DeepSLOs for the Computing Continuum

Víctor Casamayor Pujol, Boris Sedlak, Yanwei Xu, Praveen Kumar Donta, Schahram Dustdar

Distributed Systems Group - TU Wien

ApPLIED Workshop - Nantes - 17 June 2024

The Computing Continuum

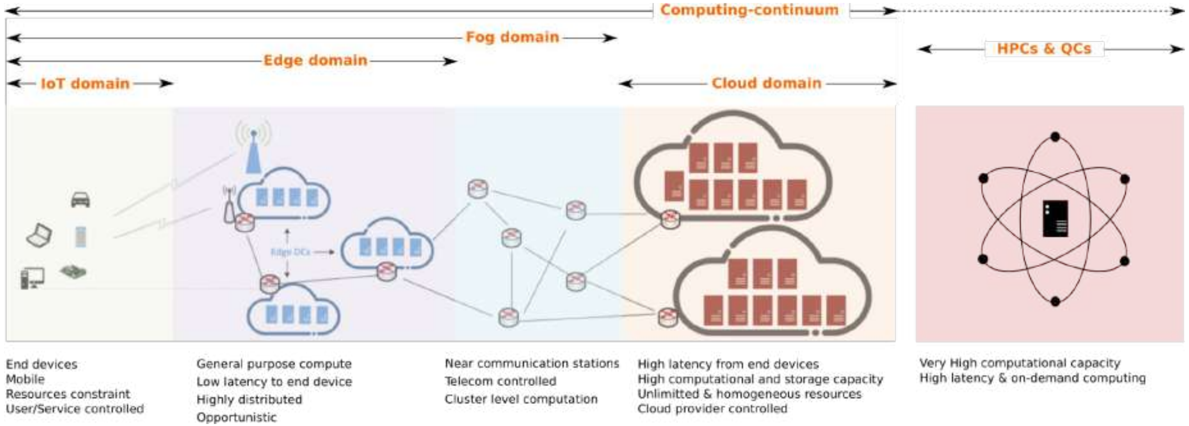
Computing fabric composed of all current computational tiers.

A seamless integration of the computing infrastructure.

Leverages the best of each tier.

Expected applications:

- ➔ eHealth
- ➔ Autonomous vehicles
- ➔ Smart cities
- ➔ Resources management


Computing-continuum
Fog domain
Edge domain
IoT domain
Cloud domain
HPCs & QCs
Edge DCs
End devices
Mobile
Resources constraint
User/Service controlled
General purpose compute
Low latency to end device
Highly distributed
Opportunistic
Near communication stations
Telecom controlled
Cluster level computation
High latency from end devices
High computational and storage capacity
Unlimitted & homogeneous resources
Cloud provider controlled
Very High computational capacity
High latency & on-demand computing

We have a centralized and limited visibility over the system performance

The Computing Continuum

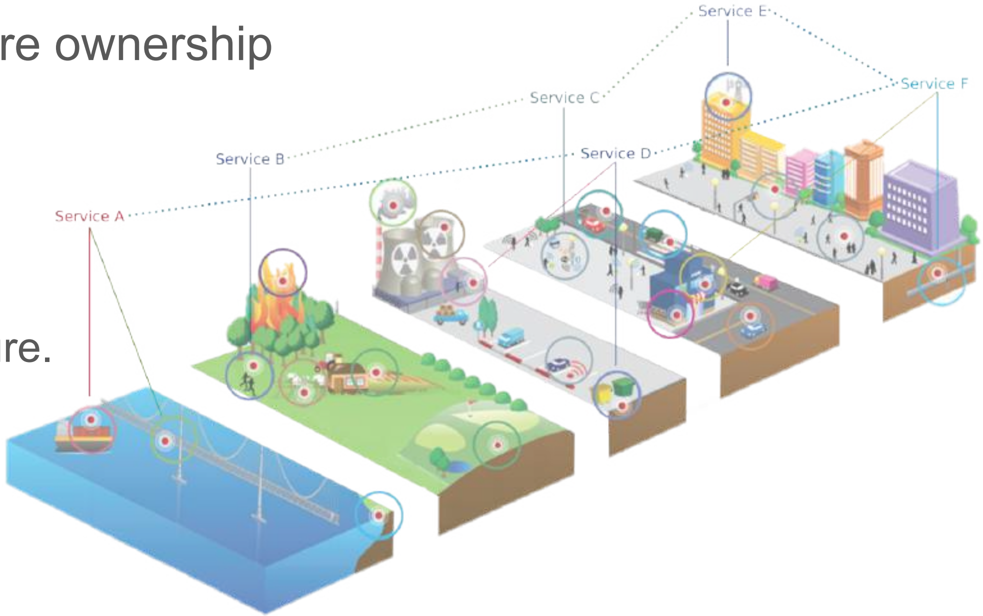
Multi-proprietary: Shared infrastructure ownership

System *issues* propagate

Each stakeholder has:

➔ Own global interest

➔ Local requirements of its infrastructure.



We need tools to understand the relationship between each SLO (requirement) and how propagation unfolds.

