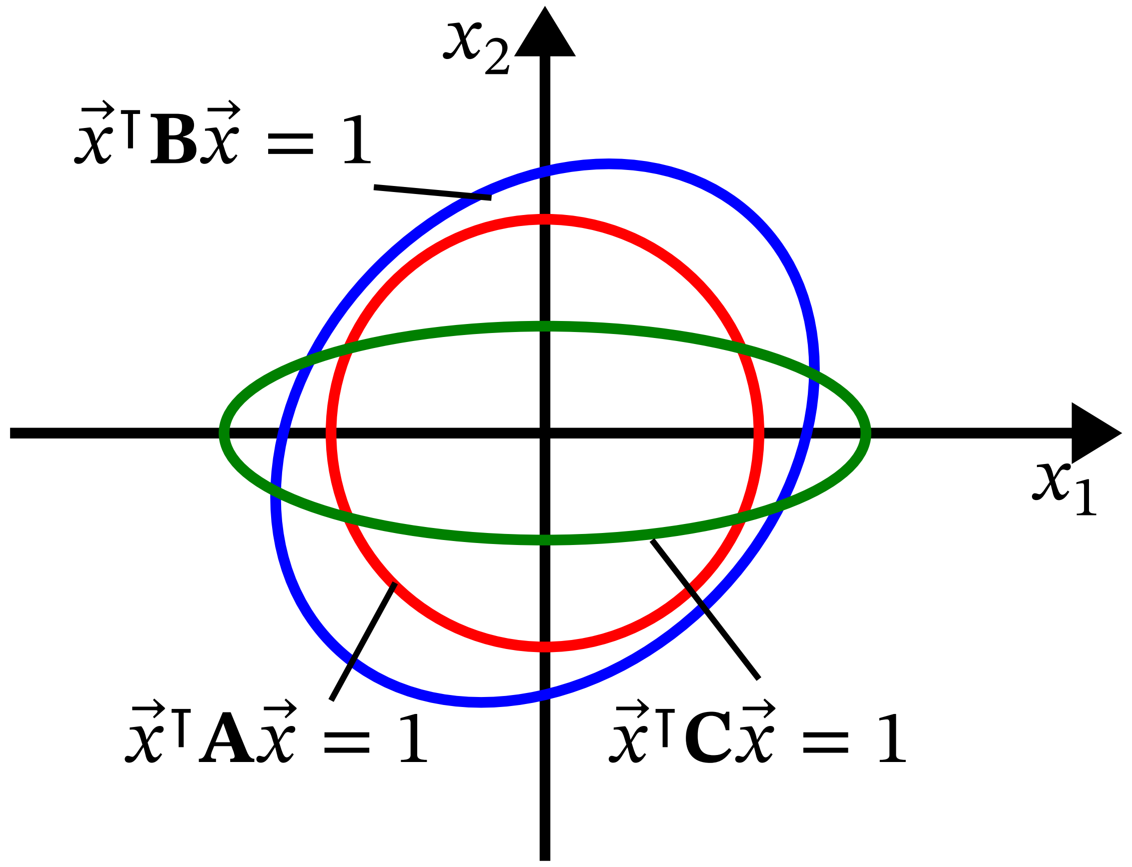
A poset of positive-definite matrices

$$\frac{\mathbf{A} \preceq_{\mathsf{PDM}(n)} \mathbf{B}}{\vec{x}^\intercal \mathbf{A} \vec{x} \leq \vec{x}^\intercal \mathbf{B} \vec{x} \quad \forall \vec{x} \in \mathbb{R}^n}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 3/4 & -1/8 \\ -1/8 & 3/4 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1/2 & 0 \\ 0 & 2 \end{bmatrix}$$

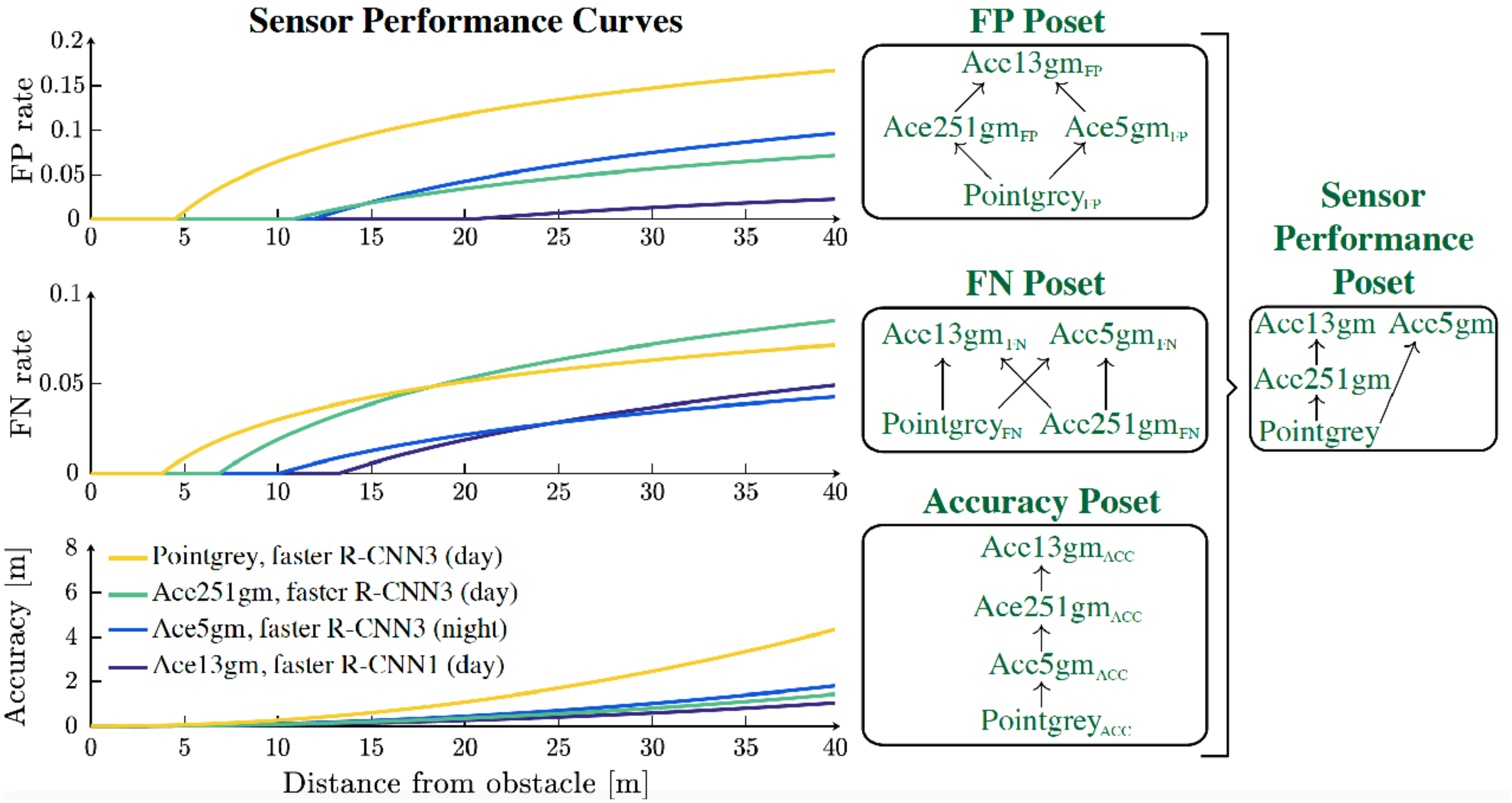
PDM(2)





Posets model trade-offs

A poset of sensor/algorithm pairs



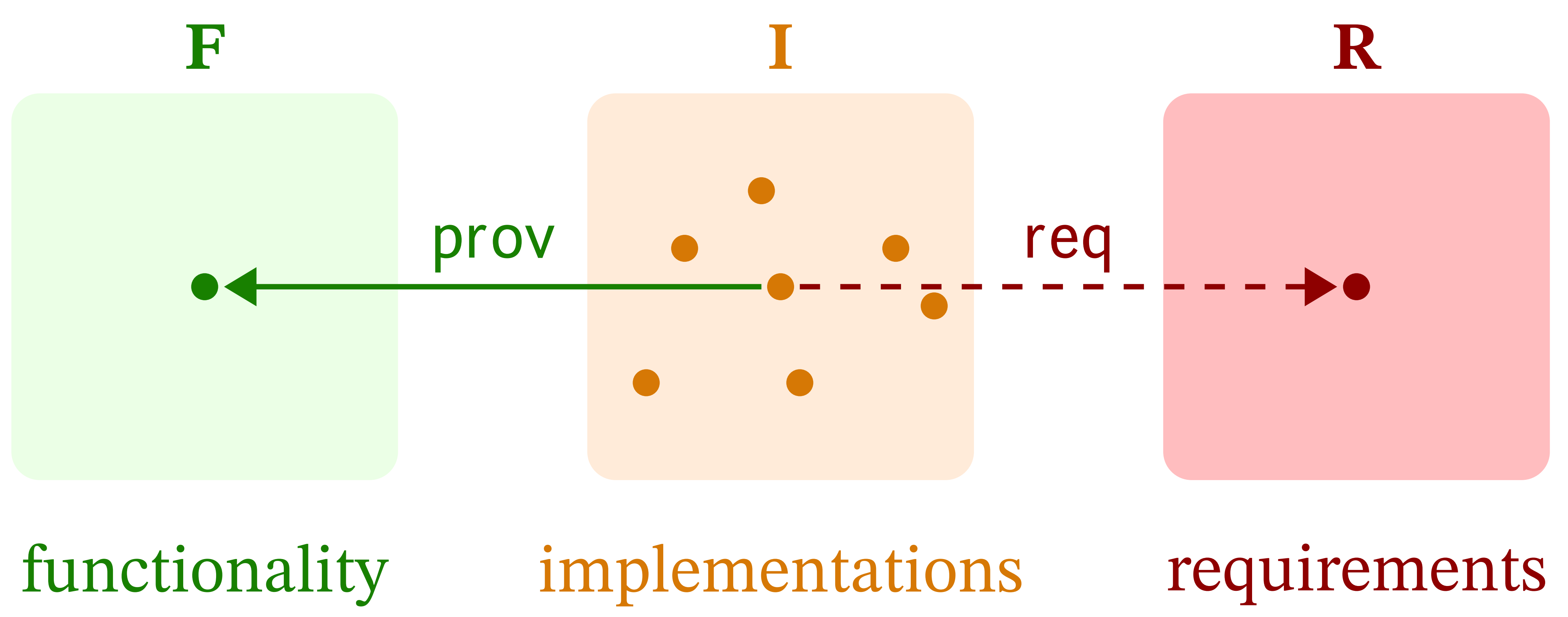
Design problem with implementation (DPIs)

Definition (Design problem with implementation). A *design problem with implementation* (DPI) is a tuple

$$\langle \mathbf{F}, \mathbf{R}, \mathbf{I}, \mathtt{prov}, \mathtt{req} \rangle,$$

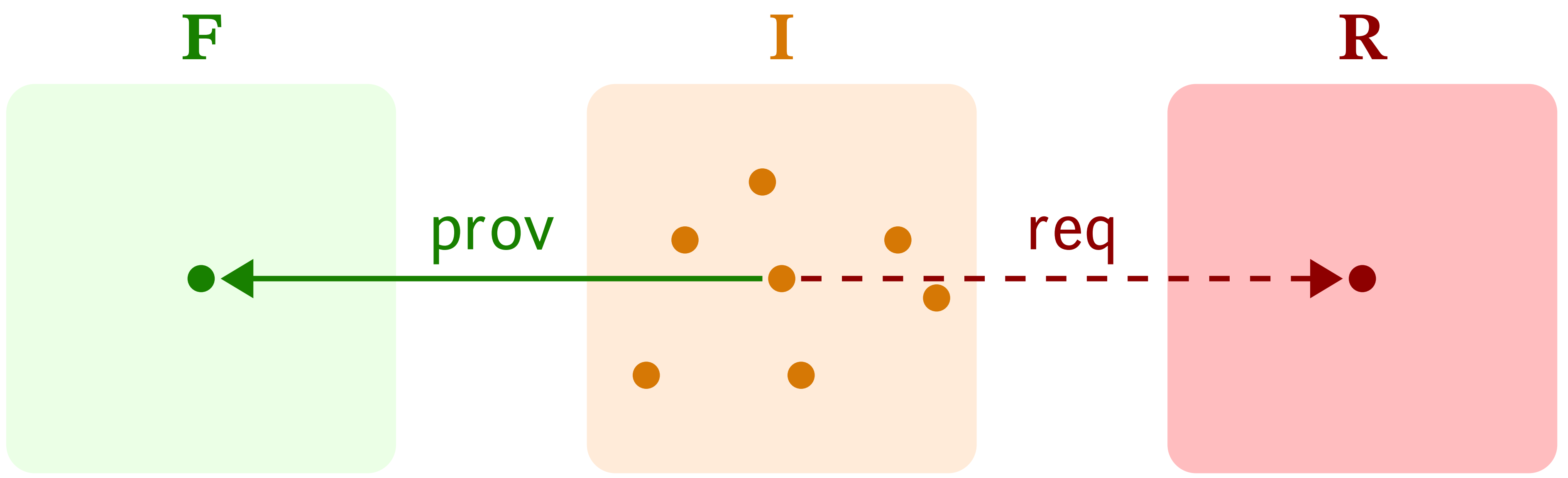
where:

- $\mathbf{F}$ is a poset, called *functionality space*;
- $\mathbf{R}$ is a poset, called *requirements space*;
- $\mathbf{I}$ is a set, called *implementation space*;
- the map $\mathtt{prov} : \mathbf{I} \to \mathbf{F}$ maps an implementation to the functionality it provides;
- the map $\mathtt{req} : \mathbf{I} \to \mathbf{R}$ maps an implementation to the resources it requires.



Transparent vs black-box models

- The **DPI model** is a **“transparent” model:**



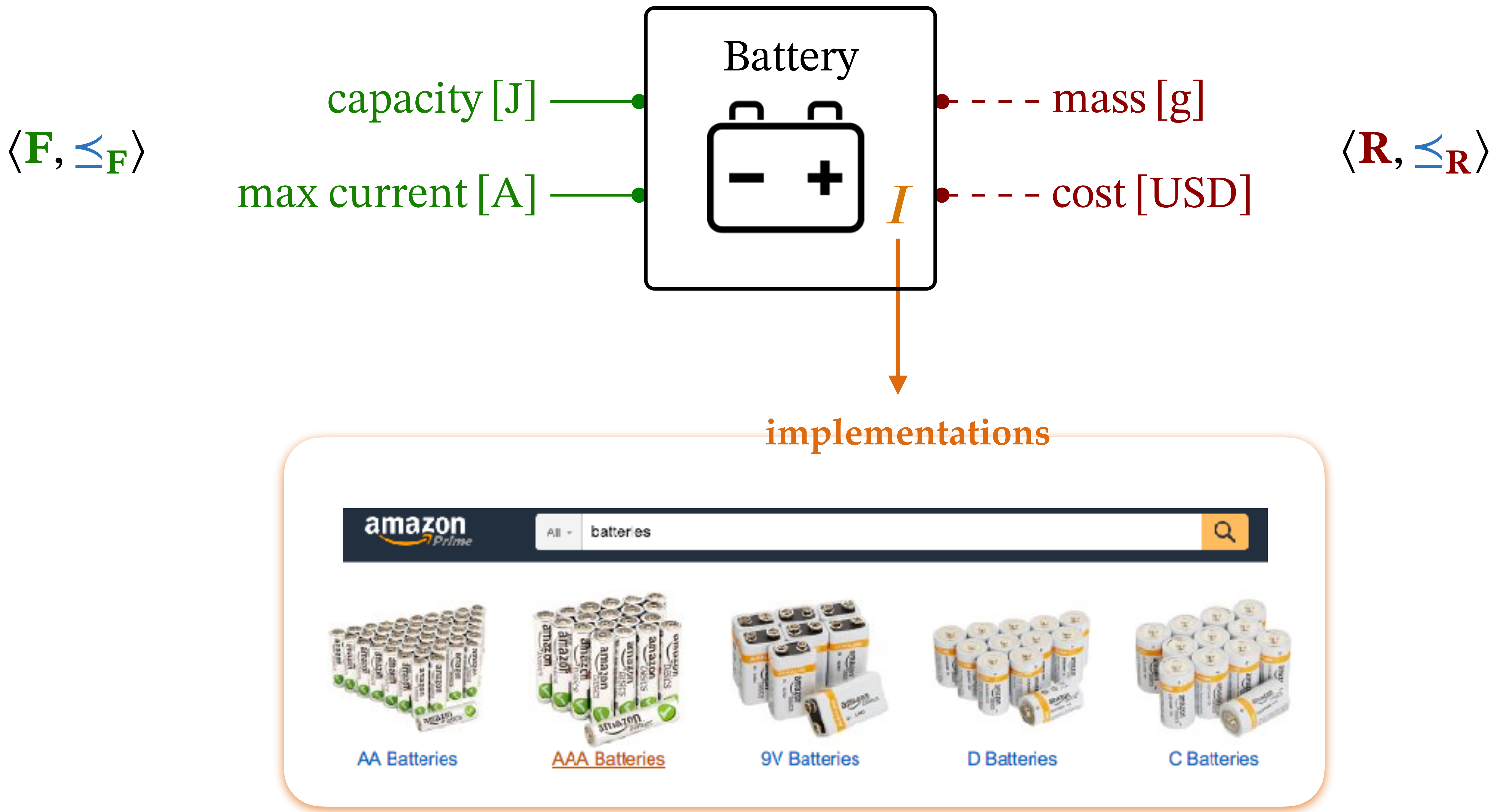
- **DP** model: **direct feasibility relation** between functionality and resources (“black box”).

$$\mathbf{d} : \mathbf{F}^{\mathrm{op}} \times \mathbf{R} \to_{\mathbf{Pos}} \mathbf{Bool}$$
$$\langle f^*, r \rangle \mapsto \exists i \in \mathbf{I} : (f \preceq_{\mathbf{F}} \mathsf{prov}(i)) \wedge (\mathsf{req}(i) \preceq_{\mathbf{R}} r)$$

- **Monotonicity** assumption:
  - **Lower functionality** does **not** require **more resources**;
  - More **resources** do not provide **less functionality.**

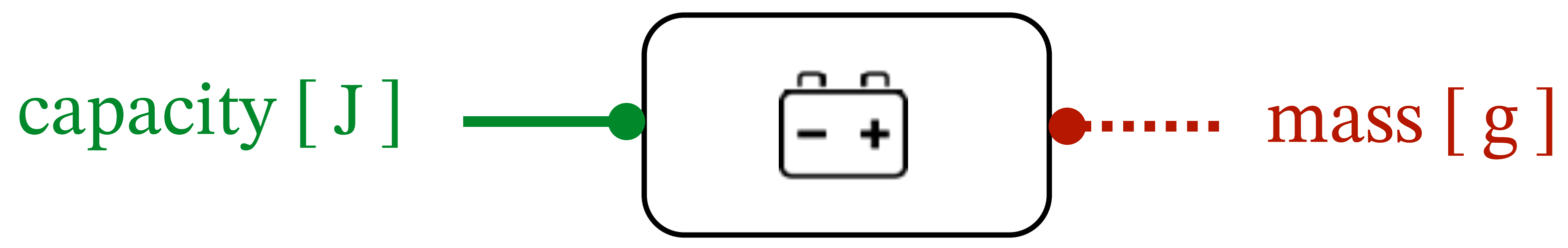
Graphical notation for DPIs

- We use this graphical notation:
  - functionality: **green continuous wires** on the left
  - requirements: **dashed red wires** on the right.

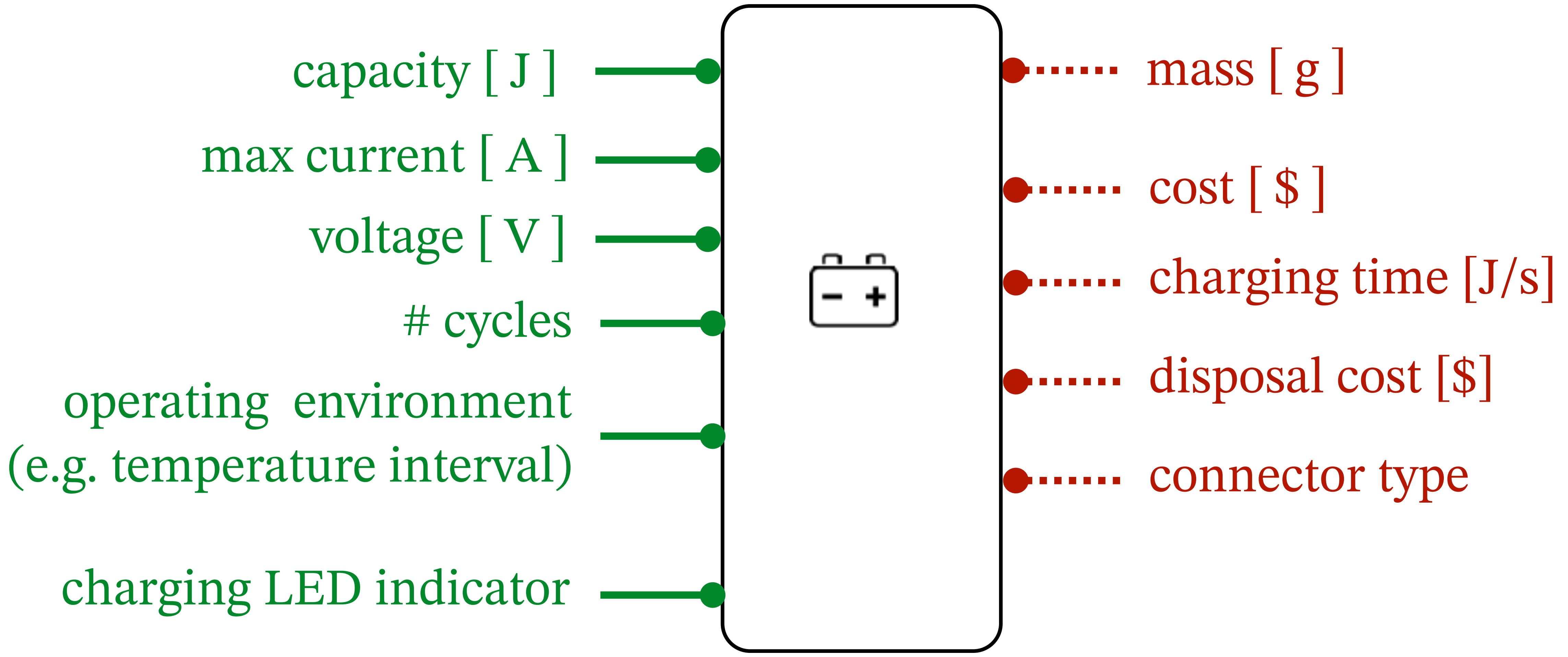


Context informs the level of detail

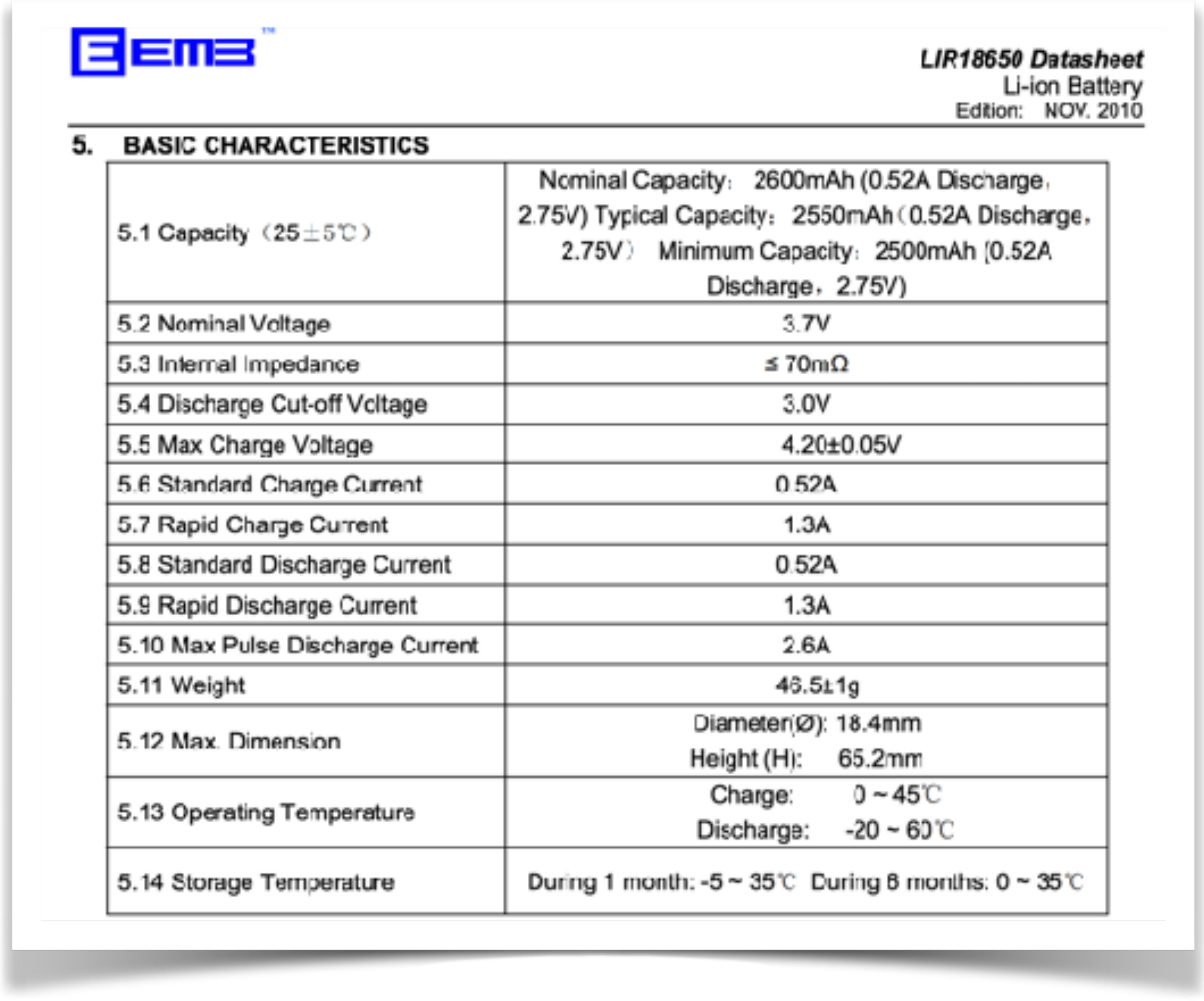
- Different scenarios will need different levels of detail.



“back of the envelope calculation”



“vendor selection”



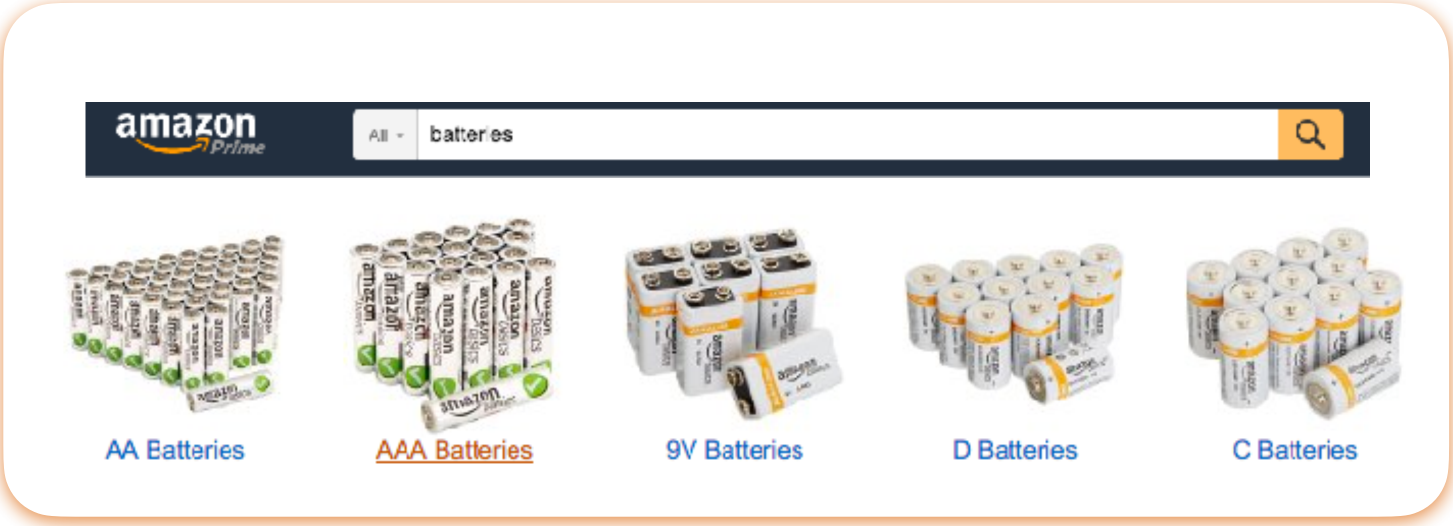
LIR18650 Datasheet
Li-ion Battery
Edition: NOV. 2010

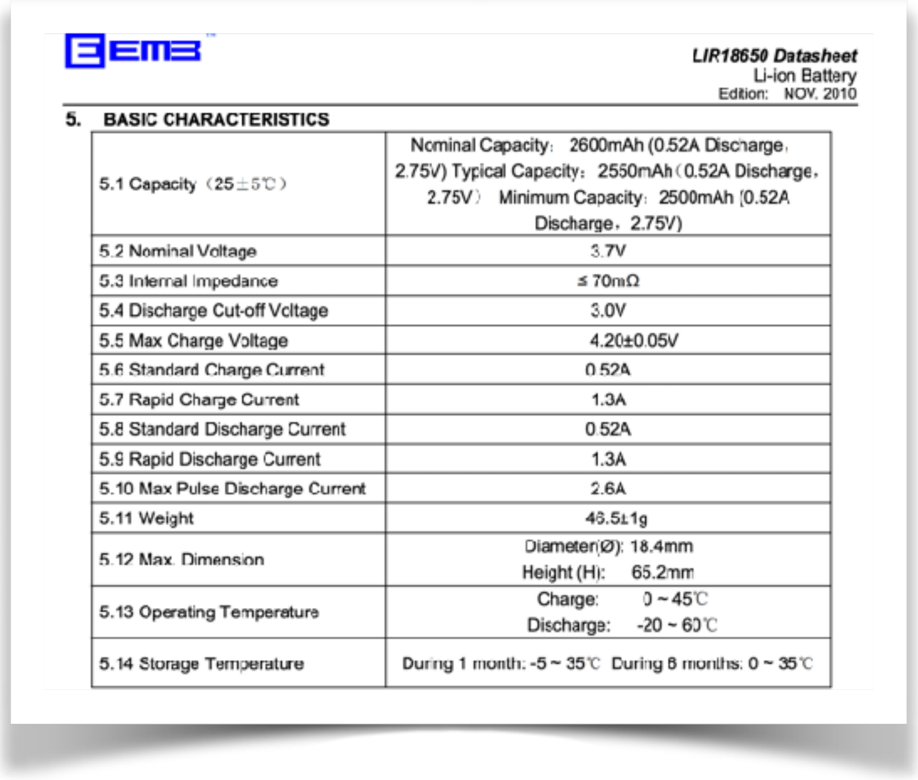
5. BASIC CHARACTERISTICS

5.1 Capacity (25±5℃)	Nominal Capacity: 2600mAh (0.52A Discharge, 2.75V) Typical Capacity: 2550mAh (0.52A Discharge, 2.75V) Minimum Capacity: 2500mAh (0.52A Discharge, 2.75V)
5.2 Nominal Voltage	3.7V
5.3 Internal Impedance	≤ 70mΩ
5.4 Discharge Cut-off Voltage	3.0V
5.5 Max Charge Voltage	4.20±0.05V
5.6 Standard Charge Current	0.52A
5.7 Rapid Charge Current	1.3A
5.8 Standard Discharge Current	0.52A
5.9 Rapid Discharge Current	1.3A
5.10 Max Pulse Discharge Current	2.6A
5.11 Weight	46.5±1g
5.12 Max. Dimension	Diameter(Ø): 18.4mm Height (H): 65.2mm
5.13 Operating Temperature	Charge: 0 ~ 45℃ Discharge: -20 ~ 60℃
5.14 Storage Temperature	During 1 month: -5 ~ 35℃ During 6 months: 0 ~ 35℃

Model types

- "**Catalogues**": already available designs



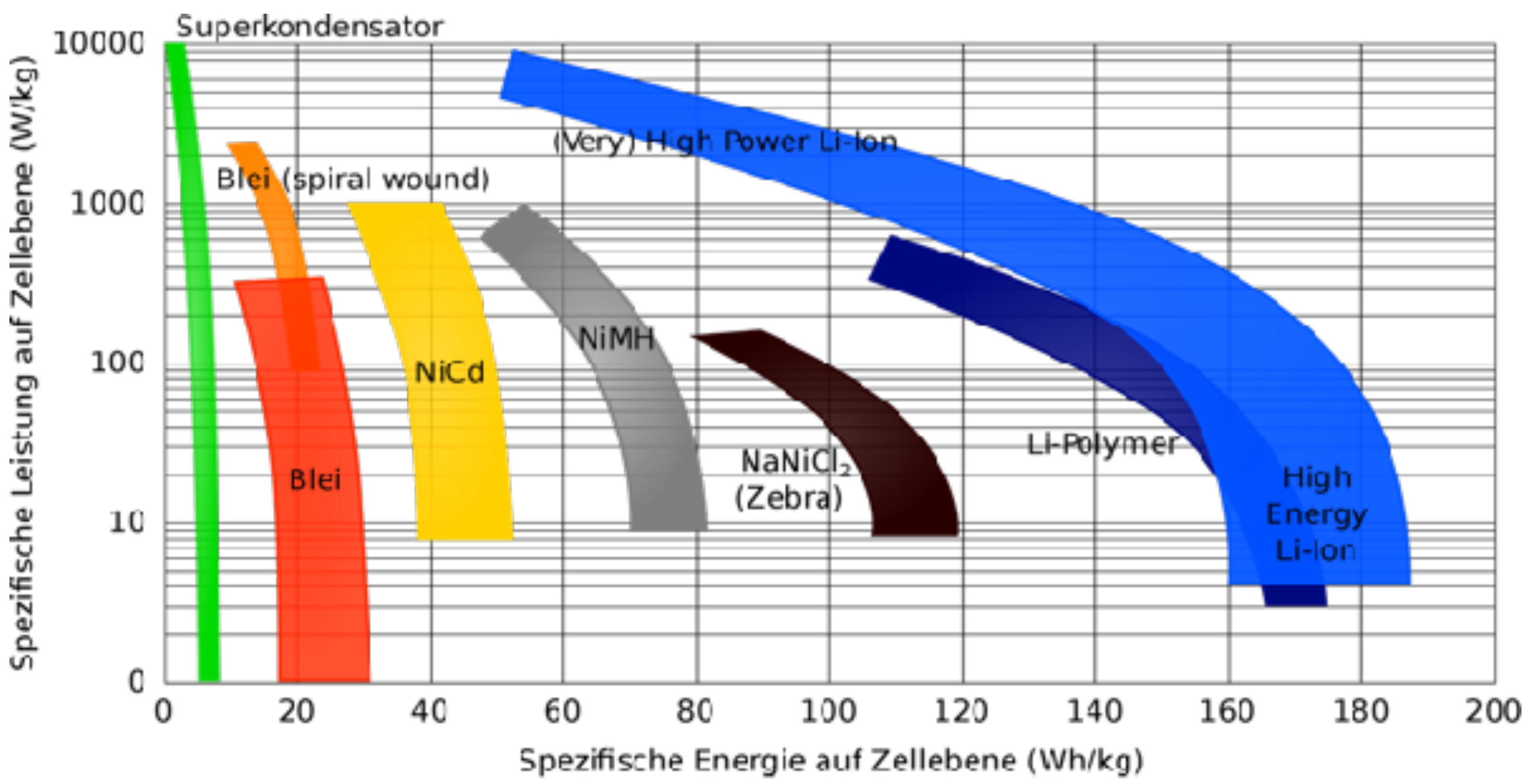


LIR18650 Datasheet
Li-ion Battery
Edition: NOV. 2010

5. BASIC CHARACTERISTICS

5.1 Capacity（25±5℃）	Nominal Capacity: 2600mAh (0.52A Discharge, 2.75V) Typical Capacity: 2550mAh（0.52A Discharge, 2.75V） Minimum Capacity: 2500mAh (0.52A Discharge, 2.75V)
5.2 Nominal Voltage	3.7V
5.3 Internal Impedance	≤ 70mΩ
5.4 Discharge Cut-off Voltage	3.0V
5.5 Max Charge Voltage	4.20±0.05V
5.6 Standard Charge Current	0.52A
5.7 Rapid Charge Current	1.3A
5.8 Standard Discharge Current	0.52A
5.9 Rapid Discharge Current	1.3A
5.10 Max Pulse Discharge Current	2.6A
5.11 Weight	46.5±1g
5.12 Max. Dimension	Diameter(Ø): 18.4mm Height (H): 65.2mm
5.13 Operating Temperature	Charge: 0 ~ 45℃ Discharge: -20 ~ 60℃
5.14 Storage Temperature	During 1 month: -5 ~ 35℃ During 6 months: 0 ~ 35℃

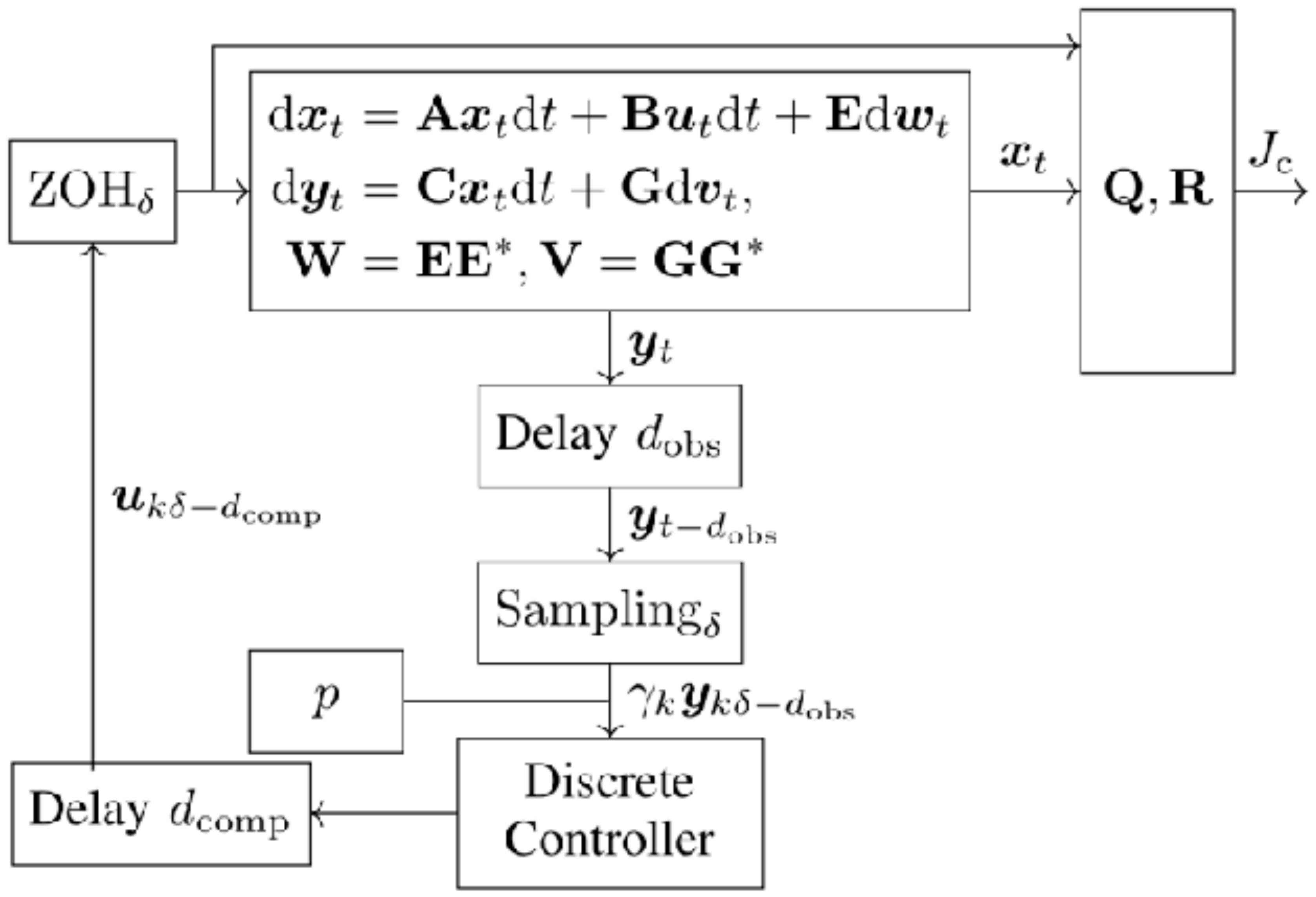
- "**First-principles**": analytical relations.



- "**Data-driven, on-demand**"
  - The optimization algorithm will only ask for a sequence of data points specific to the queries. The model is constructed incrementally (experiments, black-box simulations).
- **Uncertain models**

Re-stating existing knowledge in co-design form

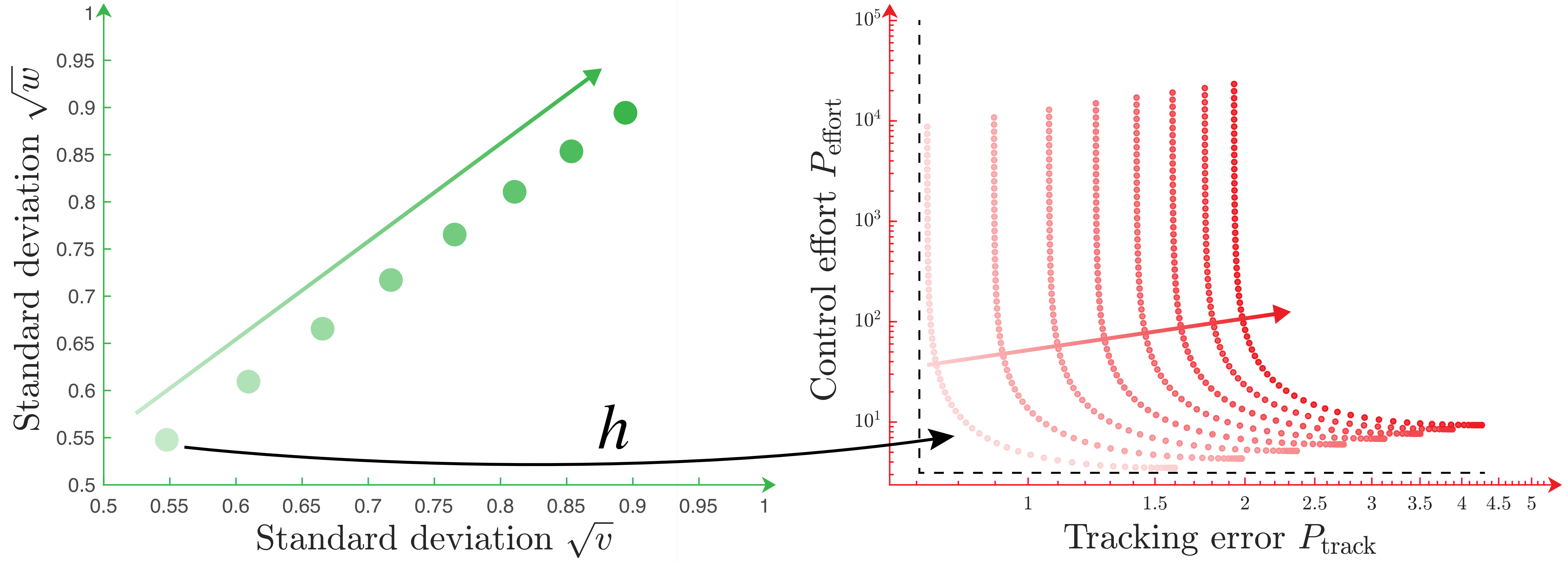
- Take the usual setup for **LQG control:**



$$P_{\text{track}} = \lim_{t\to\infty} \mathbb{E}\{\mathbf{x}_t^\intercal \mathbf{Q}\mathbf{x}_t\}$$

$$P_{\text{effort}} = \lim_{t\to\infty} \mathbb{E}\{\mathbf{u}_t^\intercal \mathbf{R}\mathbf{u}_t\}$$

- We can manipulate known results to state this **DP theorem:**



Convex Optimization Problems are Design Problems

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array}$$

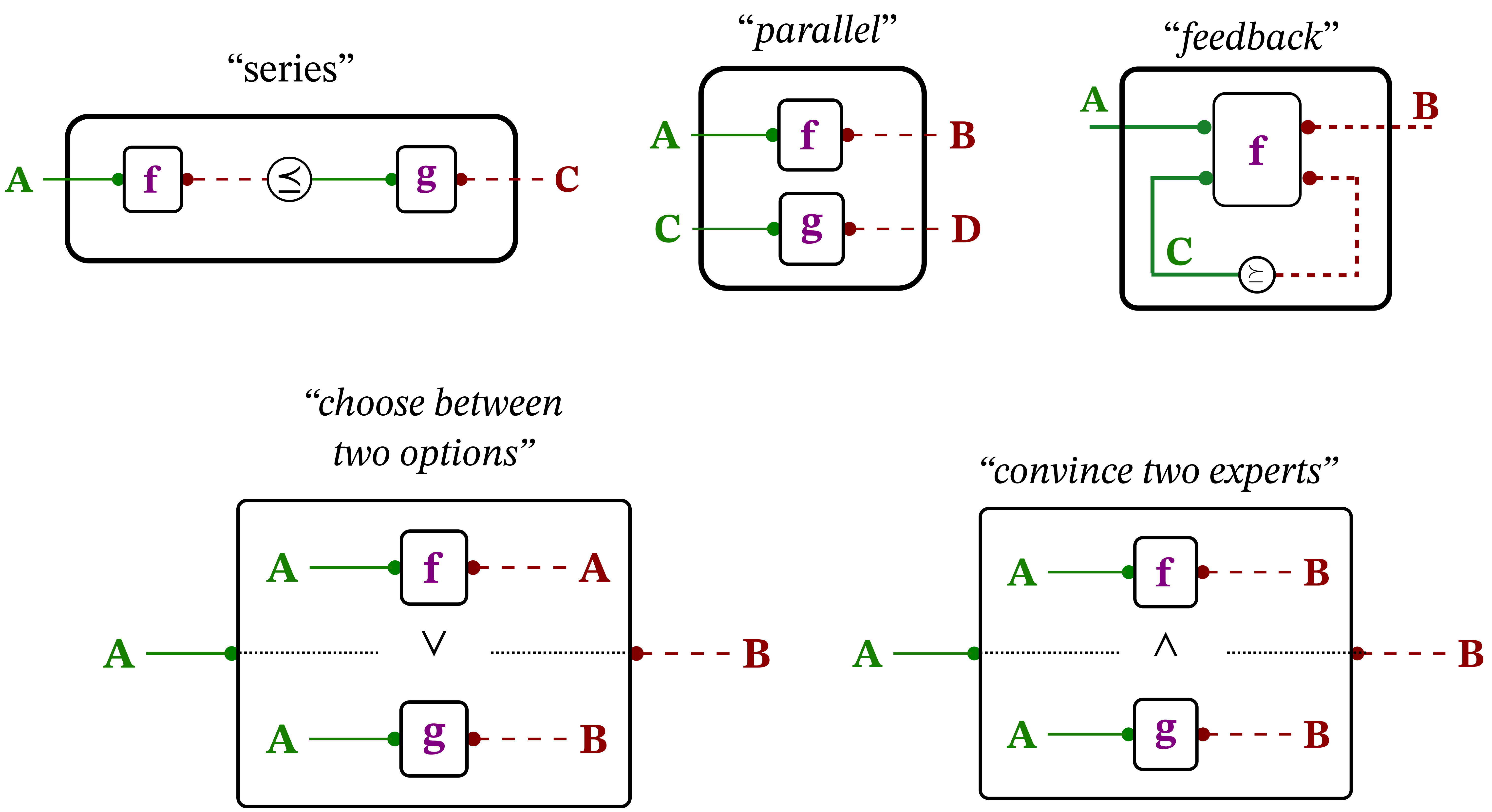
Theorem (Convex Optimization Problem as Monotone Map)

A convex optimization problem is a monotone map CvxOpt *from* $\langle \mathcal{C}, \preceq_{\mathbf{C}} \rangle \times \left\langle \mathcal{F}_{\mathcal{C}}, \preceq_{\mathbf{F}_{\mathcal{C}}} \right\rangle$ *to* $\langle \mathbb{R}, \preceq_{\mathbf{R}}^{\mathrm{op}} \rangle$.

$$\begin{aligned} \mathrm{CvxOpt} : \langle \mathcal{C}, \preceq_{\mathbf{C}} \rangle \times \left\langle \mathcal{F}_{\mathcal{C}}, \preceq_{\mathbf{F}_{\mathcal{C}}} \right\rangle &\rightarrow_{\mathbf{Pos}} \langle \mathbb{R}, \preceq_{\mathbf{R}}^{\mathrm{op}} \rangle \\ \langle \mathcal{D}_f, f_0 \rangle &\mapsto \hat{p}^* \end{aligned}$$

where $\hat{p} = \inf_{x \in \mathcal{D}_f} f_0(x)$ *and* $\mathcal{D}_f = \{x \in \mathbb{R}^n \mid f_i(x) \leq 0, i \in [m], Ax = b\}$.

Composition operators



- The **composition** of any two **DPs** returns a **DP** (closure)

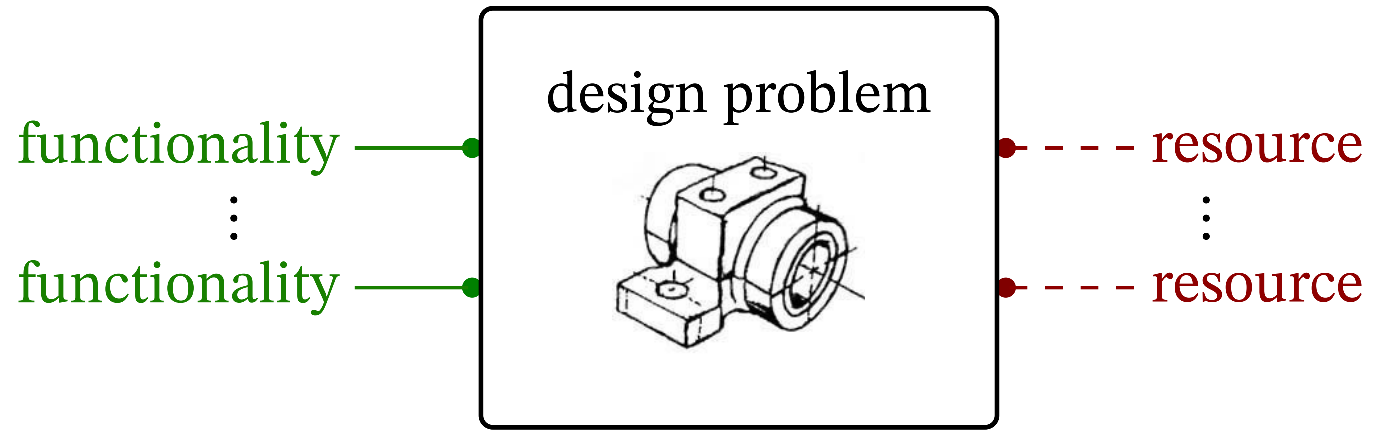
- Very practical tool to **decompose** large **problems** into **subproblems**
- This makes the category **DP** *traced monoidal and locally posetal*

Design queries

- Two basic design queries are:
  - **FixFunMinReq**: Fixed a lower bound on functionality, minimize the resources.
  - **FixReqMaxFun**: Fixed an upper bound on the resource, maximize the functionality

Given the functionality to be provided,
what are the **minimal resources** required?

FixFunMinReq →



← **FixReqMaxFun**

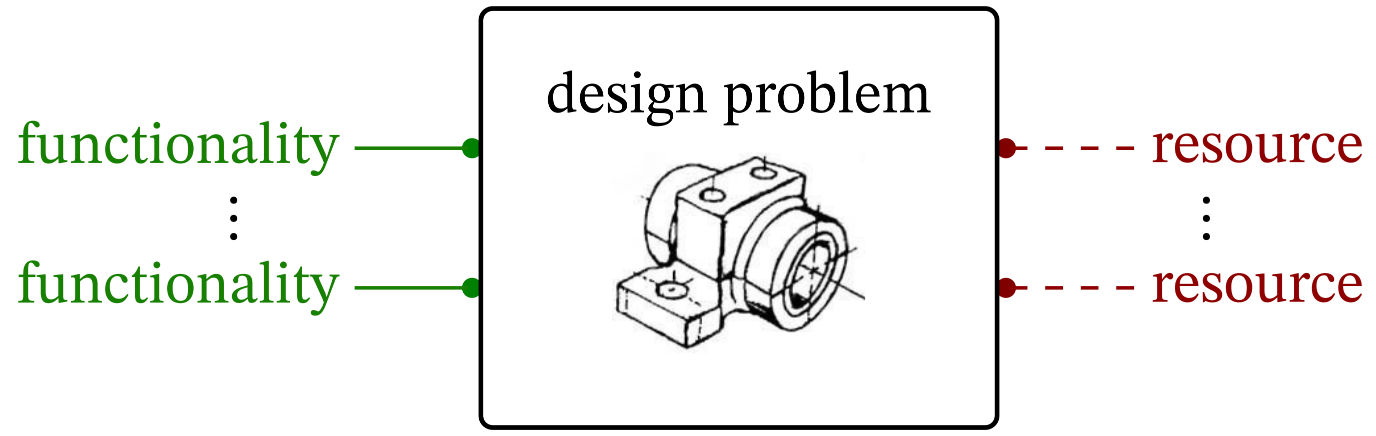
Given the resources that are available, what is
the **maximal functionality** that can be provided?

Design queries

- Two basic design queries are:
  - **FixFunMinReq**: Fixed a lower bound on functionality, minimize the resources.
  - **FixReqMaxFun**: Fixed an upper bound on the resource, maximize the functionality

Given the functionality to be provided, what are the **minimal resources** required?

FixFunMinReq



FixReqMaxFun

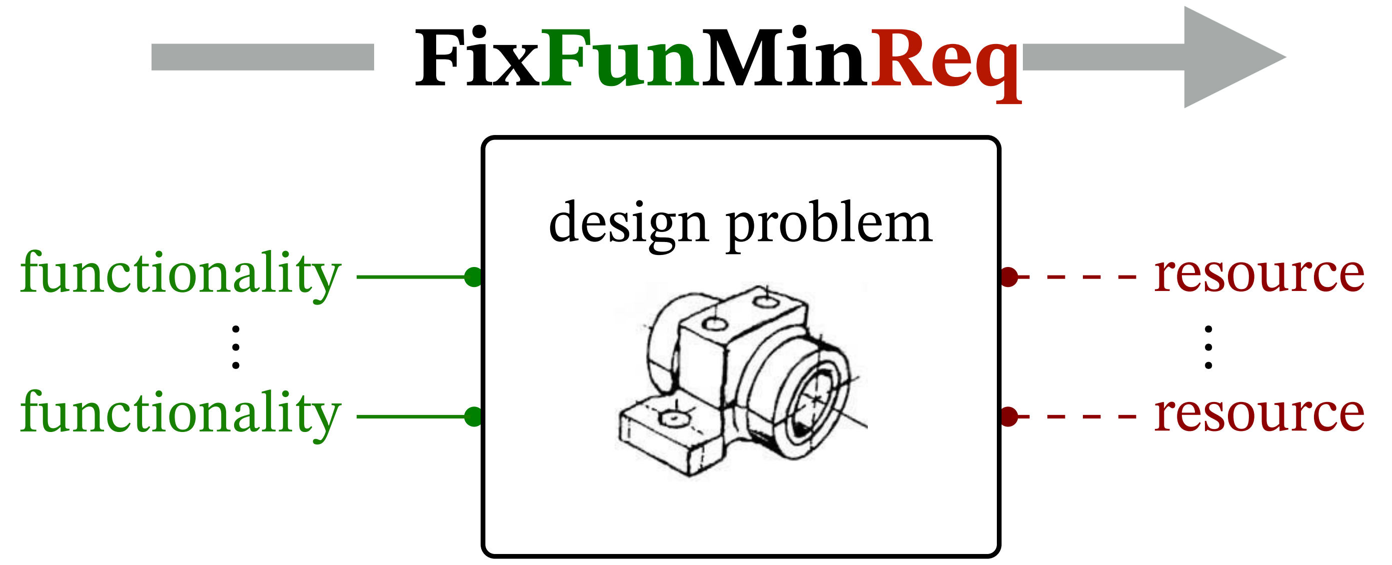
Given the resources that are available, what is the **maximal functionality** that can be provided?

- The two problems are **dual**
- From the solutions, one can retrieve the **implementations** (design choices)

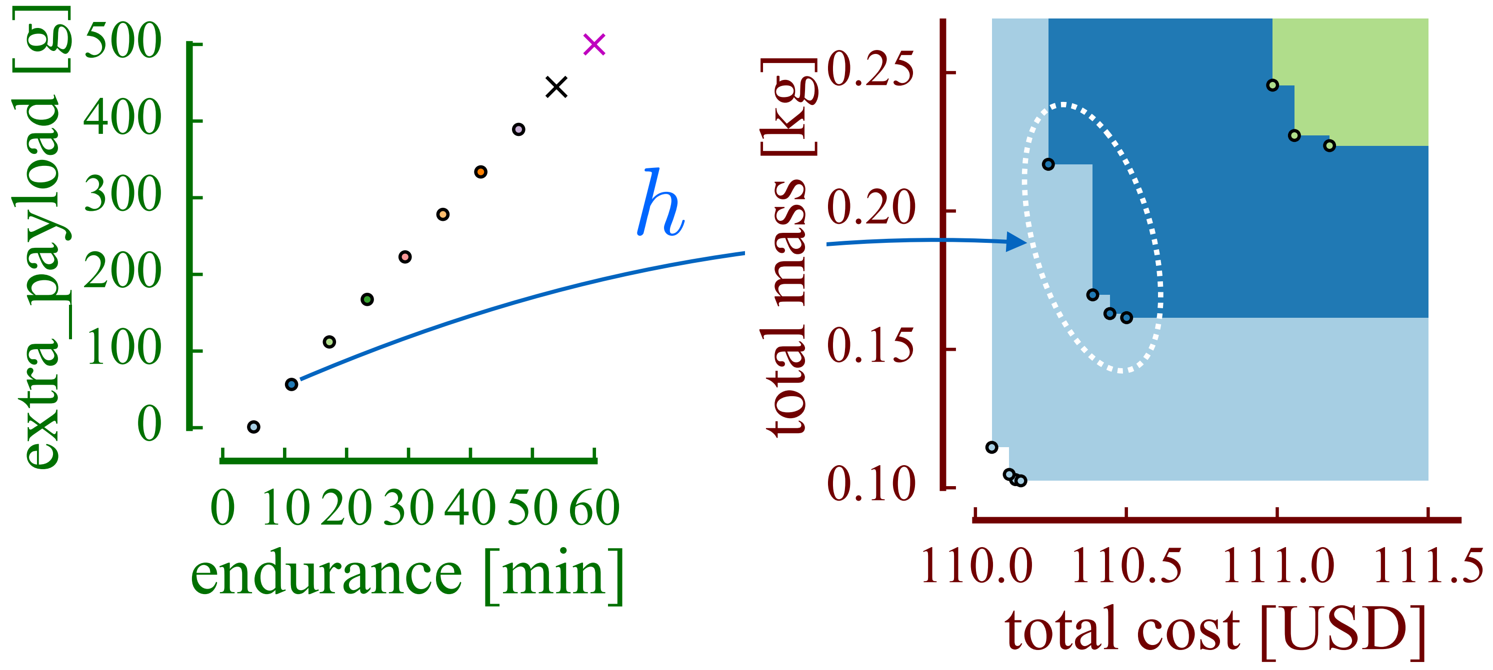
Design queries

- Two basic design queries are:
  - **FixFunMinReq**: Fixed a lower bound on functionality, minimize the resources.
  - **FixReqMaxFun**: Fixed an upper bound on the resource, maximize the functionality

Given the functionality to be provided,
what are the **minimal resources** required?



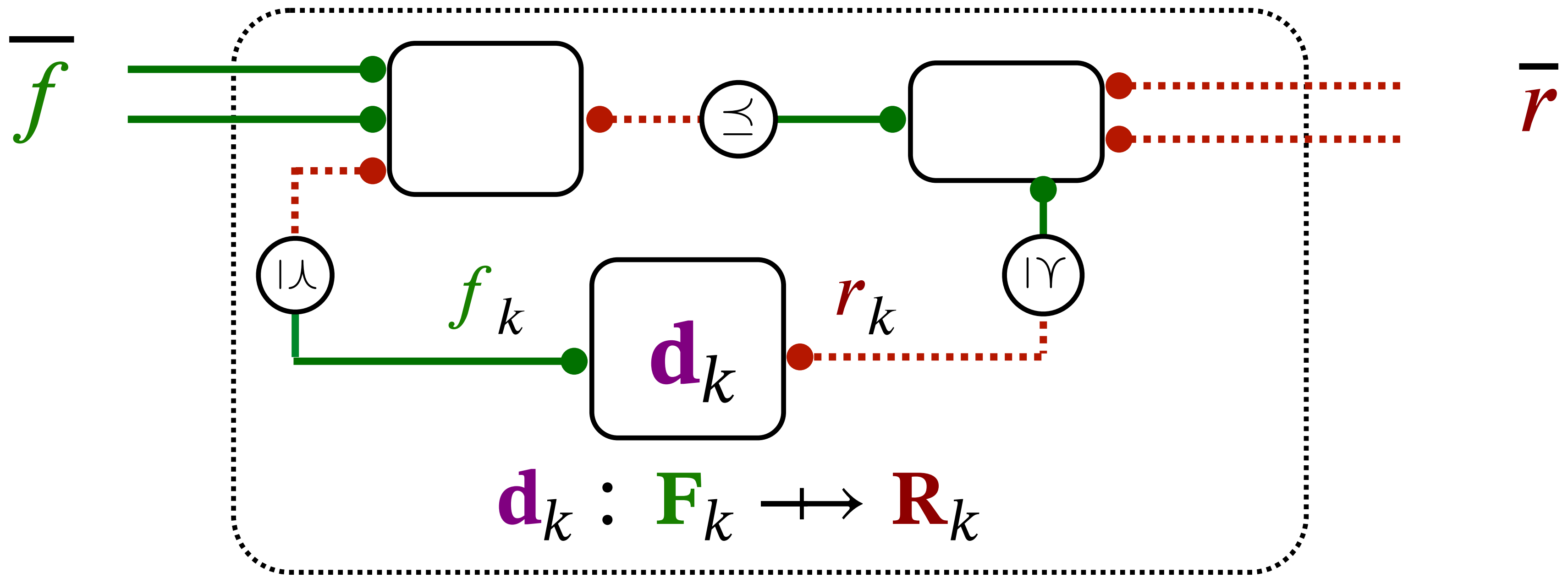
- We are looking for:
  - A map from functionality to **upper sets** of feasible resources: $h : \mathbf{F} \to \mathcal{U}\mathbf{R}$
  - A map from functionality to **antichains** of minimal resources: $h : \mathbf{F} \to \mathcal{A}\mathbf{R}$



Optimization semantics

- This is the semantics of **FixFunMinReq** as a **family of optimization problems.**

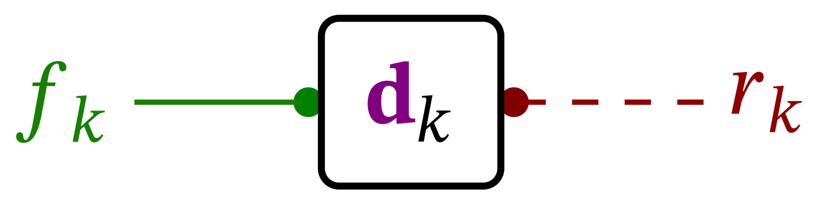
chosen by user

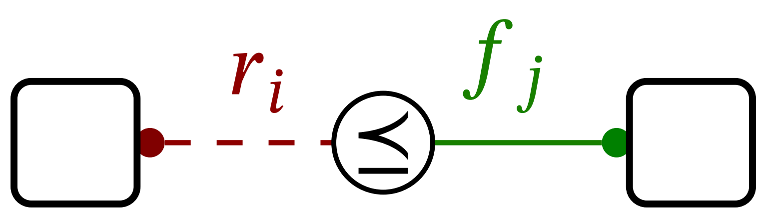


to minimize

variables $r_k \in \langle \mathbf{R}_k, \preceq_{\mathbf{R}_k} \rangle$ $f_k \in \langle \mathbf{F}_k, \preceq_{\mathbf{F}_k} \rangle$

constraints *for **each node**:* *for **each edge:***





$\mathbf{d}_k(f_k^*, r_k) = \top$ $r_i \preceq f_j$

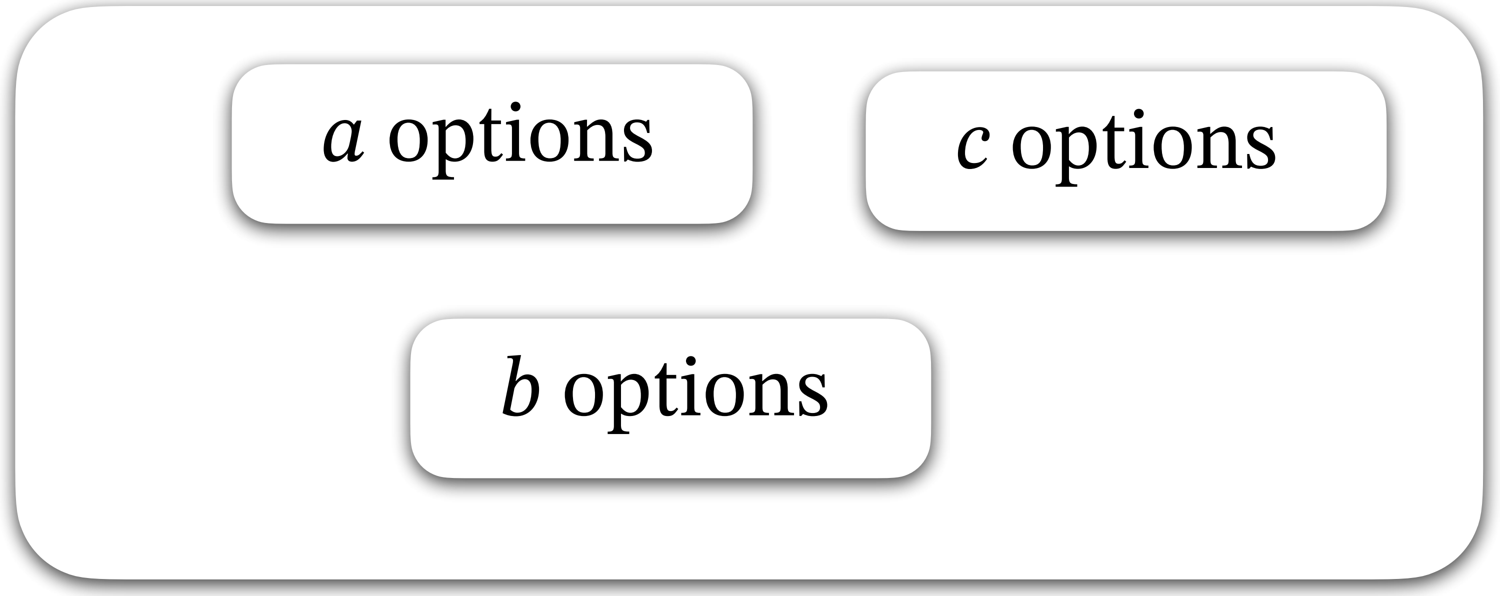
objective $\underset{\preceq}{\mathrm{Min}}\, \bar{r}$

- ! not convex
- ! not differentiable
- ! not continuous
- ! not even defined on continuous spaces

Monotone co-design problems are tractable

- We have a **complete solution**: guaranteed to find the set all optimal solutions,
  (If such set is empty, the algorithm trace is a certificate of infeasibility)

- The complexity is **not combinatorial in the number of options** for each component

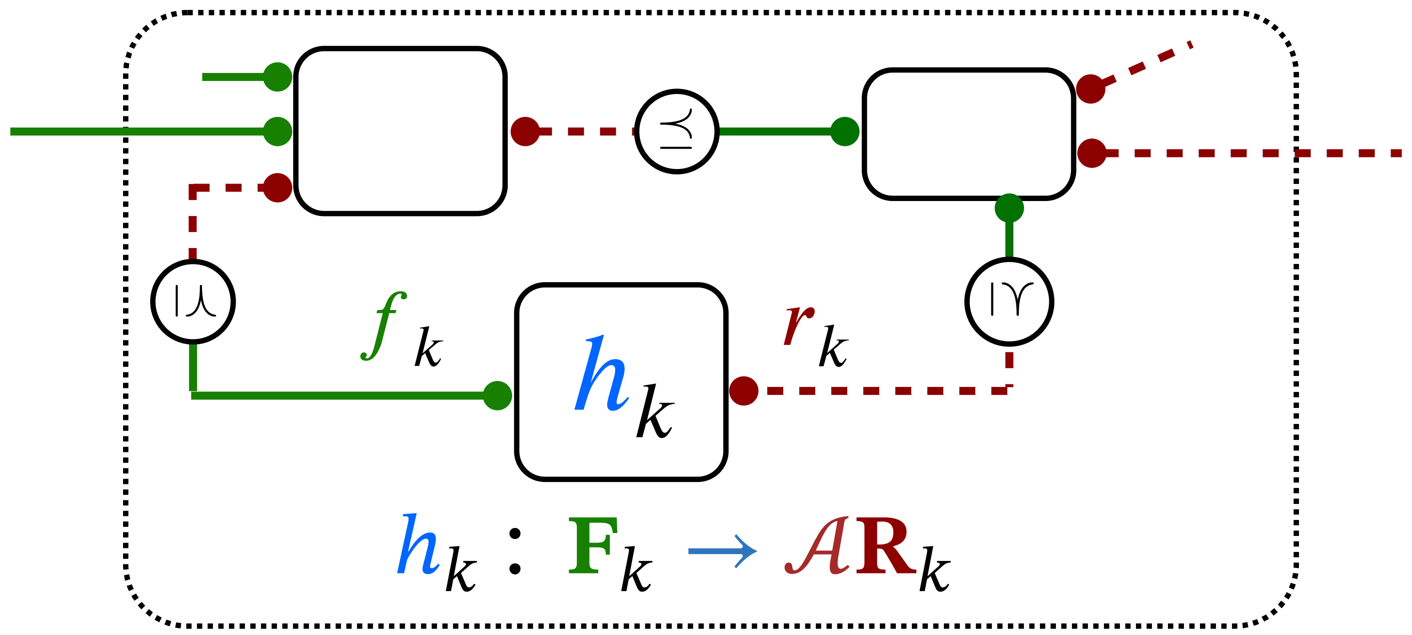


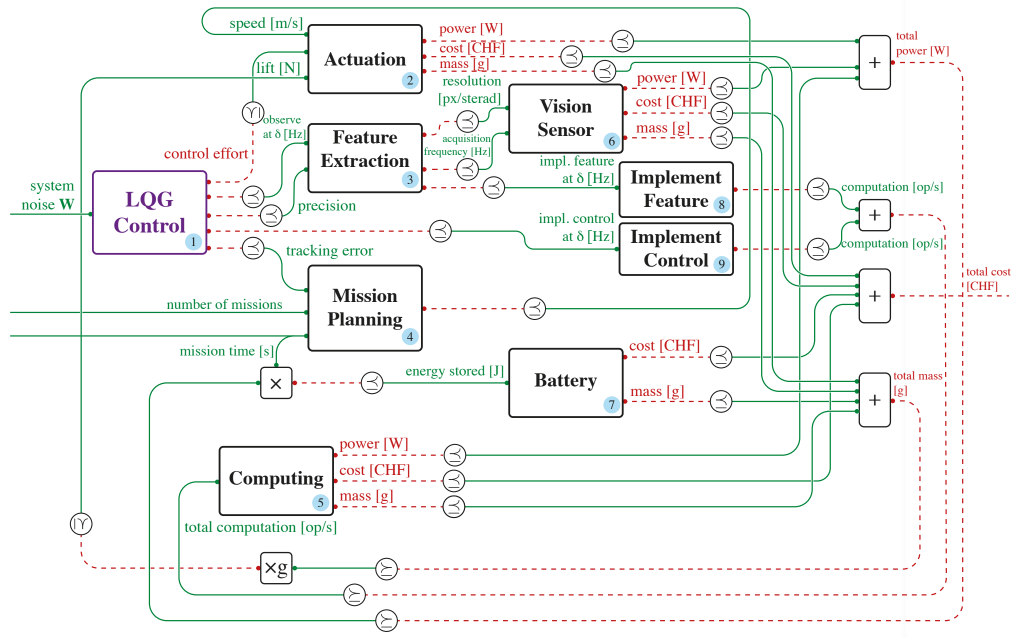
$O(a + b + c)$

- The complexity depends on the **complexity of the interactions**: the co-design constraints.

Compositional approach to optimization

- **Assume** (for now) that all posets are finite: results are finite antichains of resources.
- Suppose we are given the function $h_k : \mathbf{F}_k \to \mathcal{A}\mathbf{R}_k$ for all nodes in the co-design graph.





- How to find the map $h : \mathbf{F} \to \mathcal{A}\mathbf{R}$ for the entire diagram?
- **Compositional approach:**
  - Given that we have defined the diagram recursively using composition operations, we just need to work out the composition formulas.

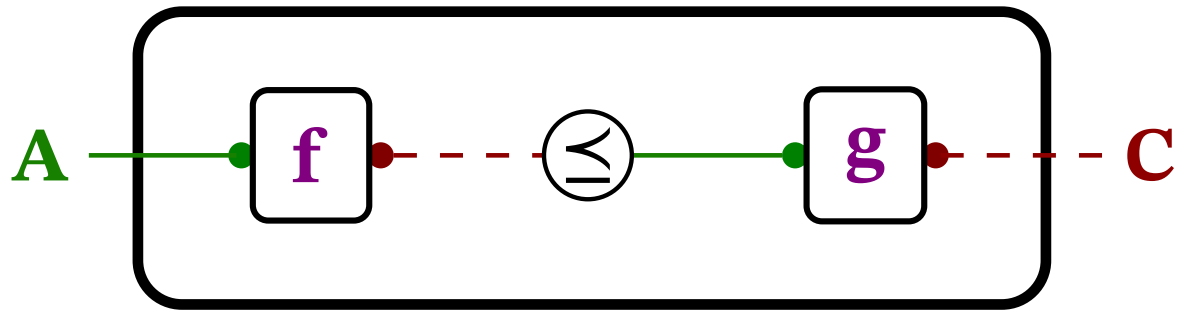
    solution(**composition**(a, b)) = **composition**(**solution**(a), **solution**(b))

  - This is **equivalent to finding a functor** (monoidal and lattice-compatible) from the category **DP** to the category of solution maps.

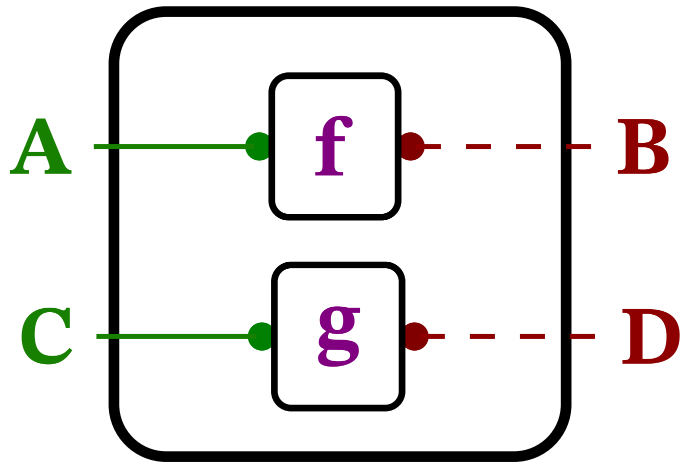
$$\mathbf{DP}(\mathbf{F}; \mathbf{R}) \xrightarrow{\textbf{FixFunMinReq}} \left(\mathbf{F} \to \mathcal{A}\mathbf{R} \right)_{\mathbf{Pos}}$$

Compositional approach to optimization

- We can *easily* write the solution for all composition operations *except feedback*.



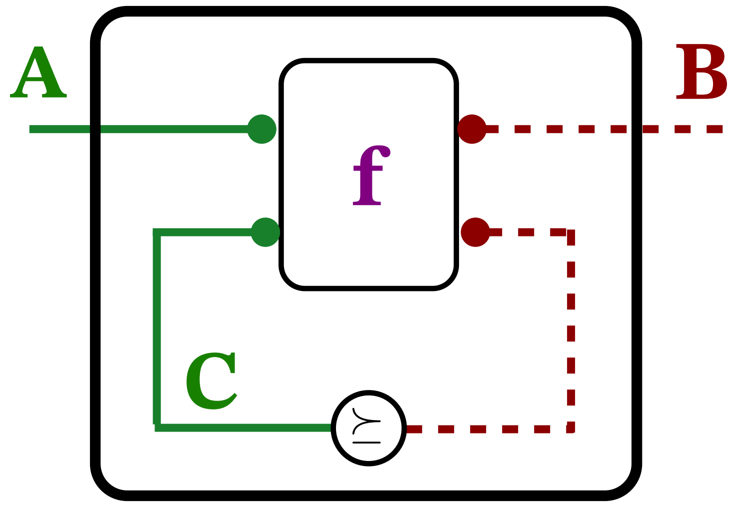
$$h_{\mathbf{f}\fatsemi\mathbf{g}} : \mathbf{A} \to \mathsf{A}\,\mathbf{C}$$
$$a \mapsto \operatorname*{Min}_{\preceq_{\mathbf{C}}} \bigcup_{s \in h_{\mathbf{f}}(a)} h_{\mathbf{g}}(s).$$

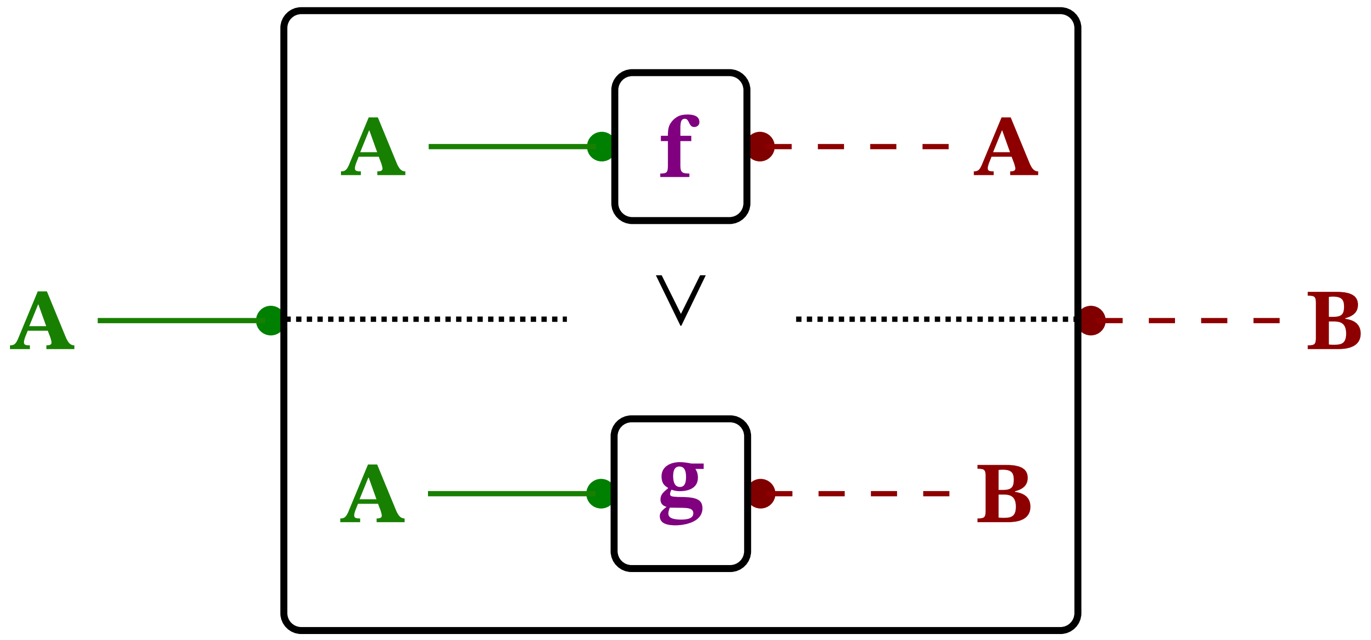


$$h_{\mathbf{f}} \otimes h_{\mathbf{g}} : (\mathbf{A} \times \mathbf{C}) \to \mathsf{A}\,(\mathbf{B} \times \mathbf{D}),$$
$$\langle a, c\rangle \mapsto h_{\mathbf{f}}(a) \times h_{\mathbf{g}}(c),$$

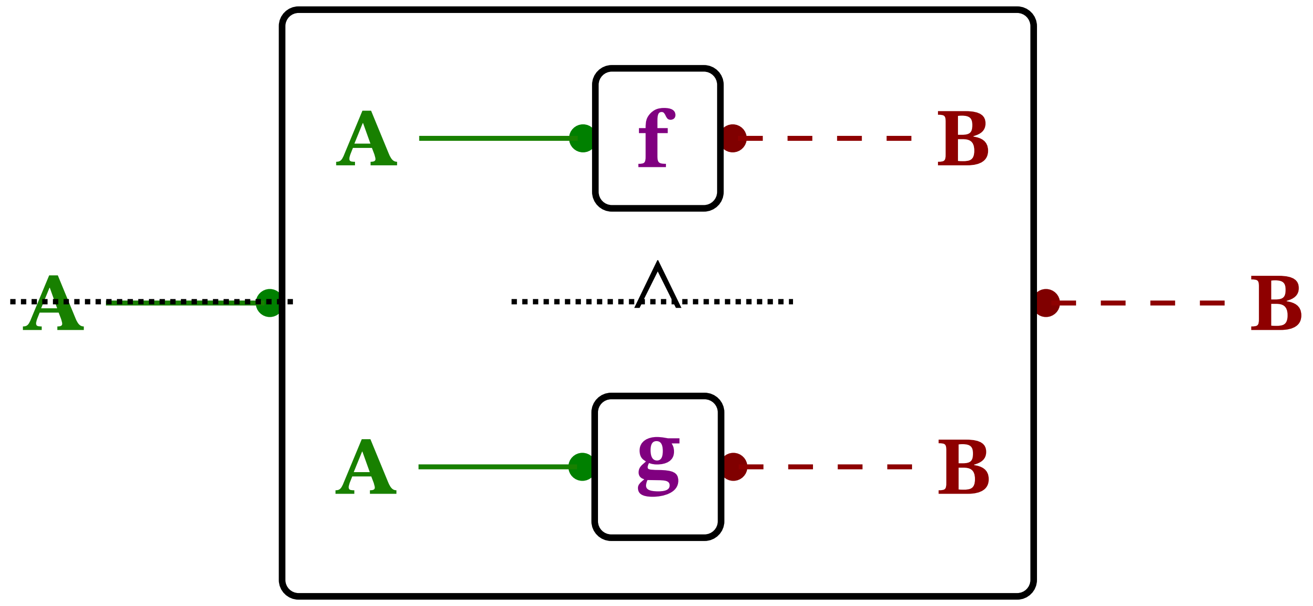
feedback is always the problem...

“*feedback*”





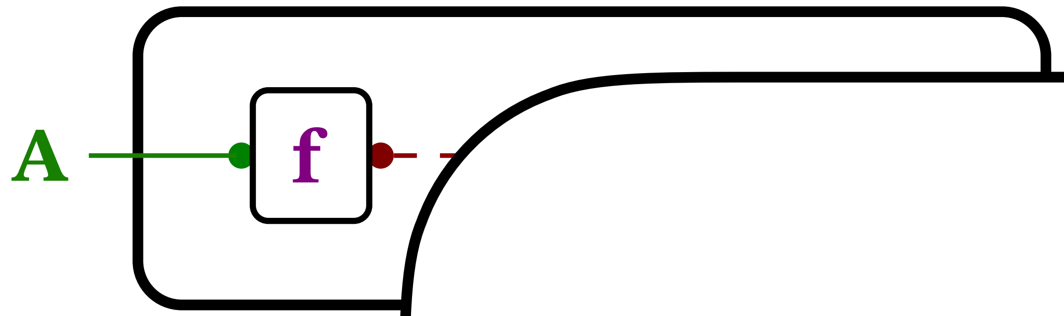
$$h_{\mathbf{f}} \vee h_{\mathbf{g}} : \mathbf{A} \to \mathsf{A}\,\mathbf{B},$$
$$a \mapsto \operatorname*{Min}_{\preceq_{\mathbf{B}}} \left(h_{\mathbf{f}}(a) \cup h_{\mathbf{g}}(a)\right).$$



$$h_{\mathbf{f}} \wedge h_{\mathbf{g}} : \mathbf{A} \to \mathsf{A}\,\mathbf{B},$$
$$a \mapsto \operatorname*{Min}_{\preceq_{\mathbf{B}}} \left(h_{\mathbf{f}}(a) \cap h_{\mathbf{g}}(a)\right).$$

Compositional approach to optimization

- We can *easily* write the solution for all composition operations *except feedback*.



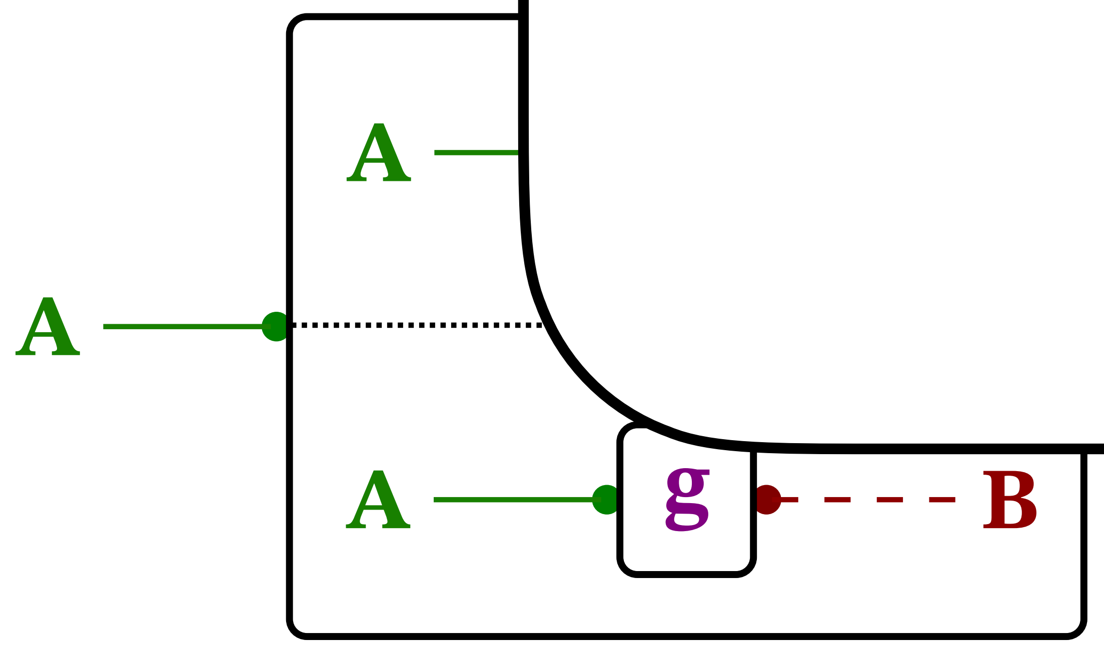
$h_{\mathbf{f} ; \mathbf{g}} : \mathbf{A} \to$

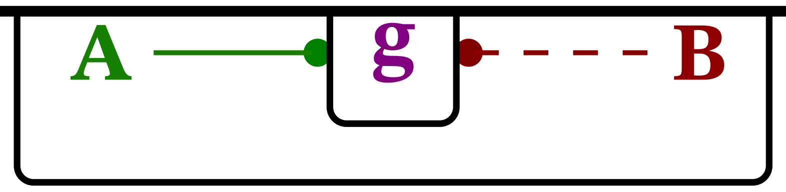
$a \mapsto$

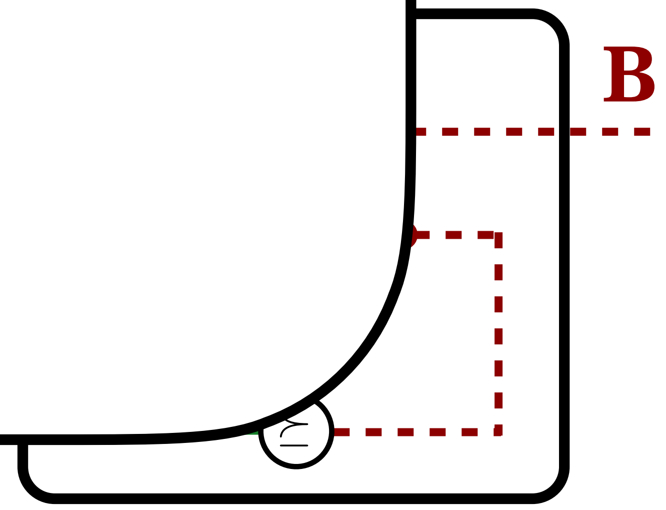
Check out the details of the solution in the book

(Rely on Kleene's Algorithm)

he problem...







$$h_{\mathbf{f}} \vee h_{\mathbf{g}} : \mathbf{A} \to \mathsf{A}\, \mathbf{B},$$
$$a \mapsto \operatorname*{Min}_{\preceq_{\mathbf{B}}} \left(h_{\mathbf{f}}(a) \cup h_{\mathbf{g}}(a) \right).$$

$$h_{\mathbf{f}} \wedge h_{\mathbf{g}} : \mathbf{A} \to \mathsf{A}\, \mathbf{B},$$
$$a \mapsto \operatorname*{Min}_{\preceq_{\mathbf{B}}} \left(h_{\mathbf{f}}(a) \cap h_{\mathbf{g}}(a) \right).$$

Developer vs. user view

Developer view

- **Applied category theory**
- **Domain theory**

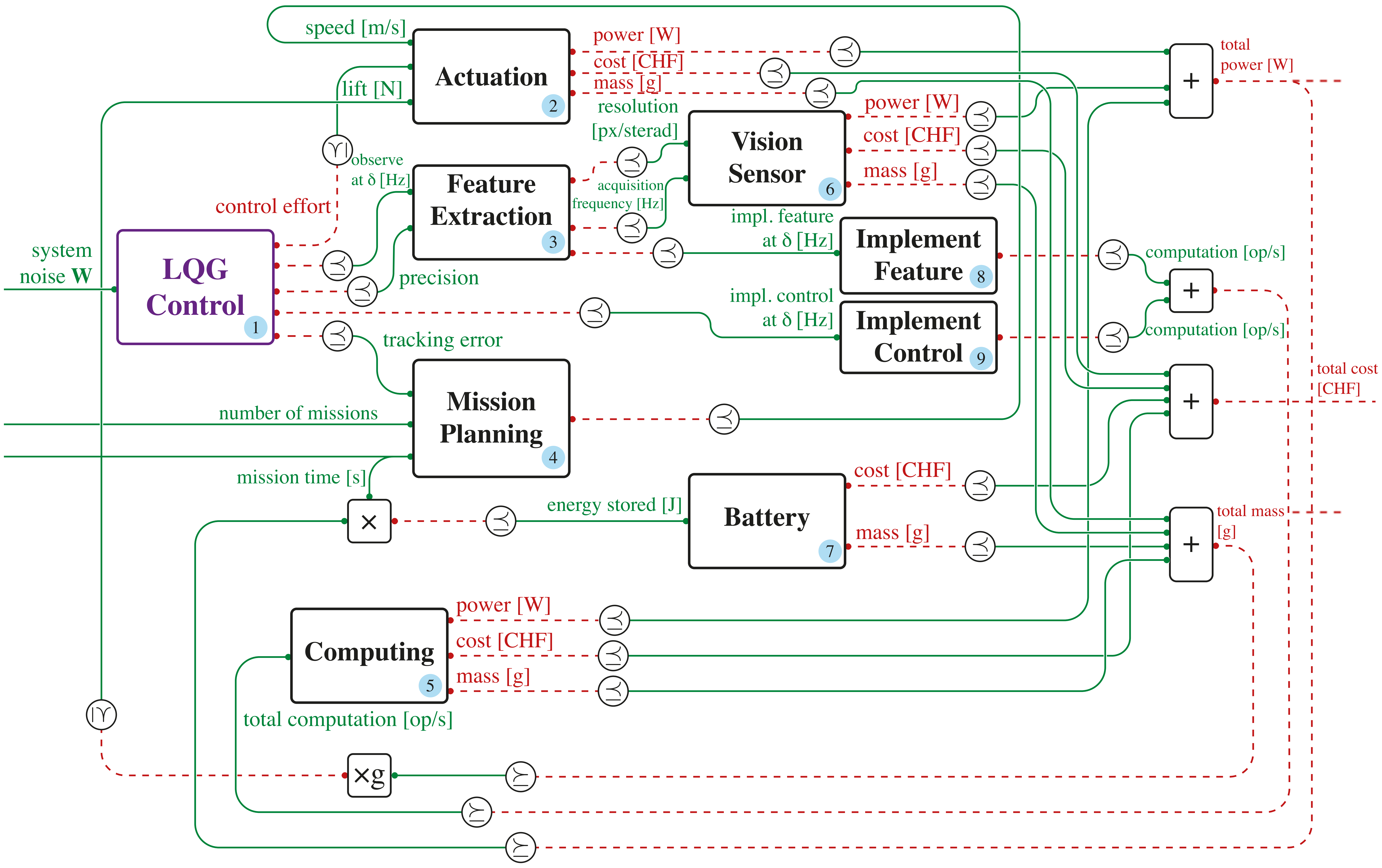
- "**Catalogues**": already available designs

```
catalogue {
    provides capacity [J]
    requires mass [g]
    requires cost [USD]

    500 kWh ⟻ model1 ⟼ 100 g, 10 USD
    600 kWh ⟻ model2 ⟼ 200 g, 200 USD
    600 kWh ⟻ model3 ⟼ 250 g, 150 USD
    700 kWh ⟻ model4 ⟼ 400 g, 400 USD
}
```

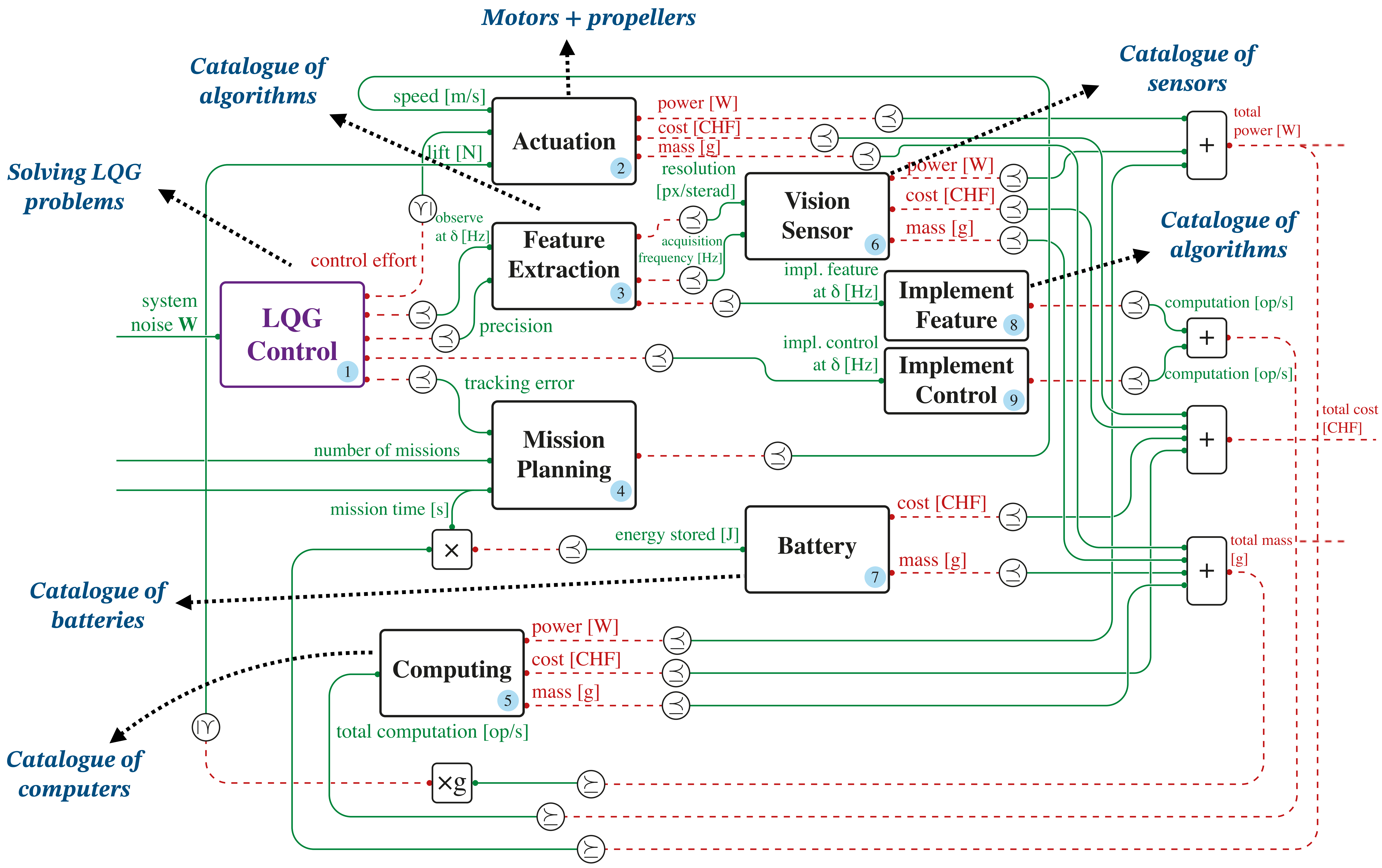
- "**First-principles**": analytical relations.

```
mcdp {
    provides capacity [J]
    requires mass [kg]

    specific_energy_Li_Ion = 500 Wh / kg

    required mass >= provided capacity / specific_energy_Li_Ion
}
```

Use case: Co-design of an autonomous drone



Use case: Co-design of an autonomous drone



Co-design of an AV: systematic process

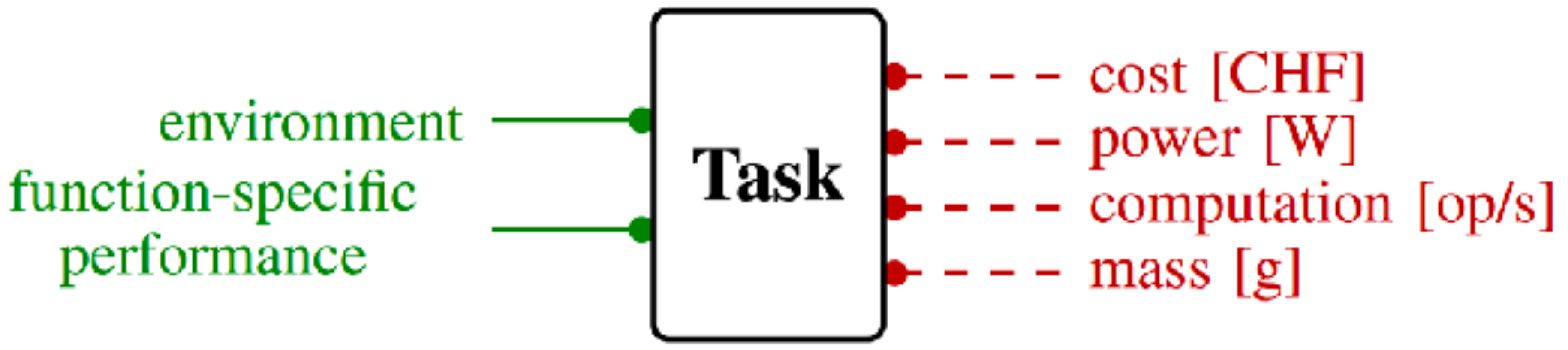
- **Systematic modeling approach:**
  - **Define the task** - *what do we need to do?*
  - **Functional decomposition** - *how to decompose the functionality?*
  - **Find components** - *decompose until you find components* (hardware and software)
  - **Find common resources** - In robotics, **size**, **weight**, **power**, **computation**, **latency** and add them.

- **Implementation:**
  - **Write a skeleton** - *write the structure using the formal language and the dependencies.*
  - **Populate the models:**

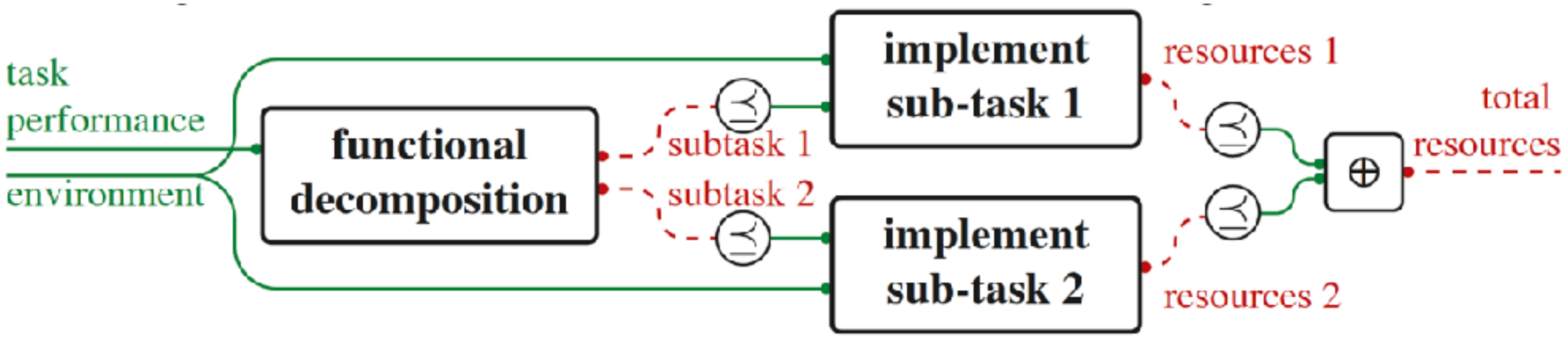
    catalogues, analytic models, simulations

Functional decomposition in autonomy

- It is useful to think of a task ("function") as a design problem:

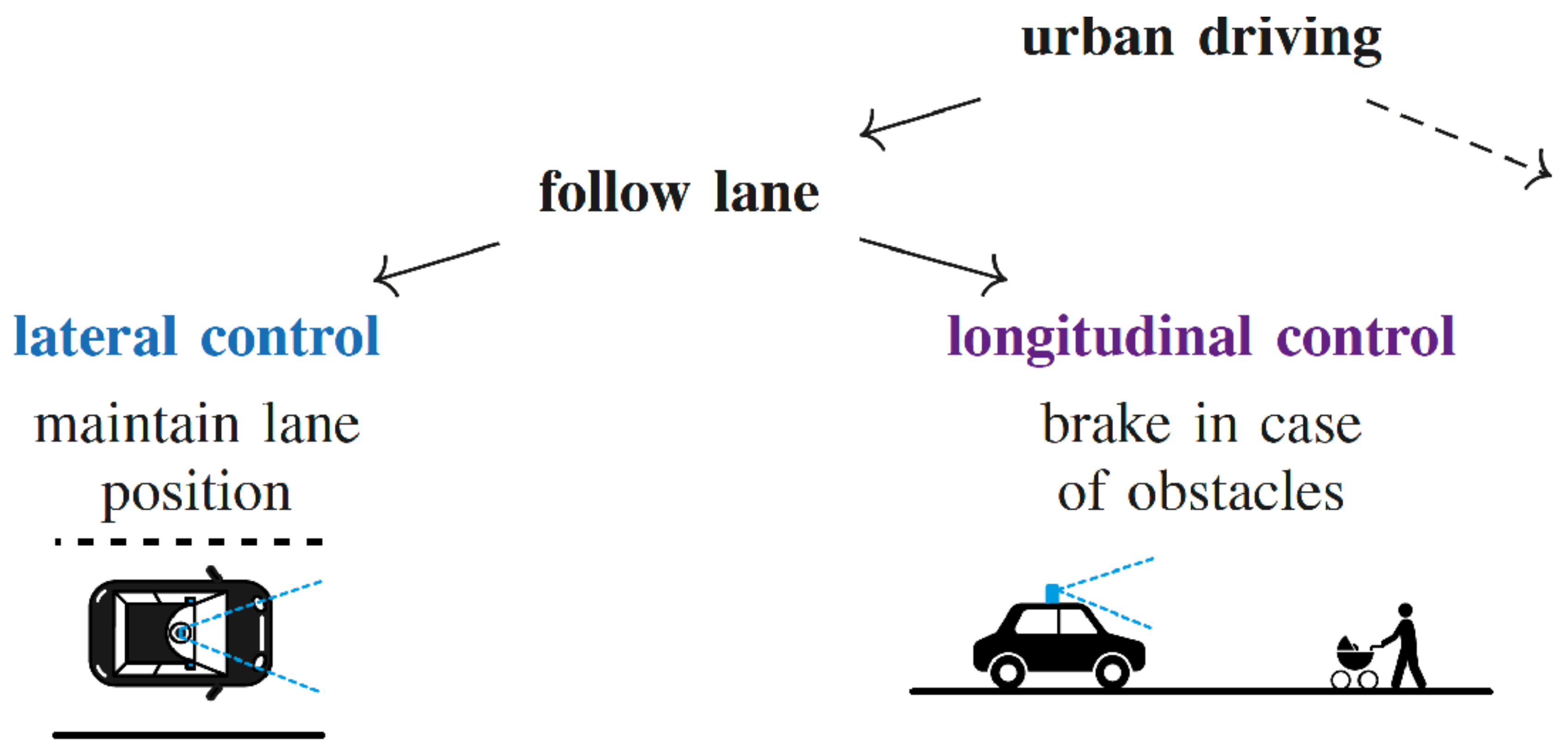


- **Functional decomposition** divides functionality and sums resources:



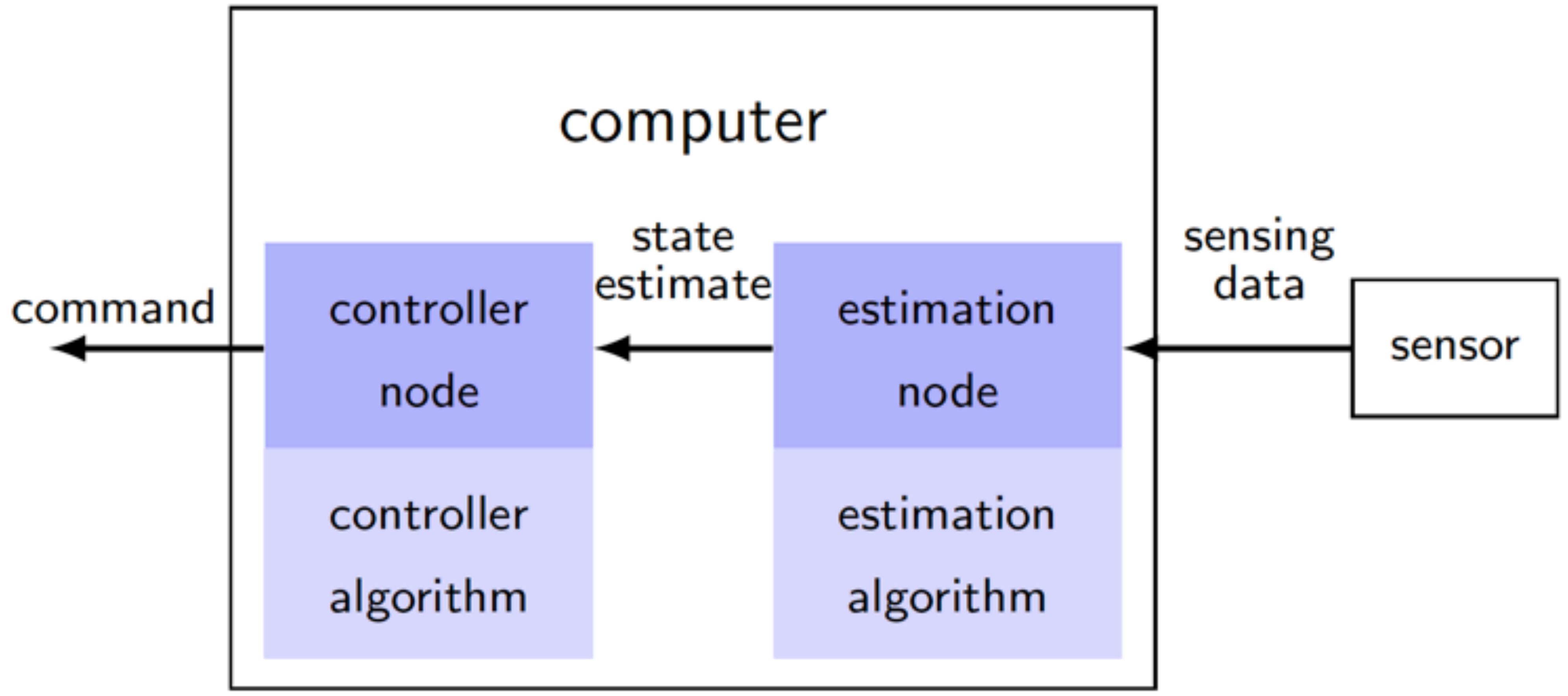
- Note that **composing** tasks returns a **task** (**compositionality**)

- In this example (**urban driving**):

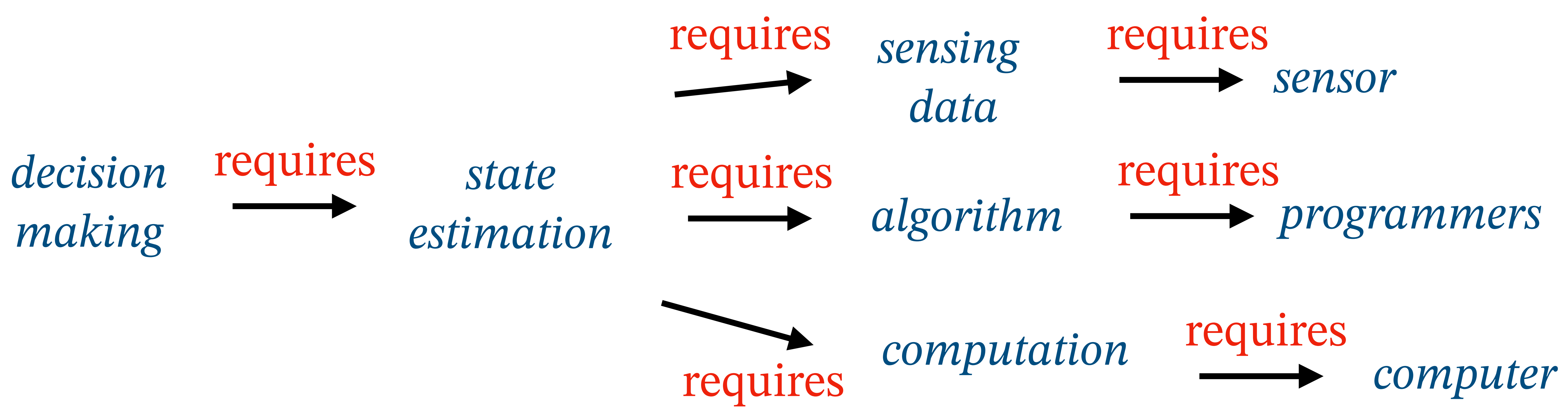


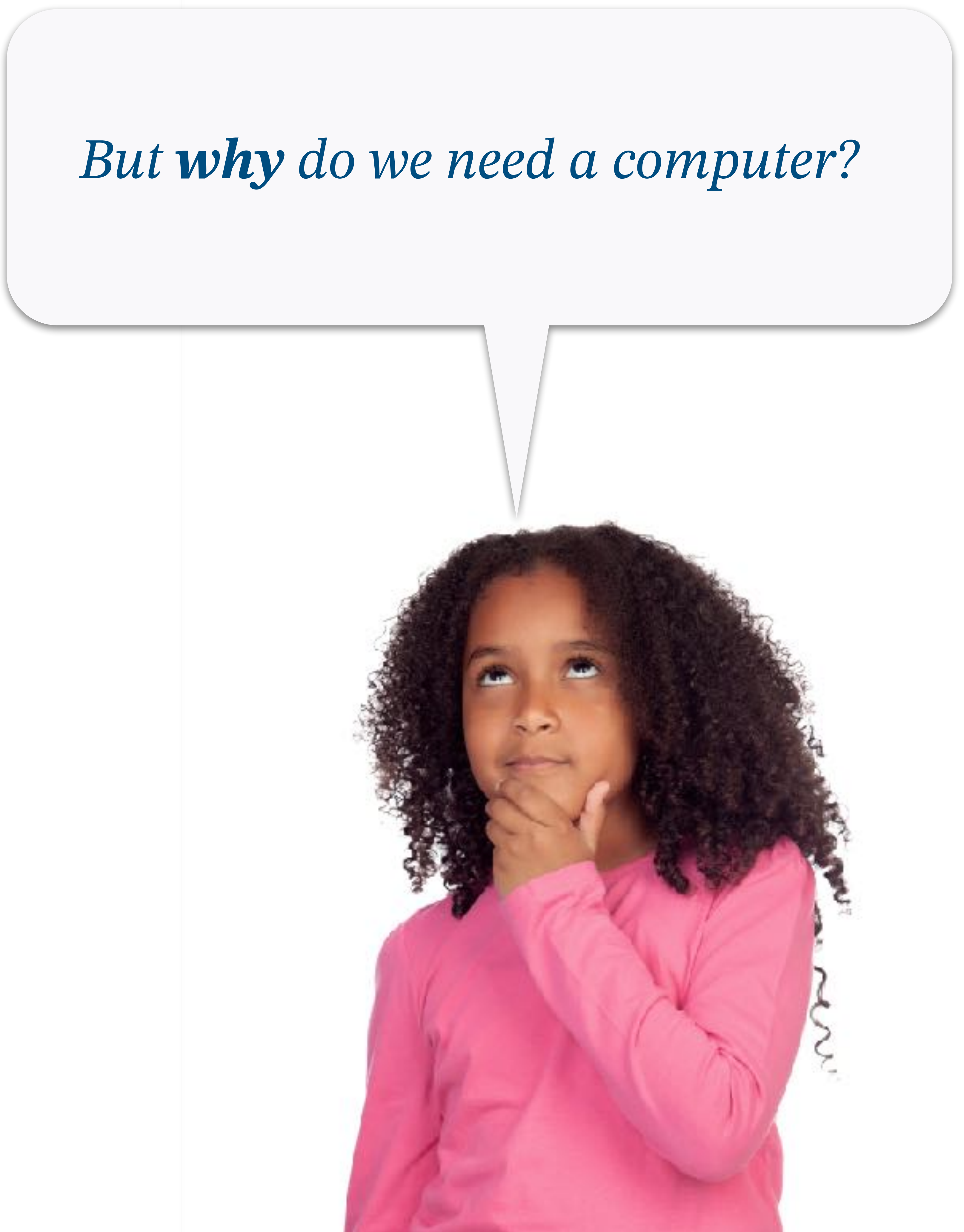
Data flow vs. logical dependencies

- In robotics, we are used to think about **data flow:**

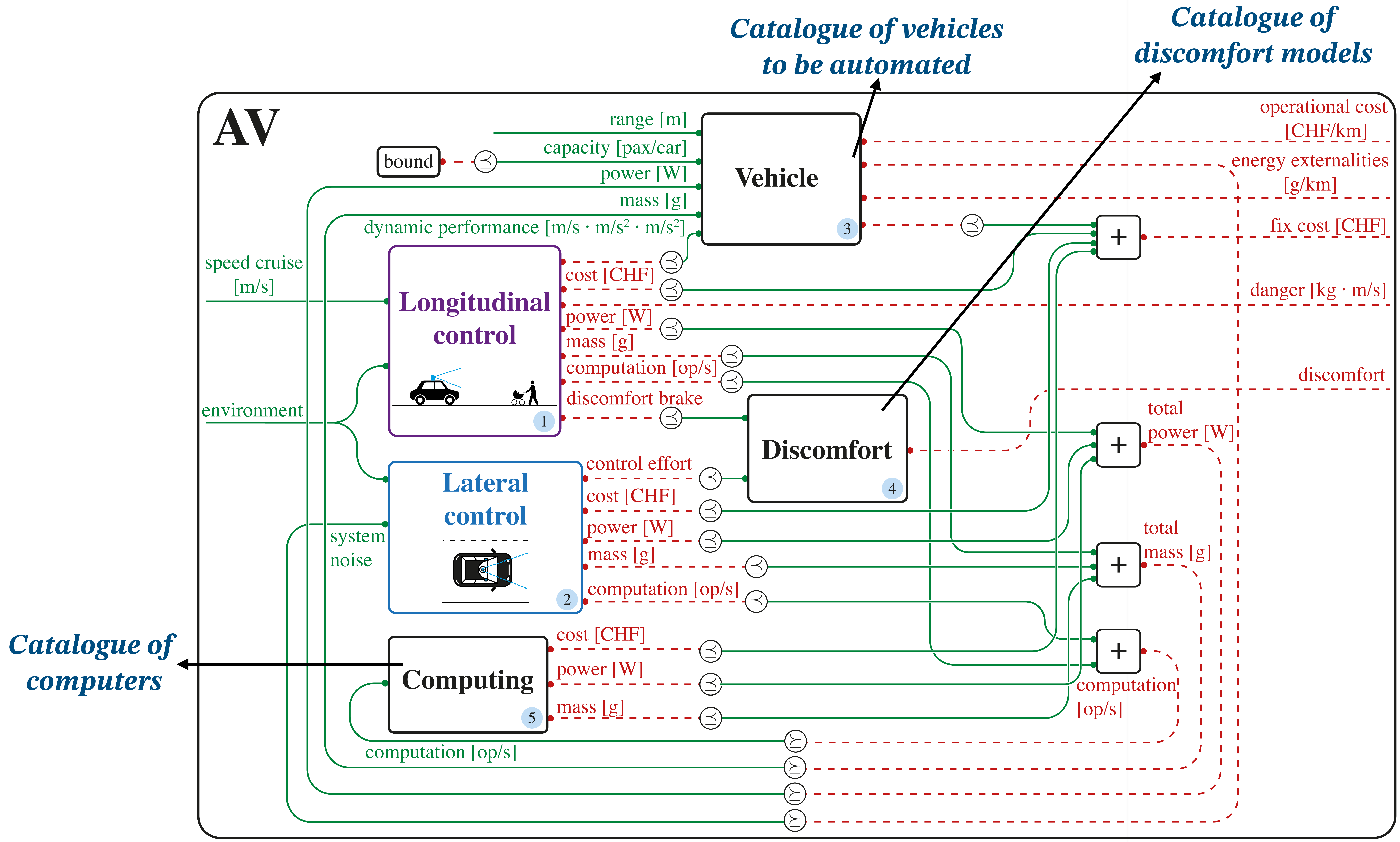


- To find **components**, it helps to reason about **logical dependencies**:

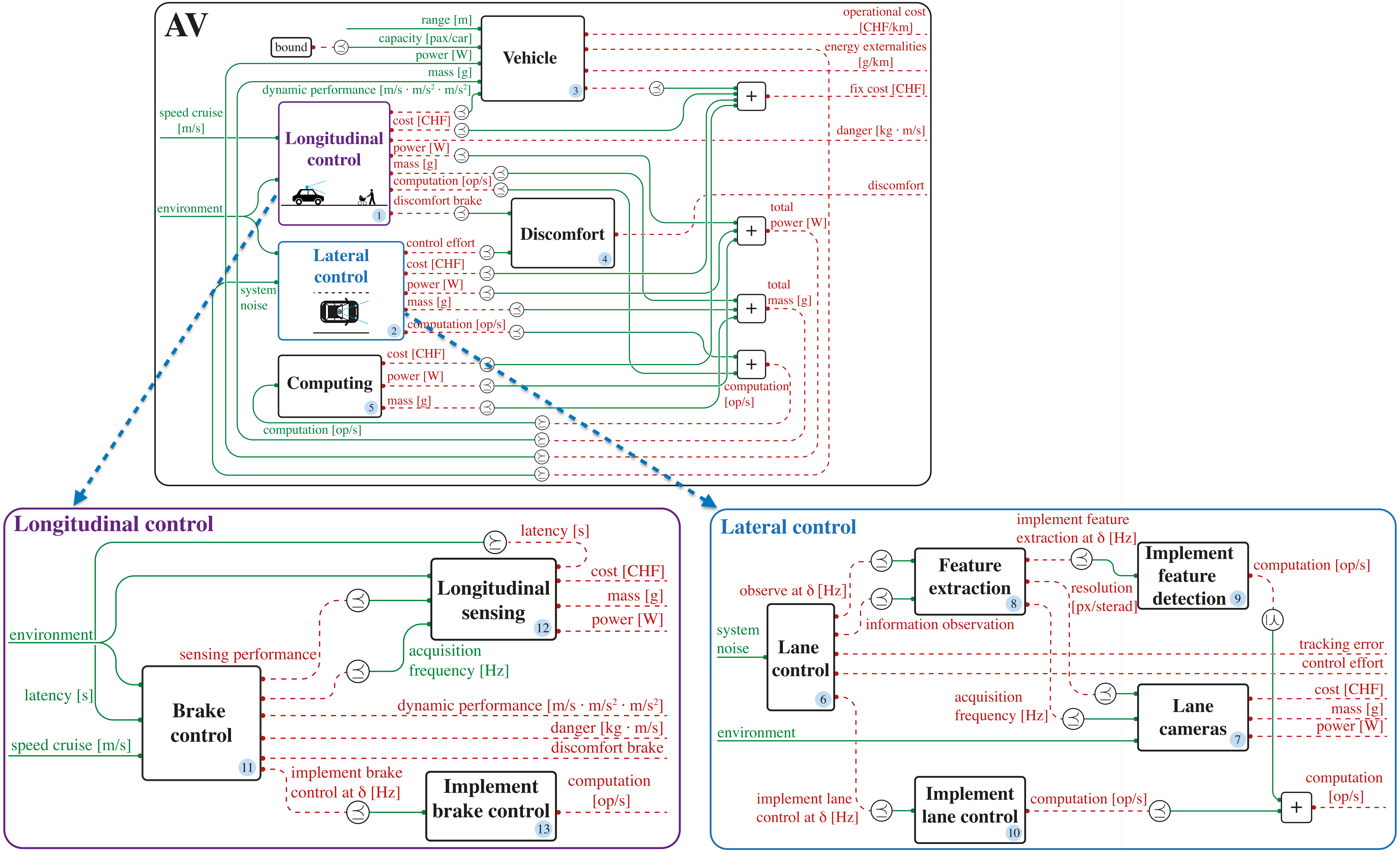




Co-design of an autonomous vehicle

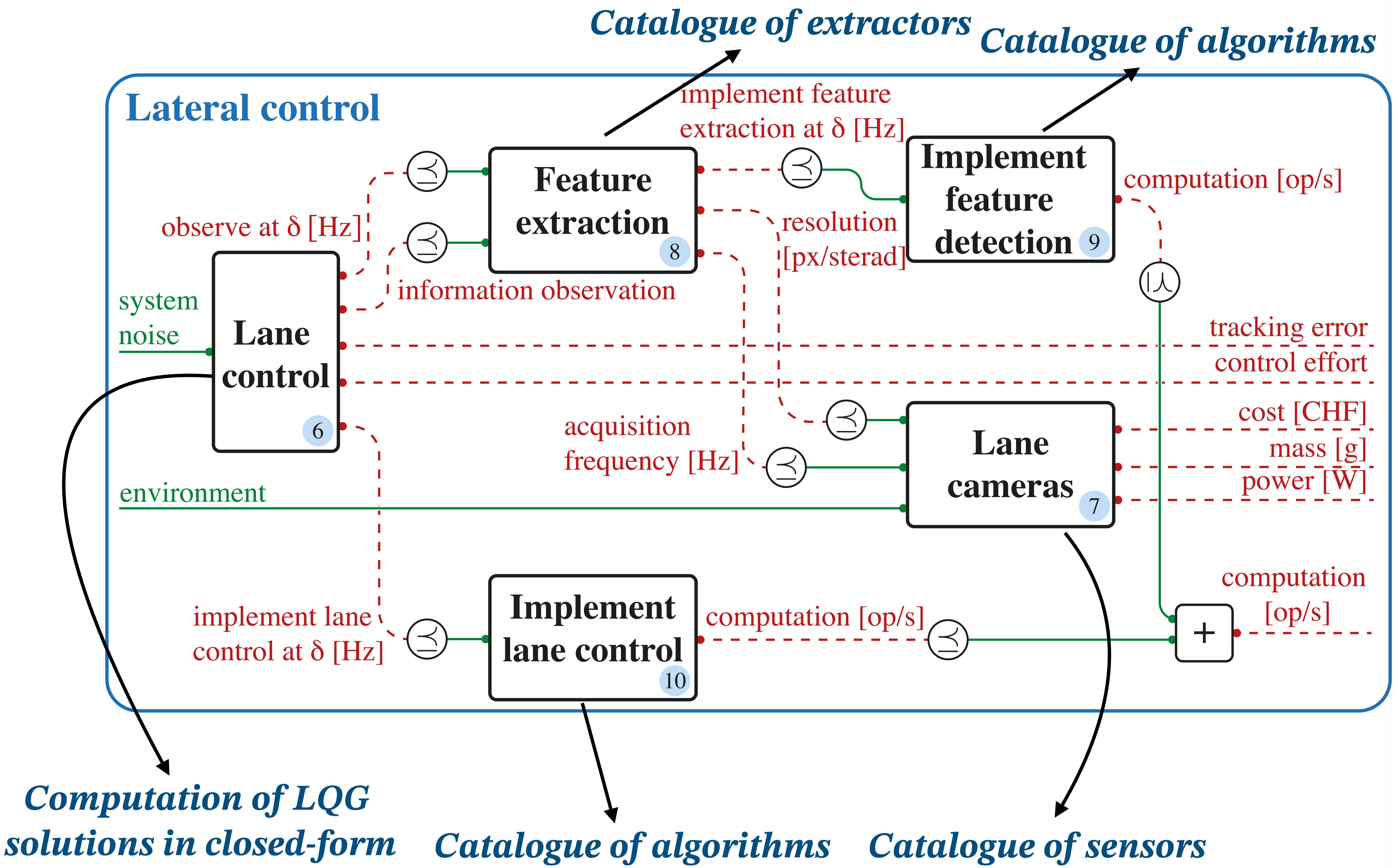


Functional decomposition



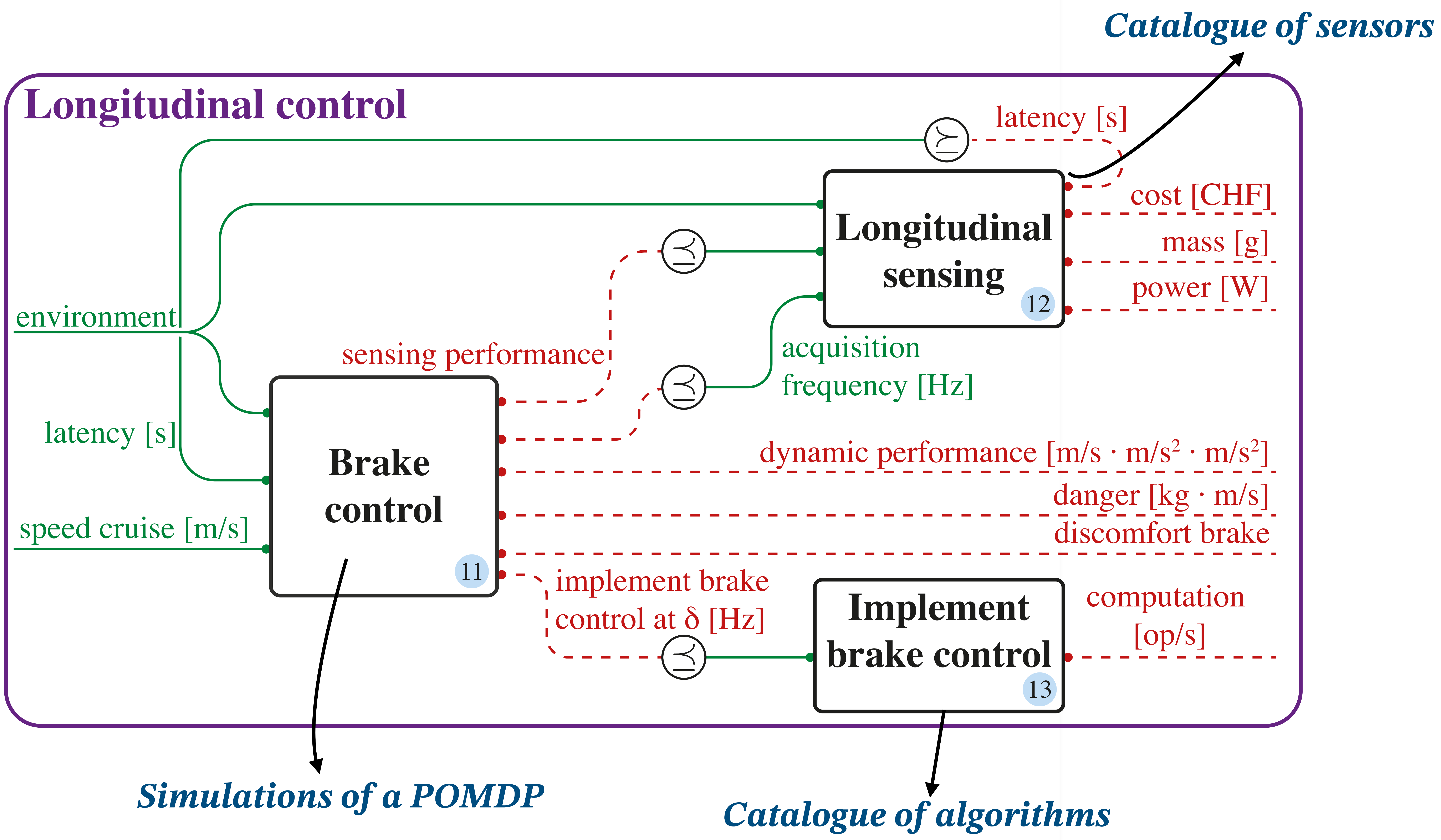
Co-design of lateral control

- Lateral control itself can decomposed in **sub-tasks:**

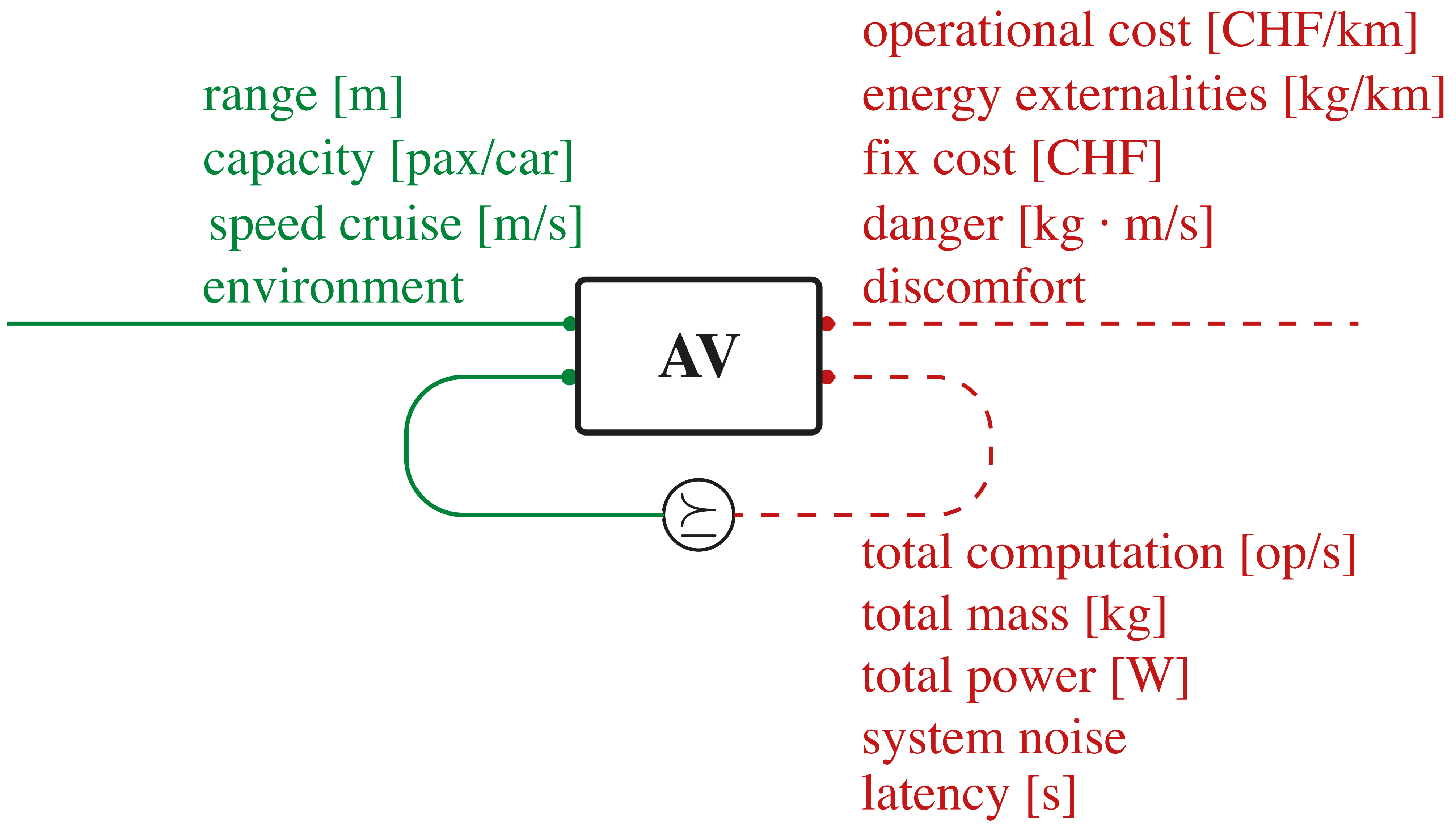


Co-design of longitudinal control

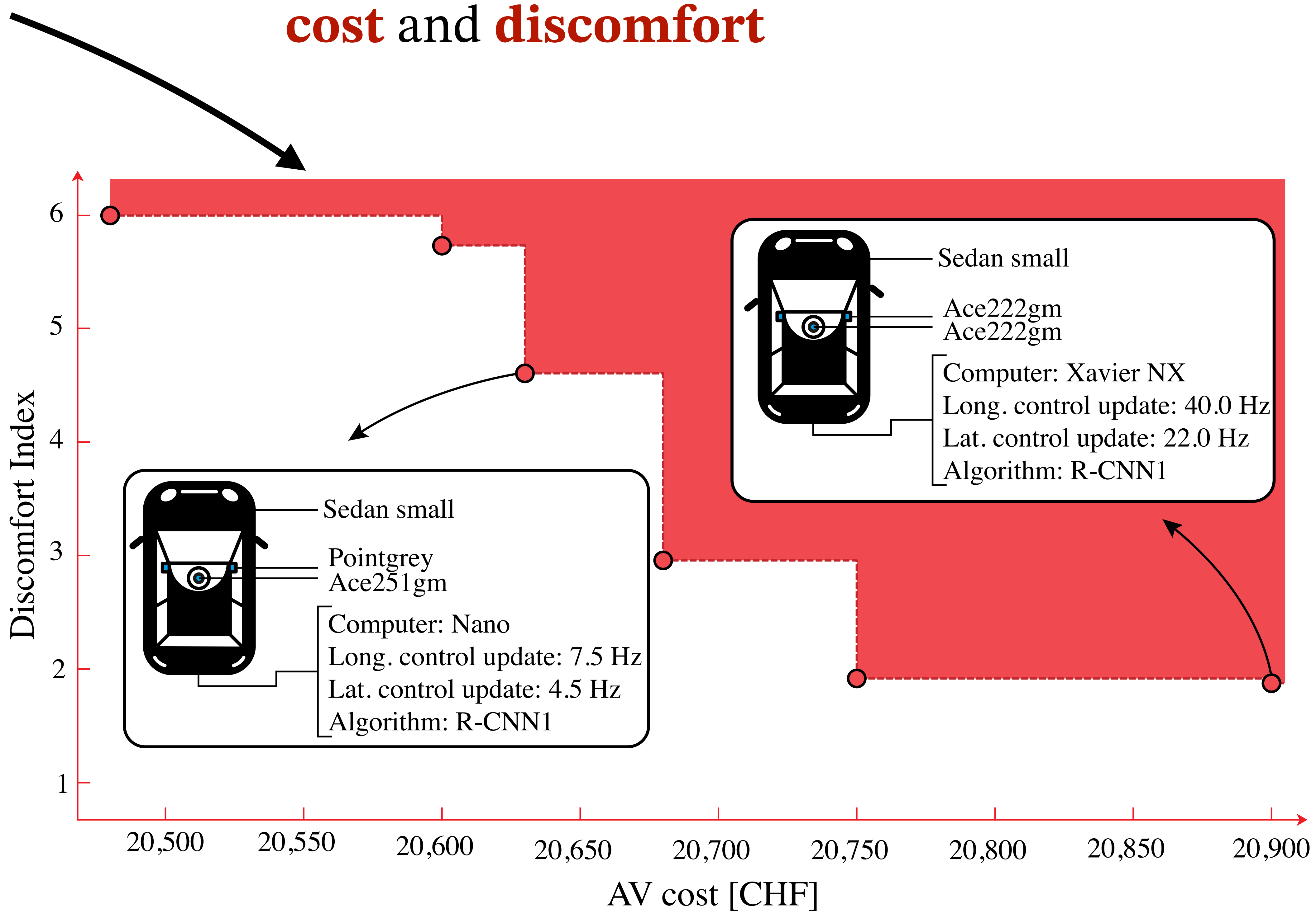
- Longitudinal control can be decomposed in **sub-tasks**:

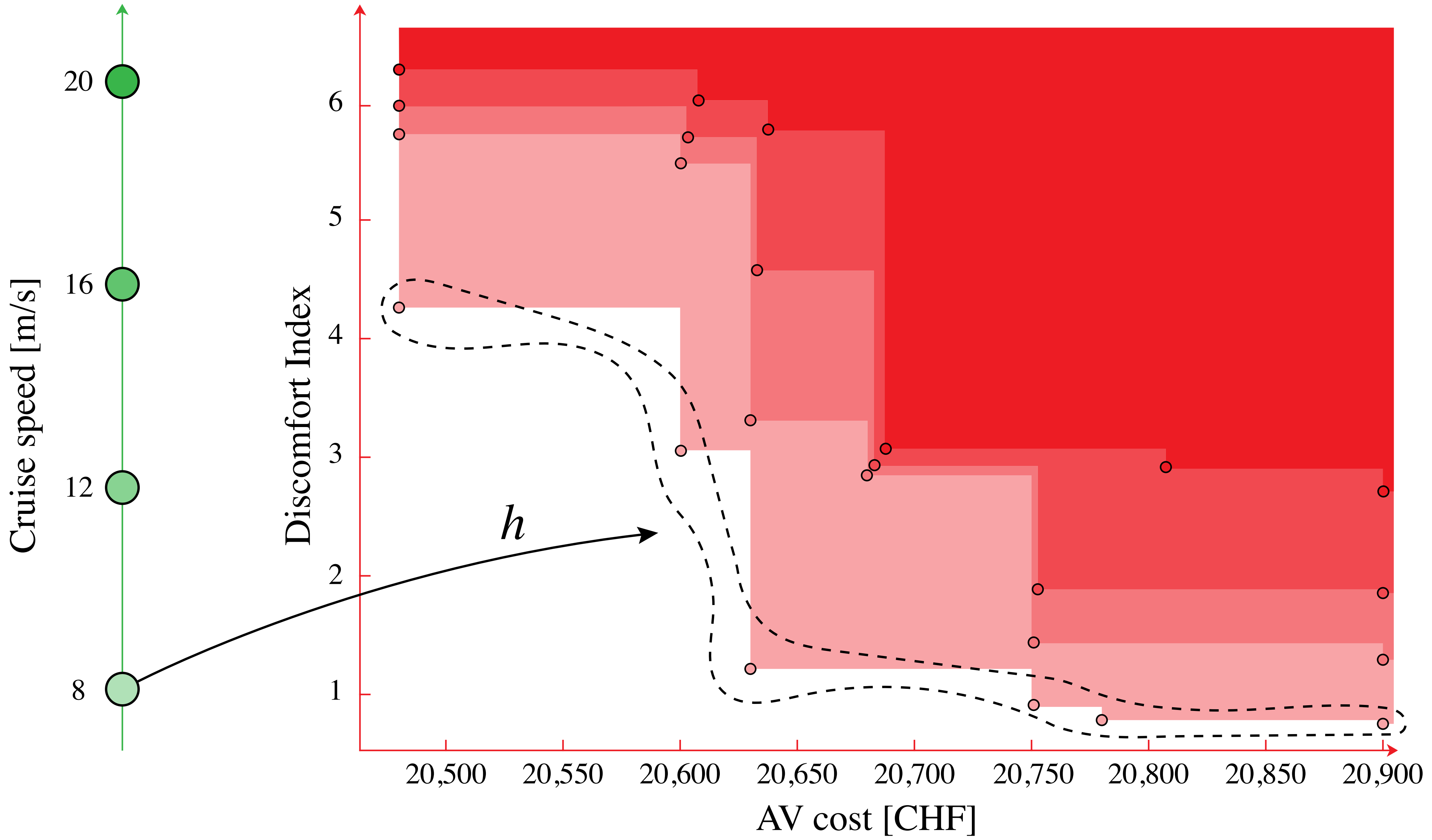


Solution of DPs



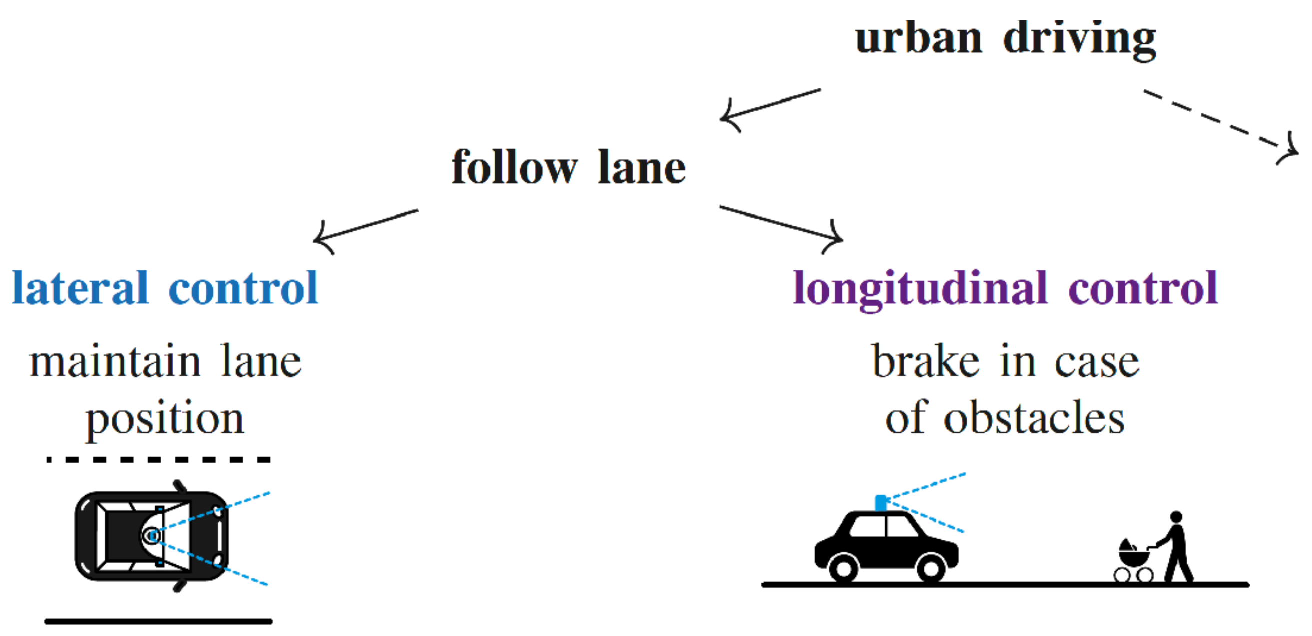
Fix **functionalities**, look at minimal **cost** and **discomfort**

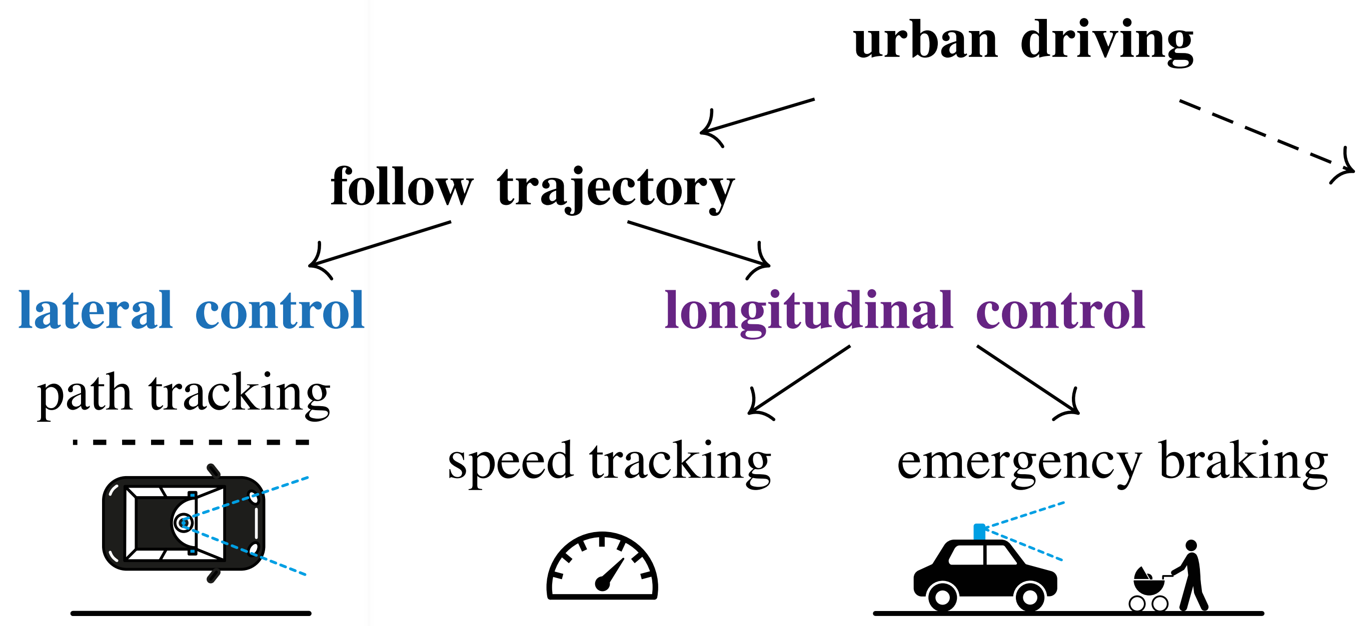


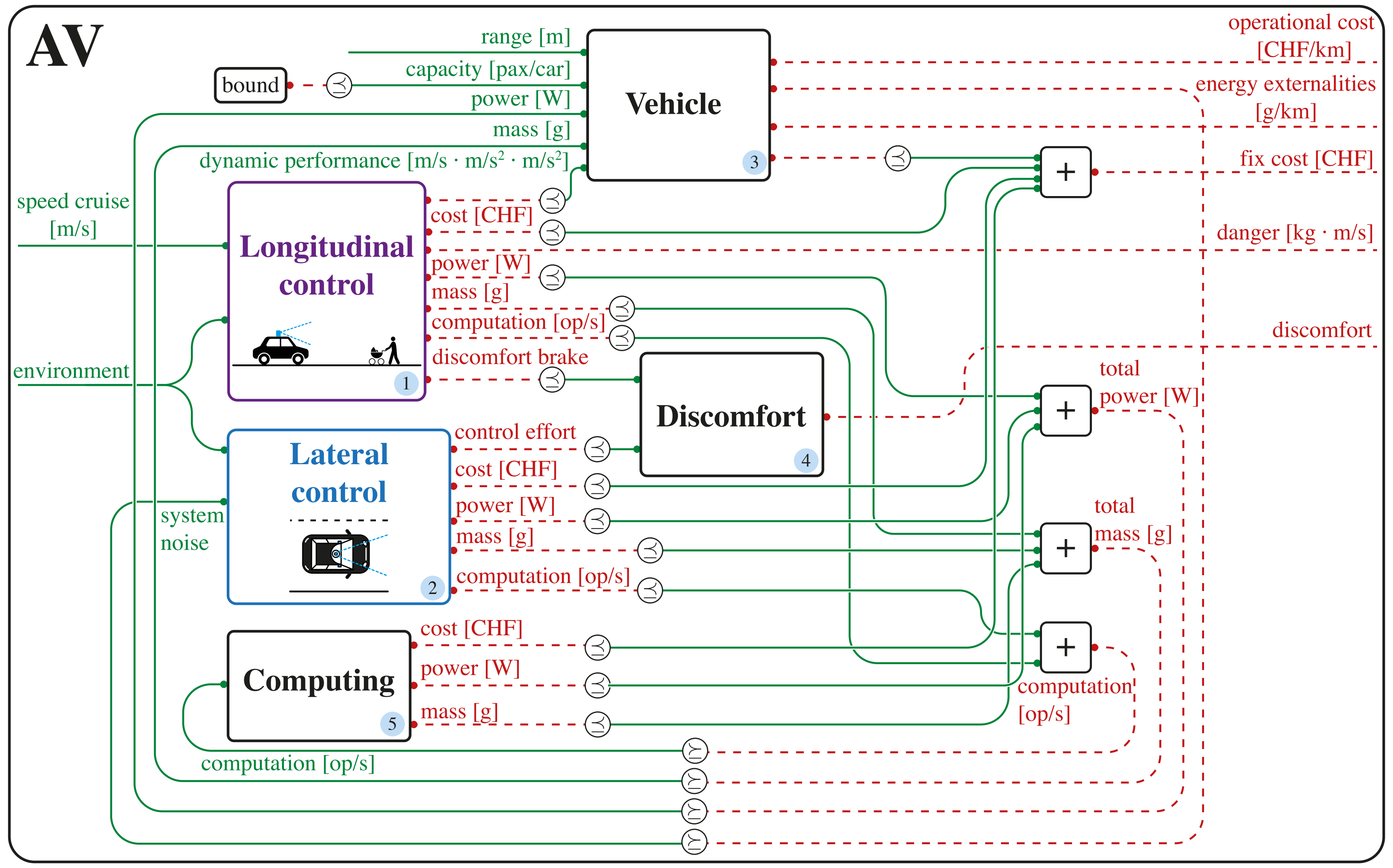


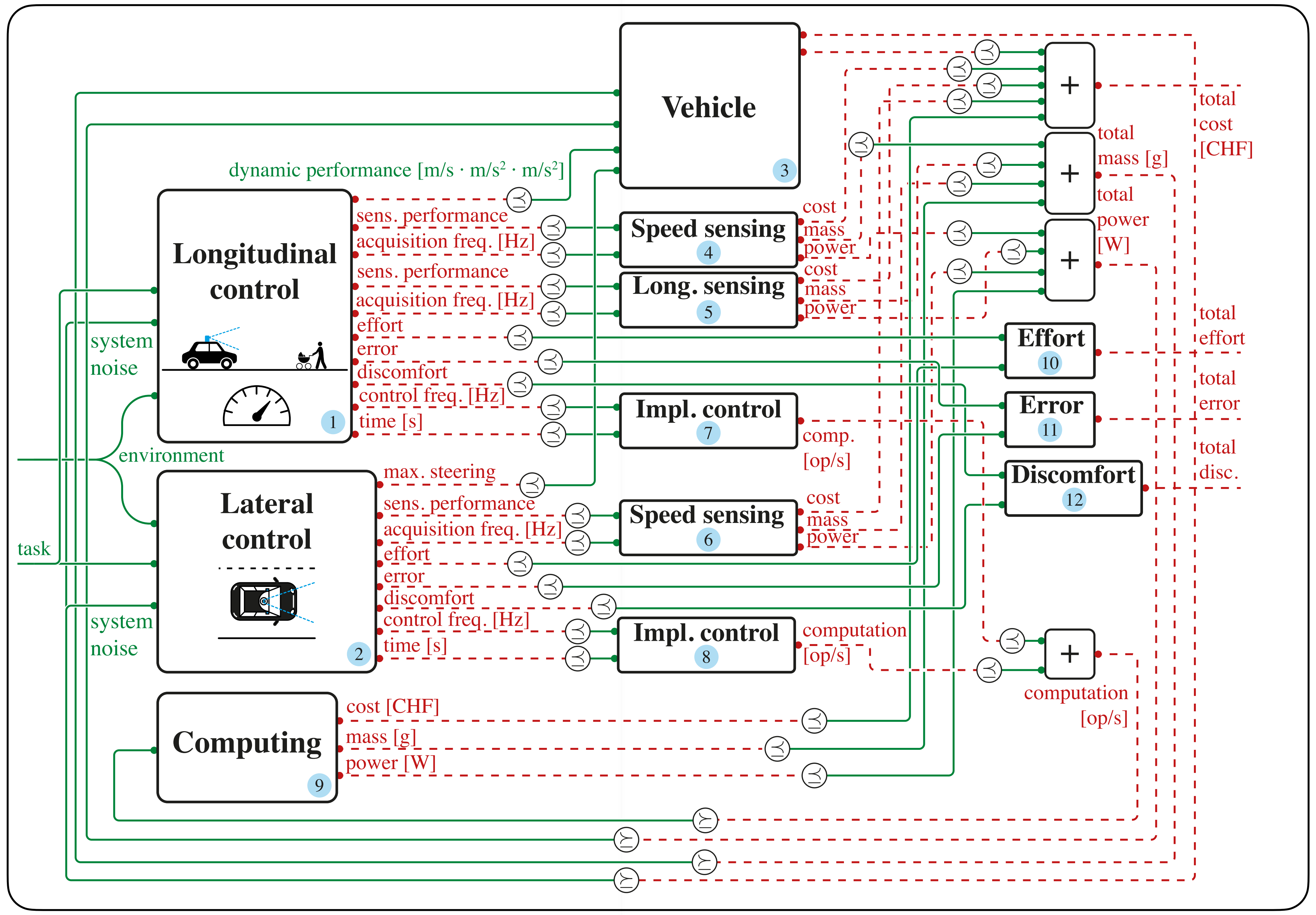
Monotonicity: *Higher achievable speeds will not require* ***less*** *resources*

Functional decompositions can be extended



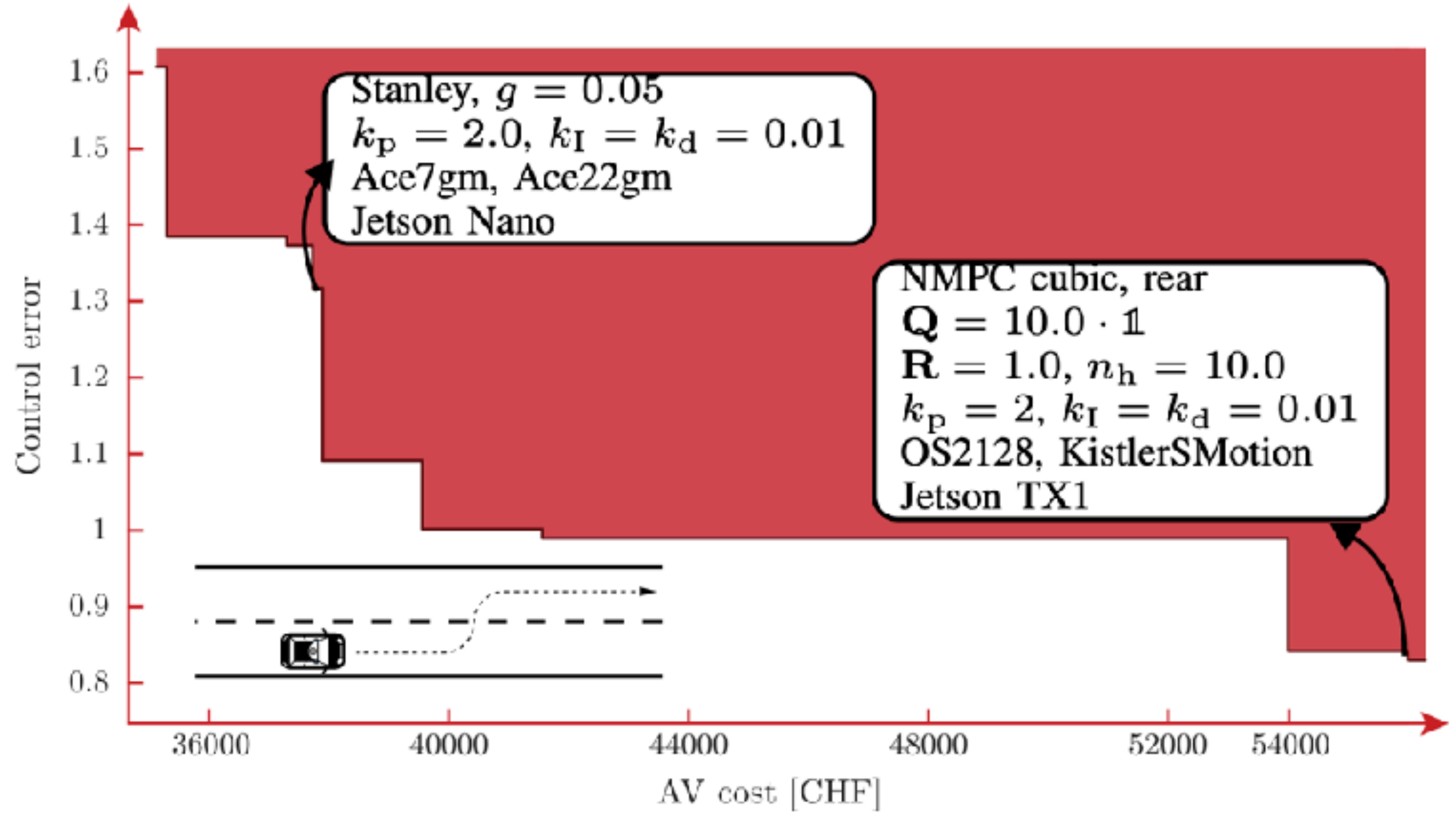


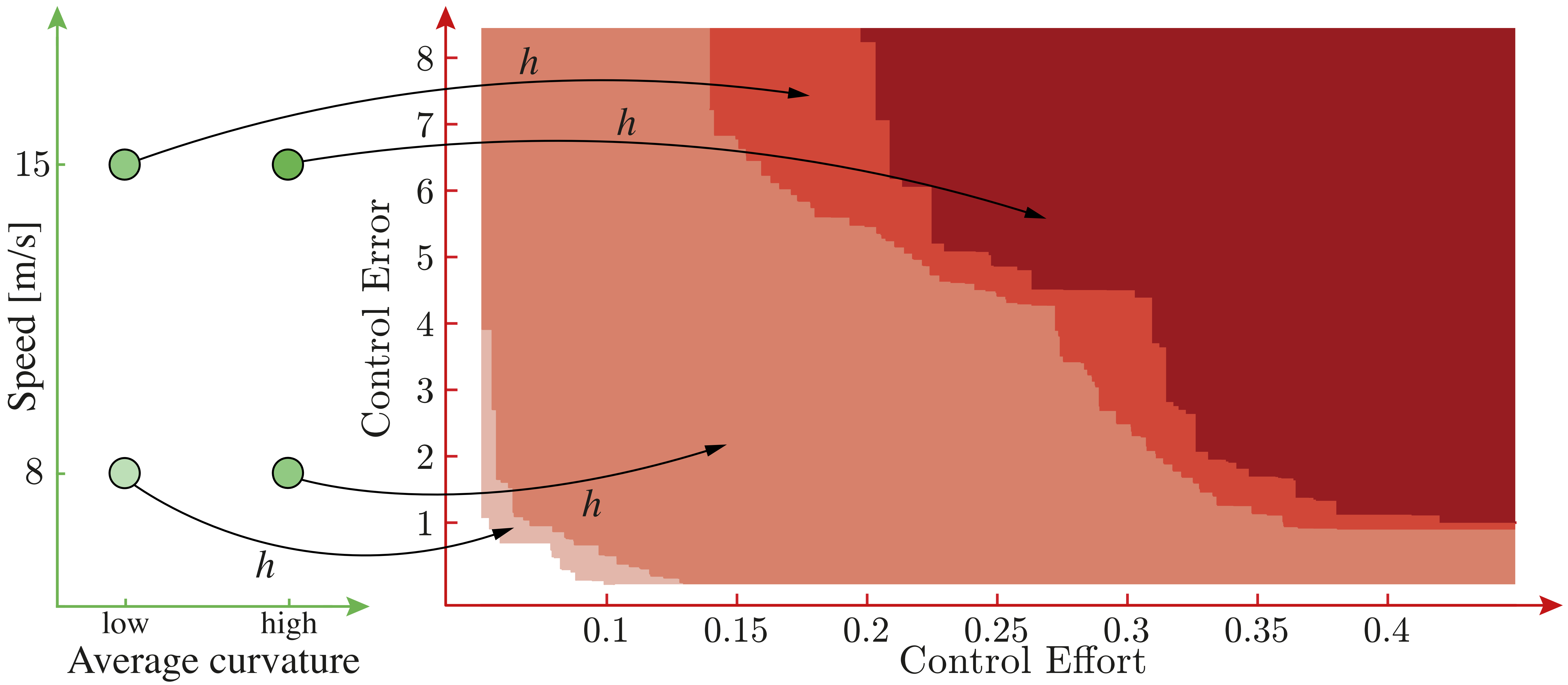




Functional decompositions can be extended

Fix an environment
Fix a task (achievable speed, kind of maneuver, curvature)





Monotonicity of task complexity

Co-design across scales: Future Mobility

- We look at the problem from the perspective of **municipalities** and **policy makers**
  - Important decisions to make:

    How many AVs should we allow? *What's the influence of AVs on public transit systems?*

    How performant should they be? *How many trains should we buy?*

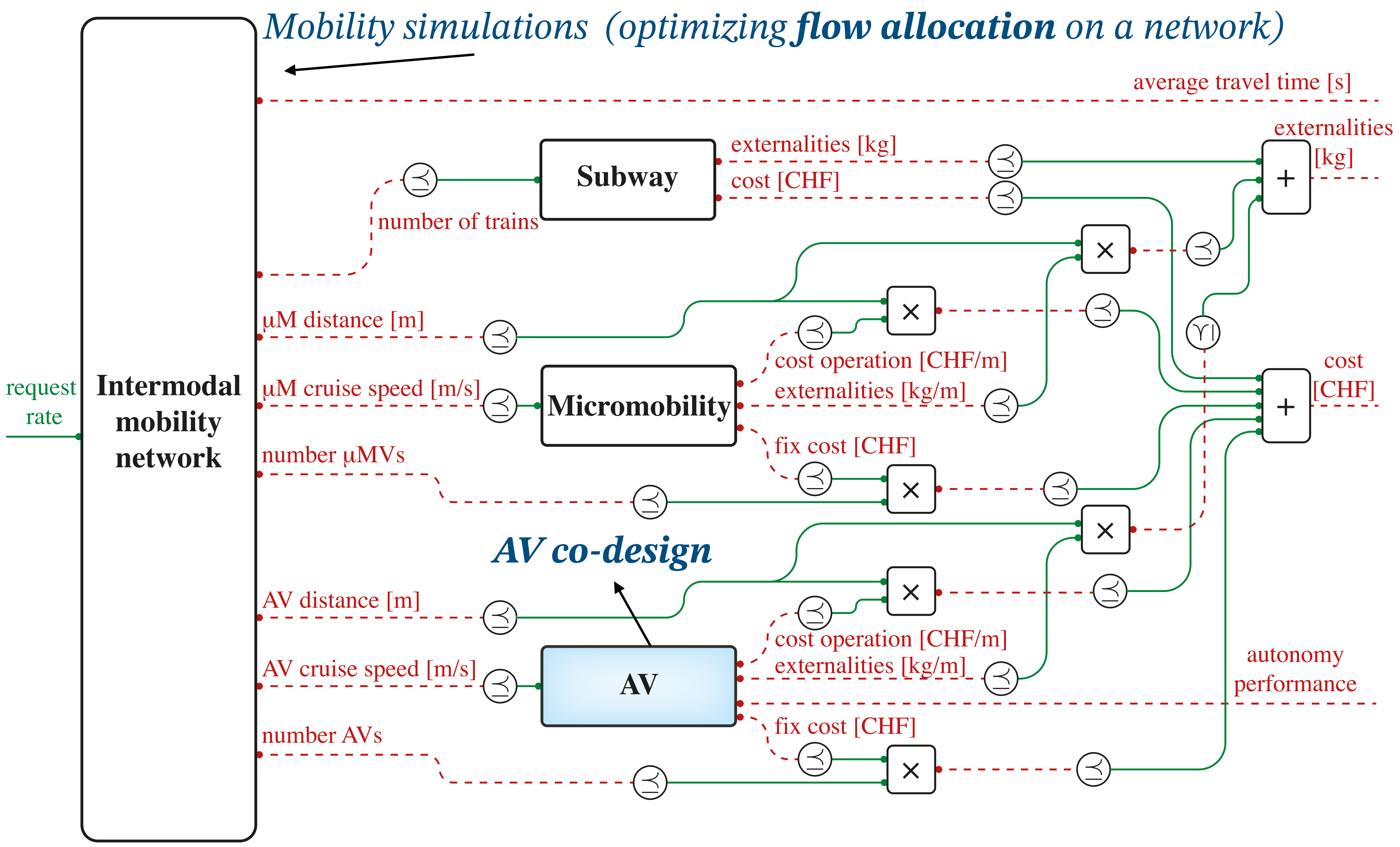
- Existing work only solves **specific problems** and does not **co-design** the whole system:
  - No **joint** design of **mobility solutions** and the **system** they enable
  - No **modularity** and **compositionality**: problem-specific
  - Often, not producing **actionable information** for stakeholders

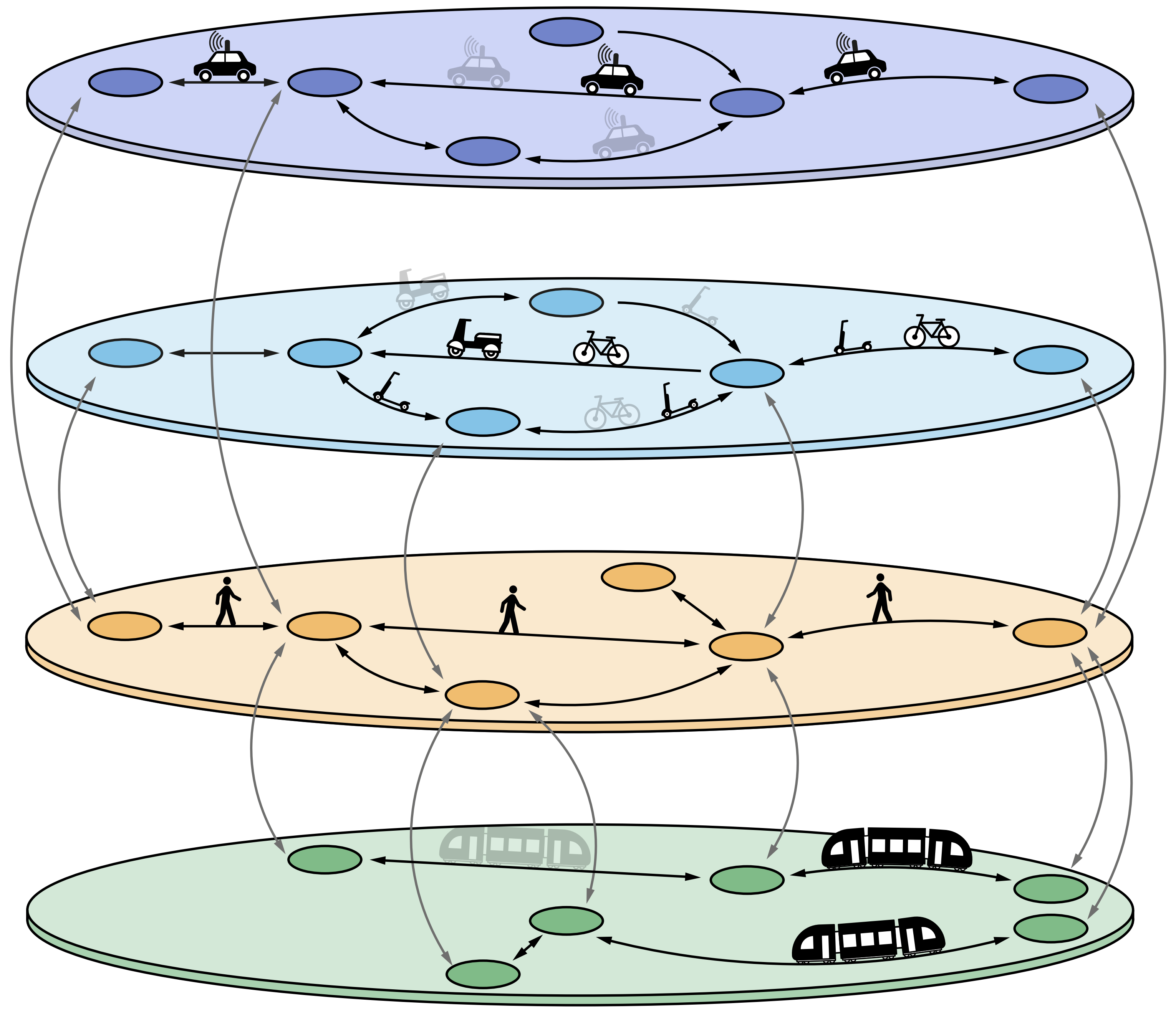
- Several **disciplines** involved (transportation science, autonomy, economics, policy-making)

- We allow **interfaces** between them via **co-design**:
  - **Functionality**: **demand** to be satisfied
  - **Costs**: **investments** ($), **externalities** ($CO_2$ kg), **service level** (average waiting time, s)

- Co-design highlights the **structure** of the problem and provides **tools** to reason about it

Mobility system co-design



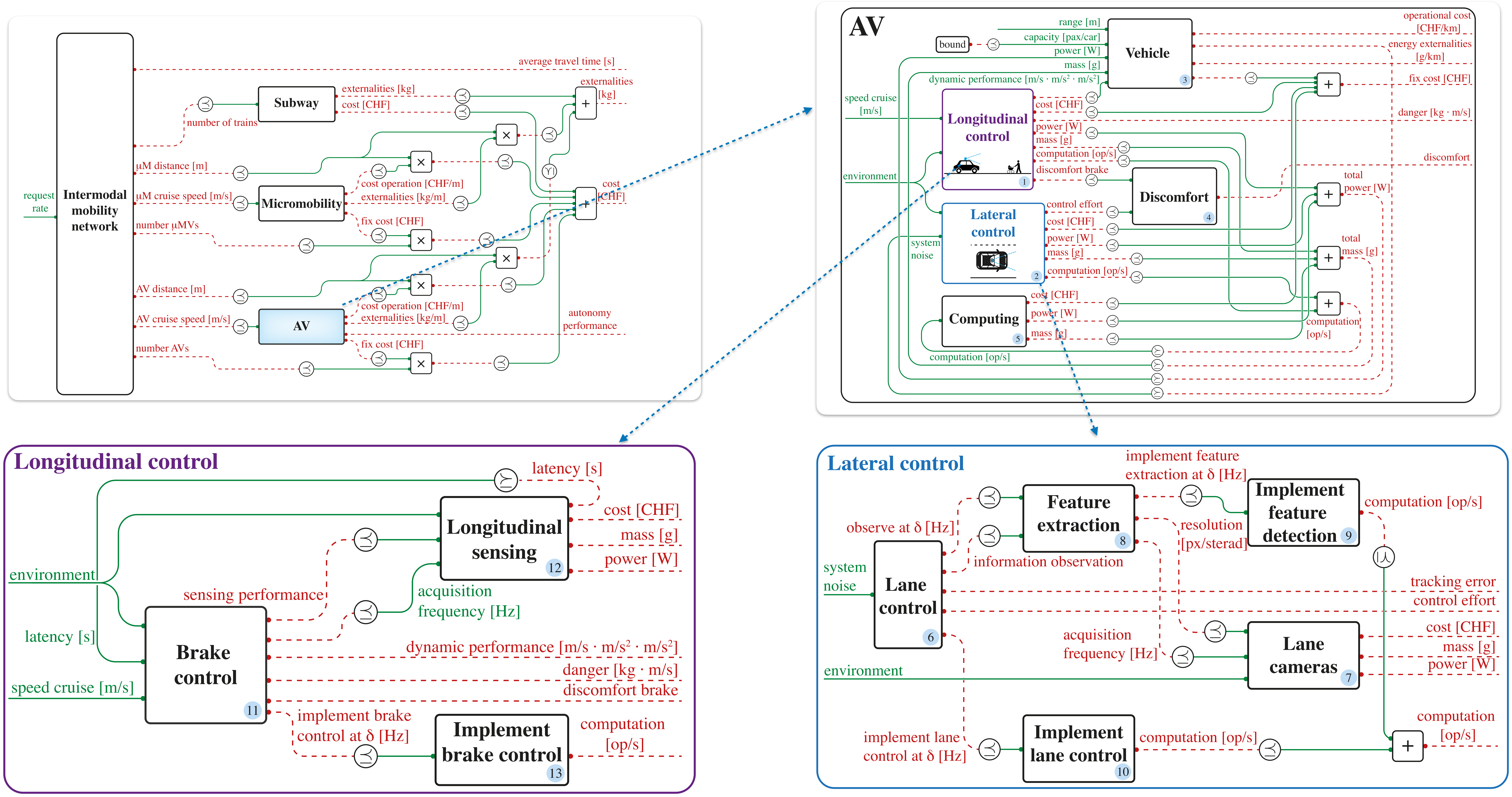


Subway:
Fun: *number* of trains to buy
Res: *costs* and *externalities*
Imp: acquisition *contracts*

Micro mobility:
Fun: *cruise speed*
Res: costs and *externalities*
Imp: vehicle *models*

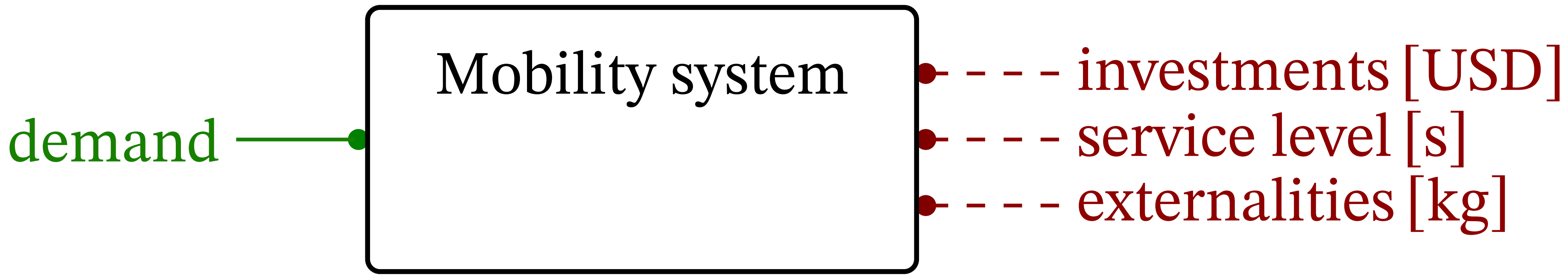
AV:
Fun: *cruise speed*
Res: costs, *externalities, performance*
Imp: vehicle *models* and autonomy

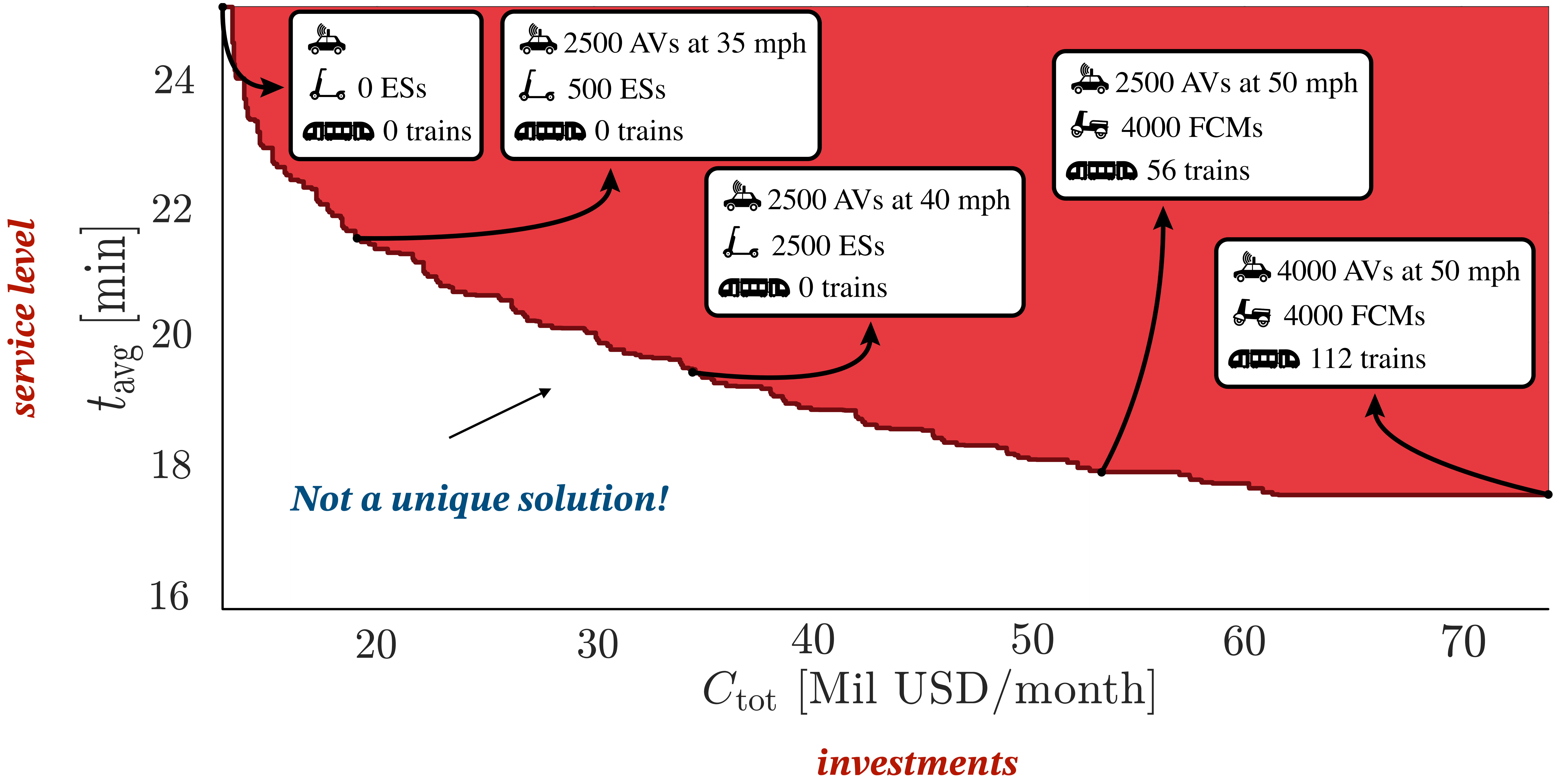
We can explode the model of the mobility system, and model AVs



Mobility systems co-design

Fixed a **demand**, we find the **Pareto front** of **incomparable**, **minimal solutions** as **cost**, **time**, and **externalities (CO_2)**





Results for real world case study of Washington D.C.

Which one is the best? Depends on what is at the upper level (policy-making, etc.)

A lot of applications …

Embodied intelligence

Autonomy-enabling Infrastructure

Control & Perception

Planning & Perception

Task-driven design of swarms of robots

Resource-aware computation

Nanorobot design for cancer treatment

Automated soft-robot design

Optimal Manufacturing

Electric motors design

If you come up with other applications, let's chat!

Outlook

- We are in the **evangelization phase:**
  - We are writing divulgatory materials (textbook, classes).
  - We are **looking for case studies.**

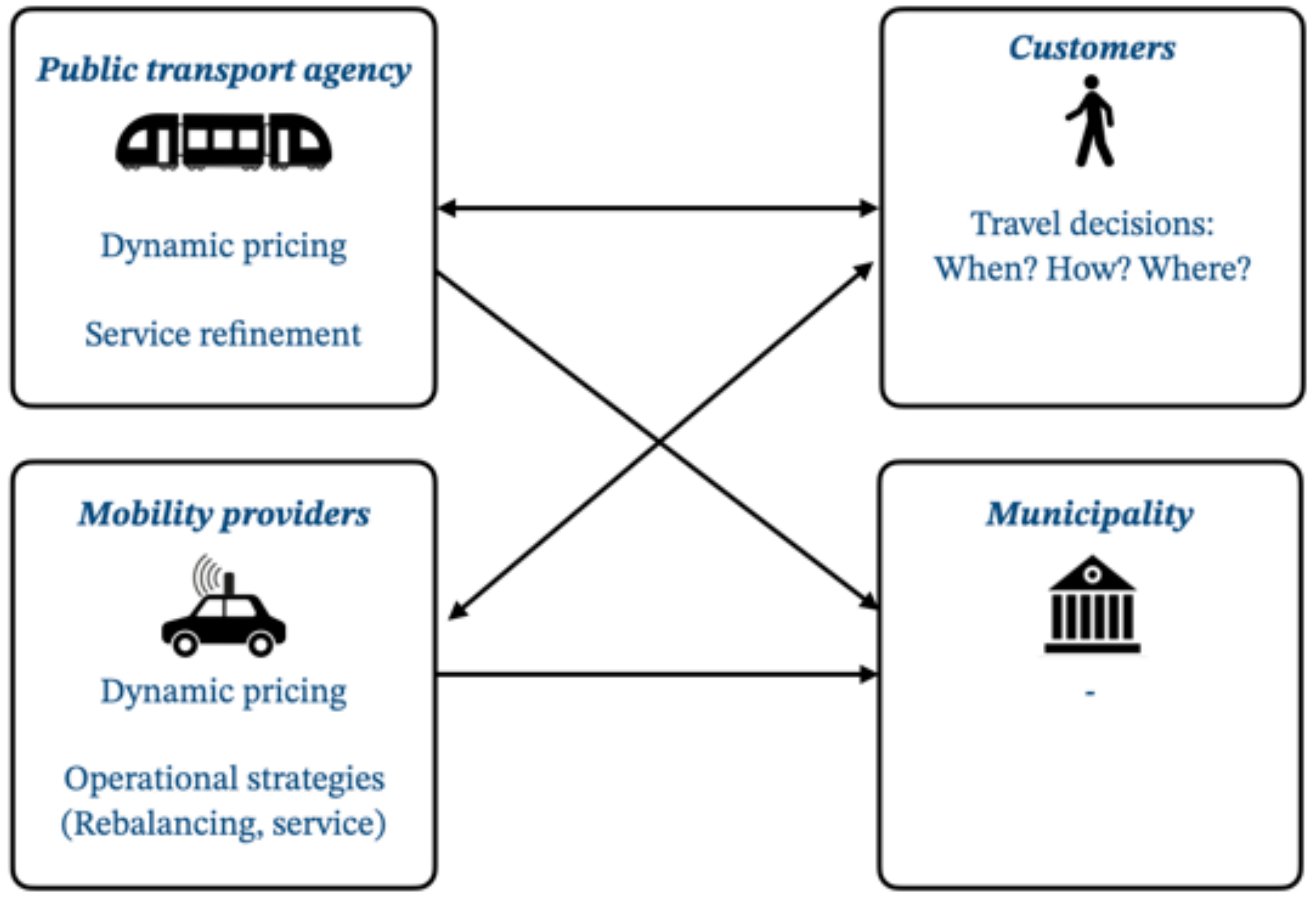
- **Algorithmics:**
  - A lot to do to make algorithms more efficient...
  - How to best change the **approximation** of each model **adaptively** and **dynamically**?

- **Theory:**
  - Finishing the rewrite in category theory.
  - Add **space** and **time** to the resources calculus.
  - Define **game semantics** (multiple agents).

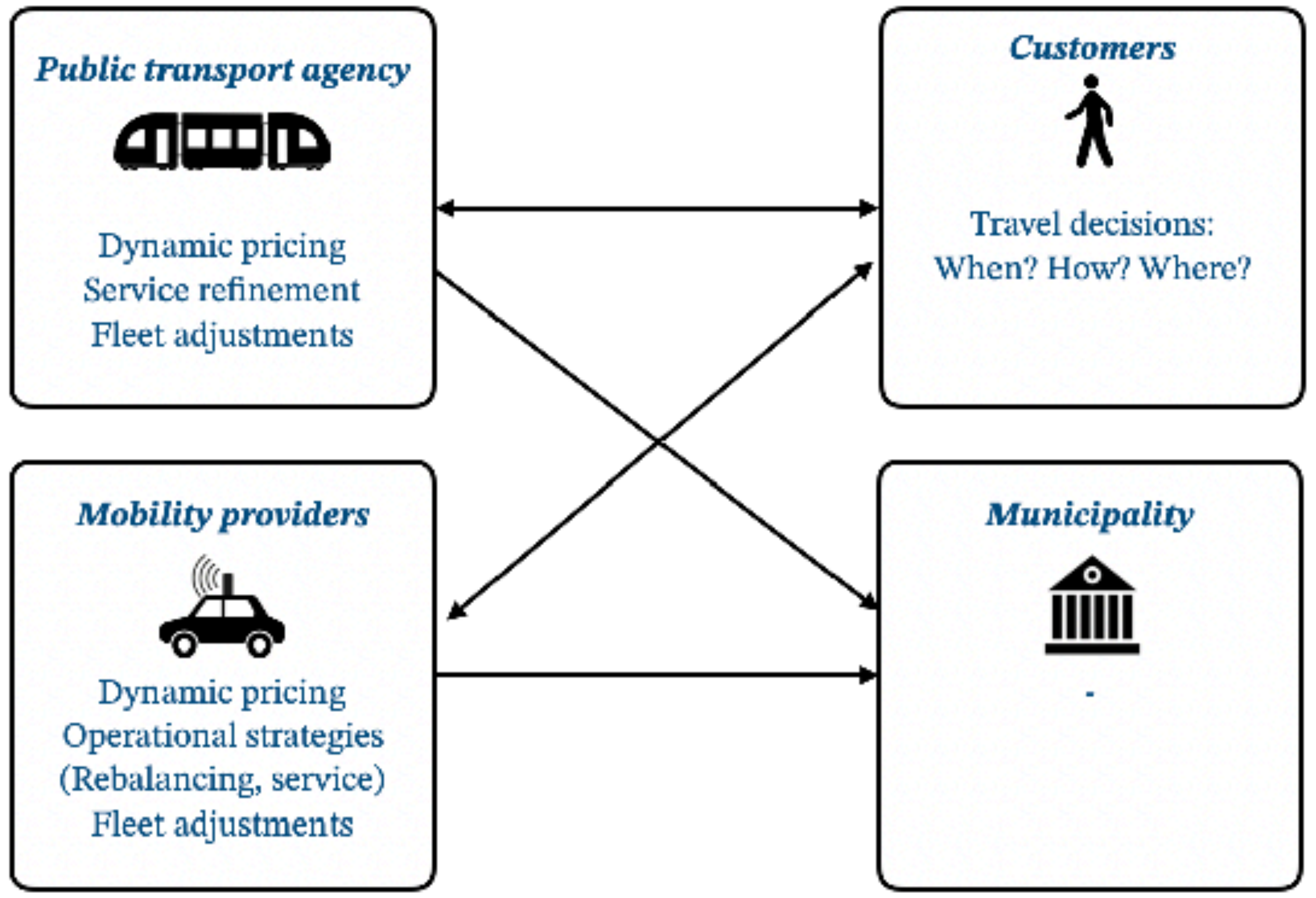
    Will merge DP with *linear logic*.

Interactions between stakeholders are characterized by different time horizons

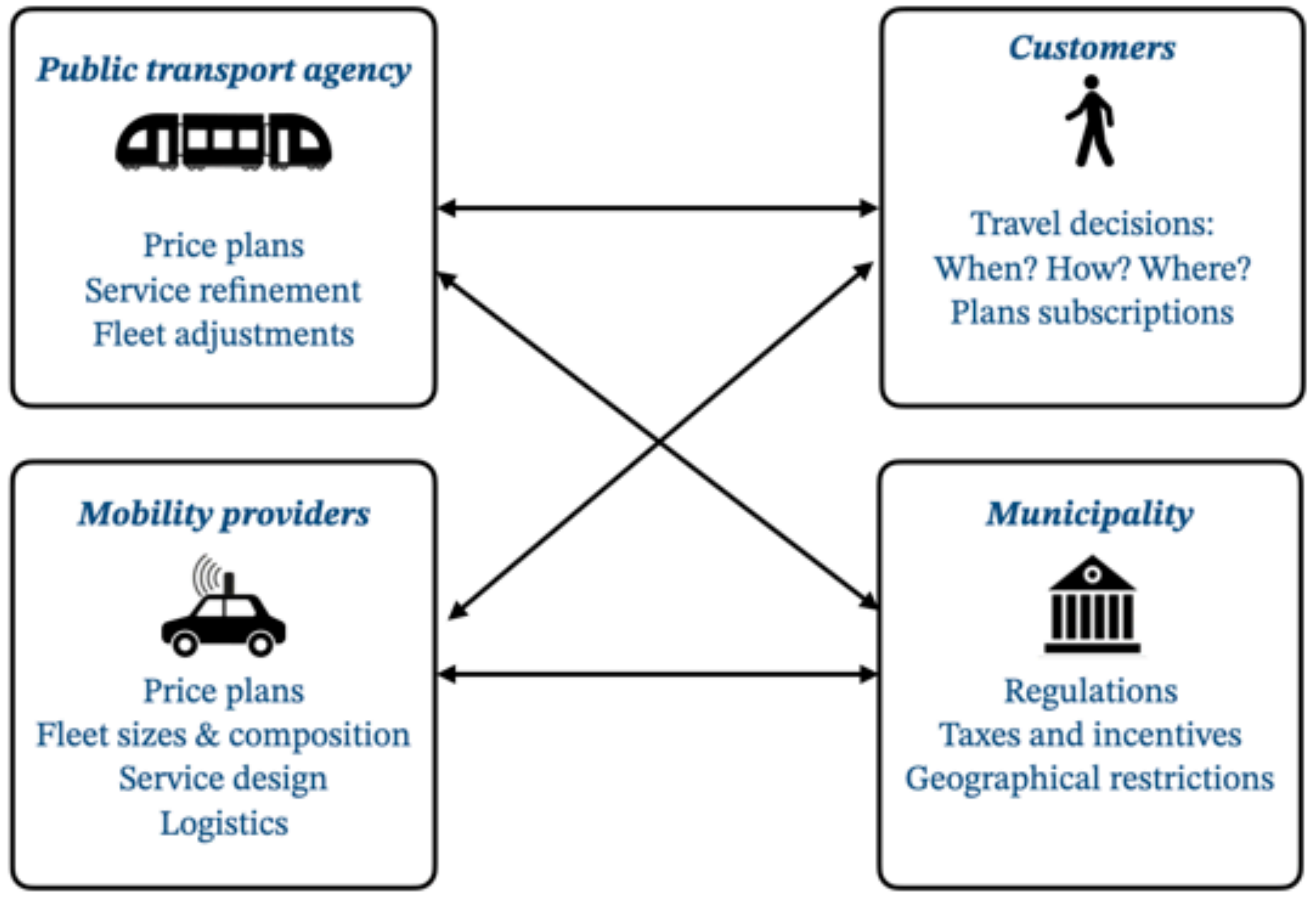
Daily



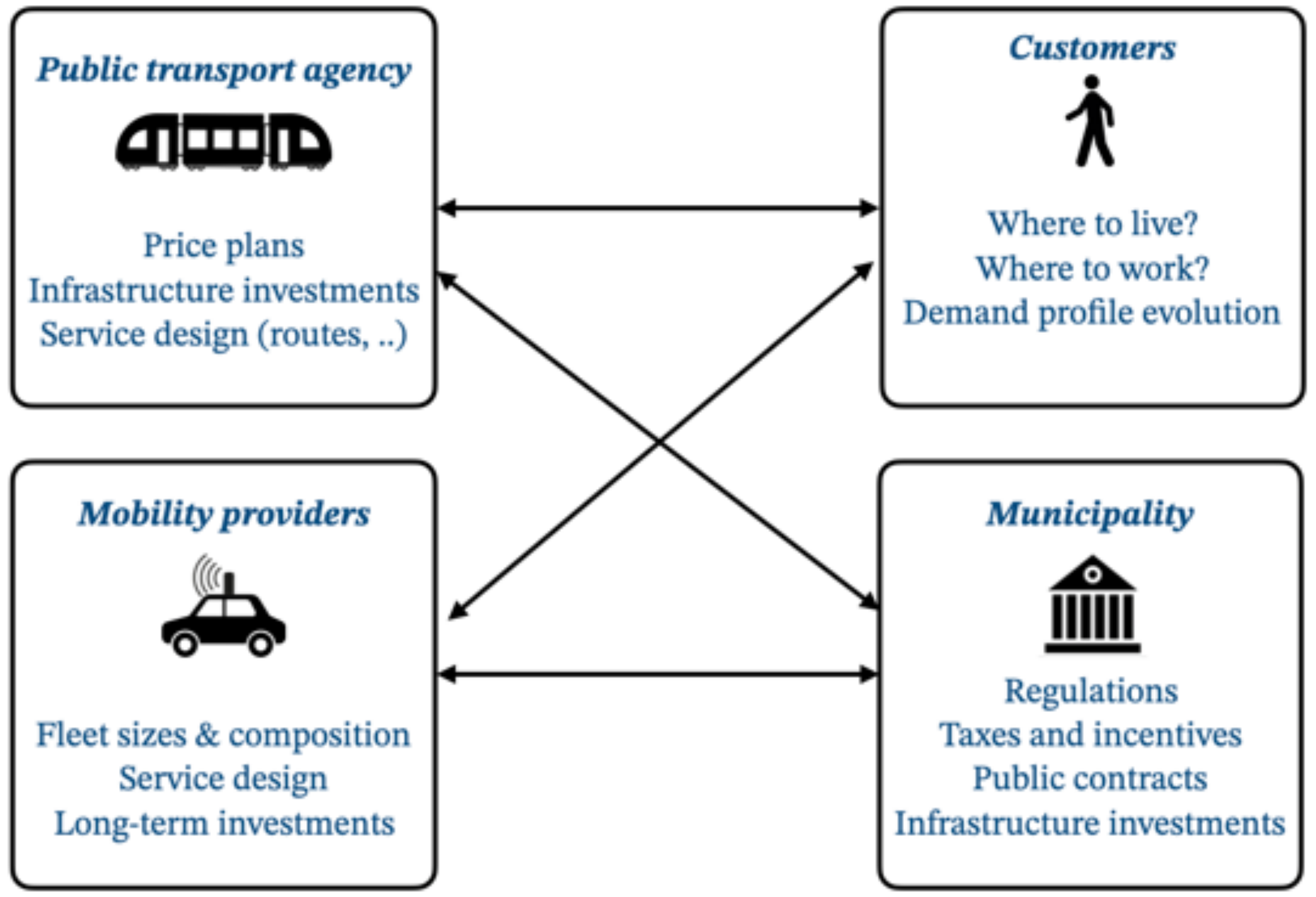
Monthly



Yearly

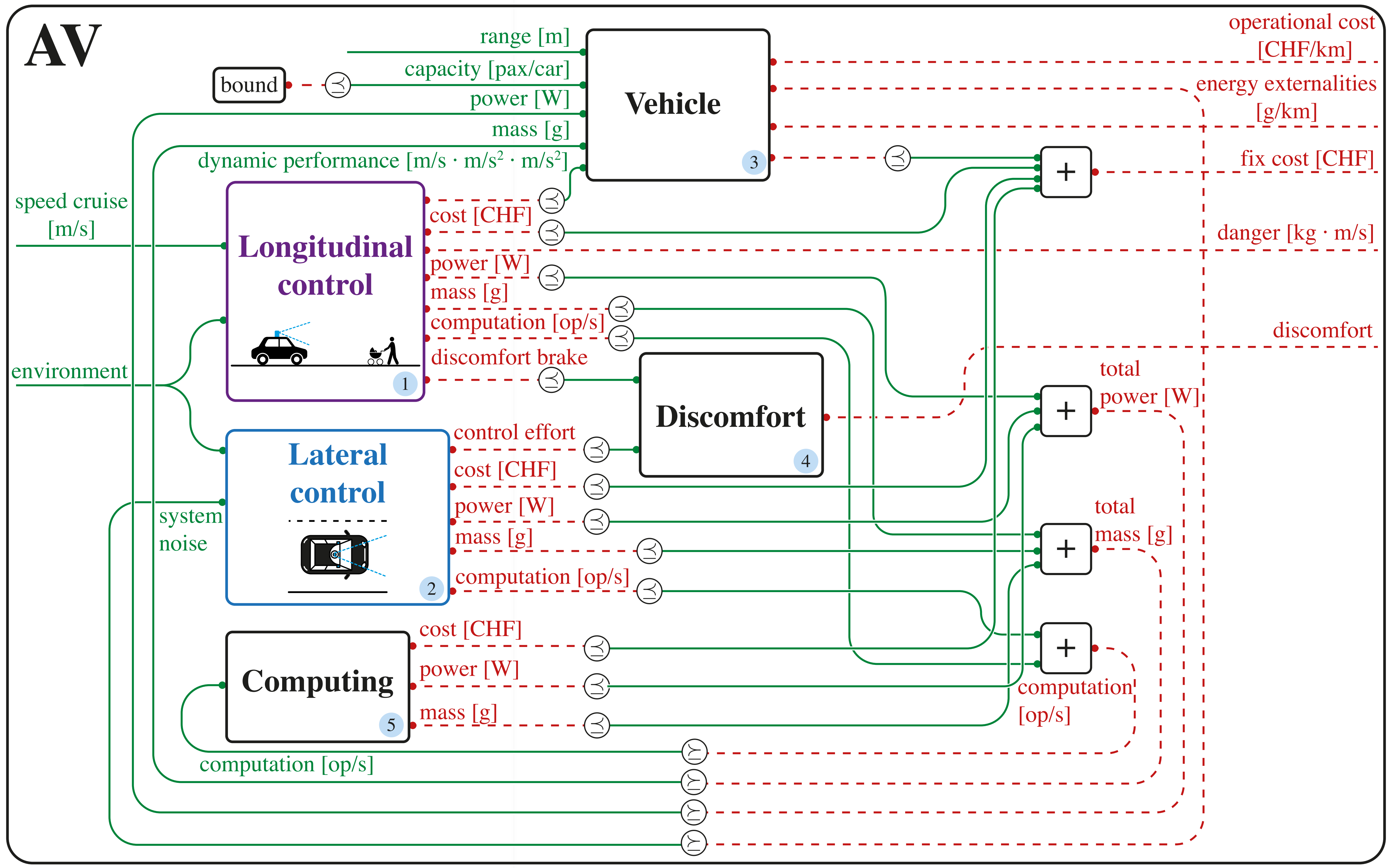


Every five years



Take-aways

- A new approach to **collaborative**, **computational**, **compositional**, **continuous** design.
  - Designed to work **across fields** and **across scales**.
  - **Compositional** horizontally and hierarchically.
  - Supports both **data-driven** and **model-based** components.
  - **Computationally tractable**.
  - **Intellectually tractable.**



- **If you are interested in using applied category theory:**
  - https://applied-compositional-thinking.engineering
  - New online classes series announcement soon!

Posetal Games

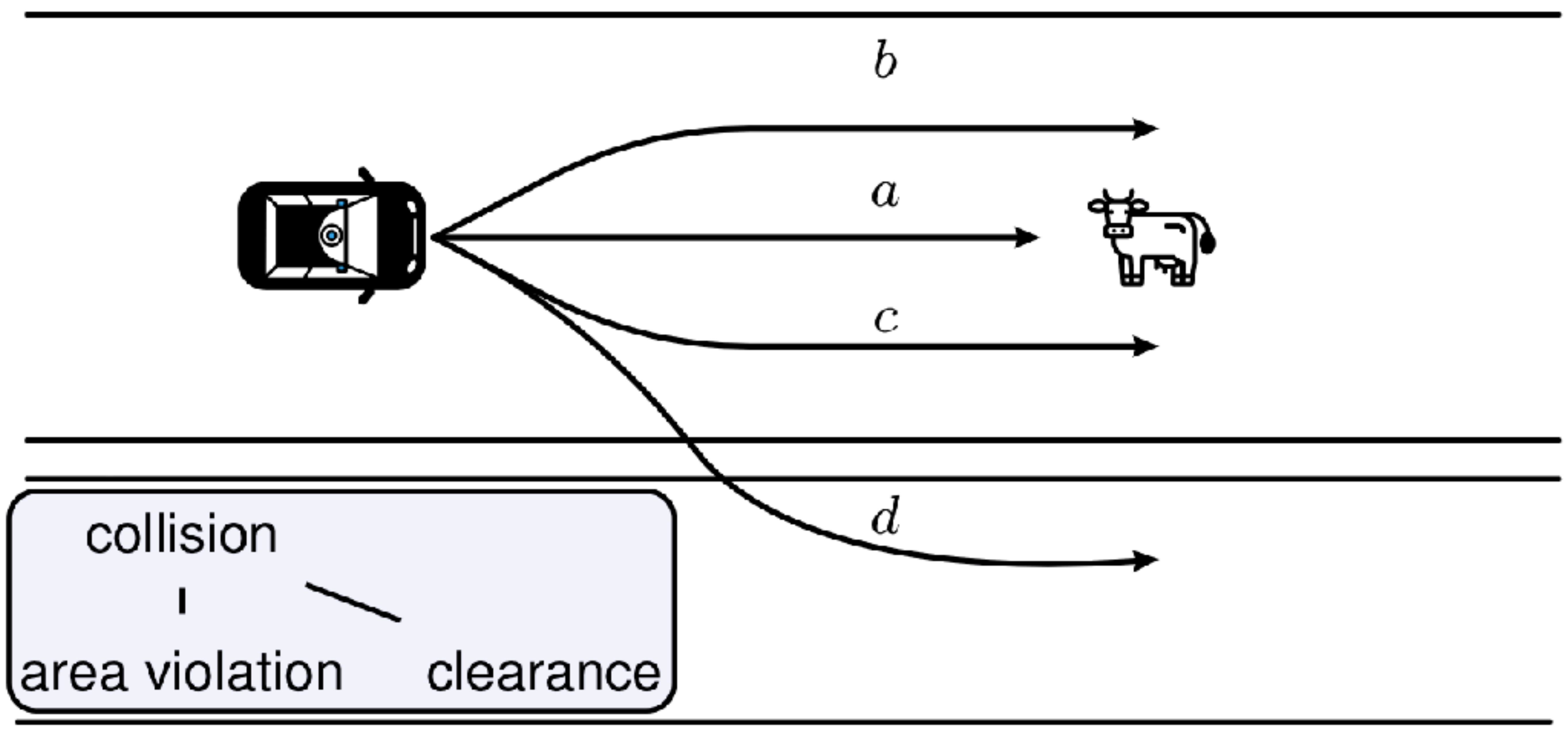
- We present **Posetal games.** In short:
  - Each **player** expresses a *partially ordered* **preference** over a set of metrics (scores)
  - Based on **preferences**, players select an **action** from a decision space
  - Given the joint **action profile** of players, we obtain a **game outcome** for each player via a *deterministic* **metric function**

- Preferences over metrics **induce** preferences over the **decision space:**

b and *c* are **indifferent**

b, c, d are **preferred** over *a*

b, c are **incomparable** with *d*



RA-L 2022: https://bit.ly/3cPsW9Y

Current and future collaborators

- **Collaborators** for the presented works

Nicolas Lanzetti

Dejan Milojevic

Alessandro Zanardi

Saverio Bolognani

Andrea Censi

Emilio Frazzoli

Florian Dörfler

Marco Pavone

ETH zürich

Swiss National Science Foundation

NCCR Automation

References

- **Papers I talked about and dedicated talks available at** **https://gioele.science**

- [1] **Zardini, G.**, Lanzetti, N., Guerrini, L., Frazzoli, E., & Dörfler, F. (2021, September). Game theory to study interactions between mobility stakeholders. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (pp. 2054-2061). (***Best Paper Award***).
- [2] Zanardi, A., **Zardini, G.**, Srinivasan, S., Bolognani, S., Censi, A., Dörfler, F., & Frazzoli, E. (2021). Posetal games: Efficiency, existence, and refinement of equilibria in games with prioritized metrics. *IEEE Robotics and Automation Letters*, *7*(2), 1292-1299.
- [3] **Zardini, G.**, Lanzetti, N., Pavone, M., & Frazzoli, E. (2022). Analysis and control of autonomous mobility-on-demand systems. *Annual Review of Control, Robotics, and Autonomous Systems*, *5*, 633-658.
- [4] **Zardini, G.**, Lanzetti, N., Salazar, M., Censi, A., Frazzoli, E., & Pavone, M. (2020, September). On the co-design of AV-enabled mobility systems. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1-8).
- [5] **Zardini, G.**, Lanzetti, N., Censi, A., Frazzoli, E., & Pavone, M. (2020). Co-design to enable user-friendly tools to assess the impact of future mobility solutions. *arXiv preprint arXiv:2008.08975*.
- [6] **Zardini, G.**, Censi, A., & Frazzoli, E. (2021, June). Co-design of autonomous systems: From hardware selection to control synthesis. In *2021 European Control Conference (ECC)* (pp. 682-689).
- [7] **Zardini, G.**, Milojevic, D., Censi, A., & Frazzoli, E. (2021, January). Co-design of embodied intelligence: A structured approach. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 7536-7543).
- [8] **Zardini, G.**, Suter, Z., Censi, A., & Frazzoli, E. (2022). Task-driven Modular Co-design of Vehicle Control Systems. *arXiv preprint arXiv:2203.16640*.
- [9] Censi, A., Lorand, J., & **Zardini, G.** Applied Compositional Thinking for Engineers, 2022, work in progress book.