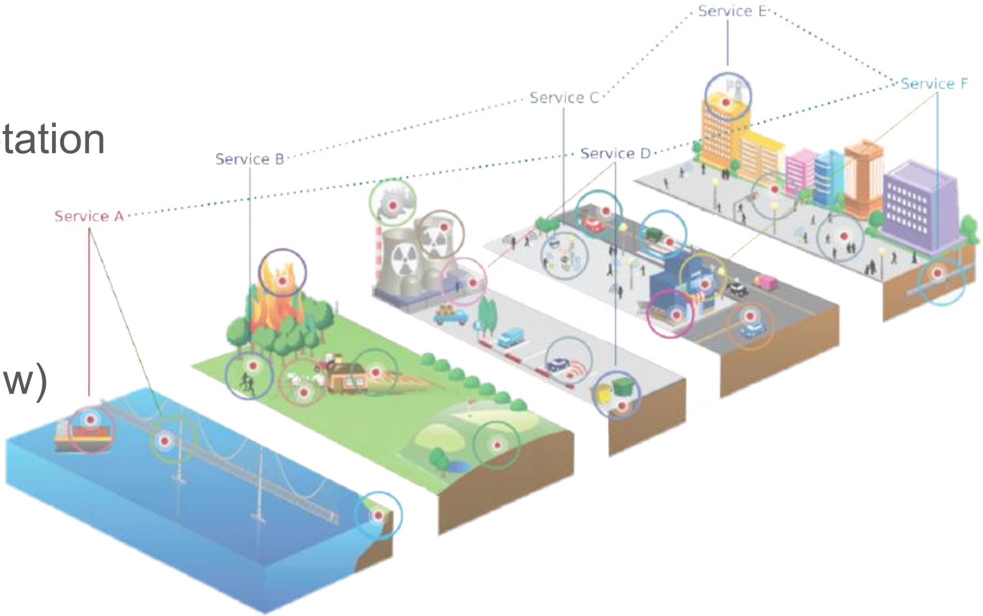
The Computing Continuum

Geographically distributed

Challenges deployment and service adaptation

Centralized governance falls short (intensified by stricter requirements)

Tailored runtime adaptations (Service + Hw)



We need decentralized governance, which considers local characteristics of the service and the host.

Motivation

We need a global view on the SLOs (requirements) but with a decentralized (local) governance capability

We propose:

- ➔ Distribute SLOs with services as the intelligent entity through the application (All services get specified —> Leading to application accountability)
- ➔ The DeepSLO to structure all SLOs.

Think global, act local

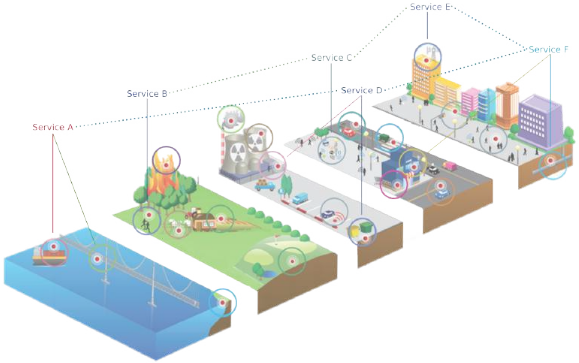
Service Level Objective as system requirements

We propose to specify (with SLOs) all application services with their hosts.

Reduce the combinatorial space for both services and hosts: clustering [2] and sampling [1]

We define an SLO:

$$SLO = P(x \leq sli \leq y) \mid x \leq y; \ \forall x, y \in SLI$$

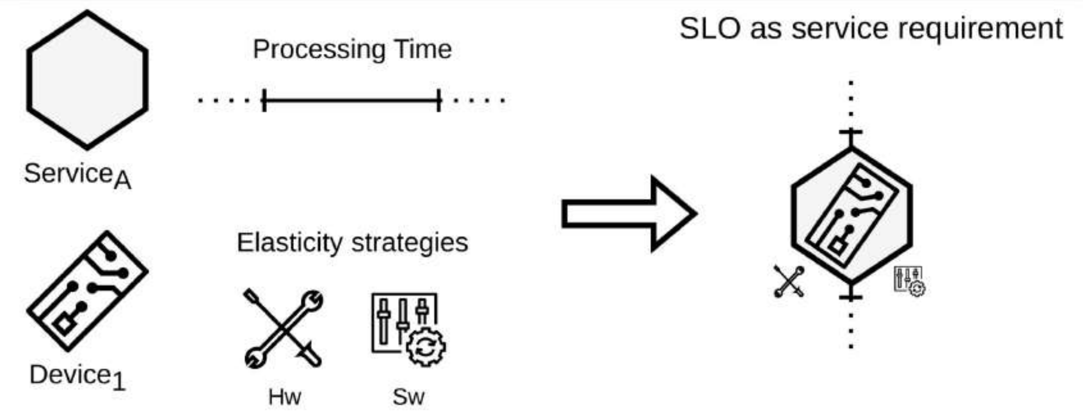


[1] V. C. Pujol, A. Morichetta, and S. Nastic, “Intelligent Sampling: A Novel Approach to Optimize Workload Scheduling in Large-Scale Heterogeneous Computing Continuum,” in *2023 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, Jul. 2023, pp. 140–149. doi: 10.1109/SOSE58276.2023.00024.

[2] A. Morichetta *et al.*, “PolarisProfiler: A Novel Metadata-Based Profiling Approach for Optimizing Resource Management in the Edge-Cloud Continnum,” in *2023 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, Jul. 2023, pp. 27–36. doi: 10.1109/SOSE58276.2023.00010.

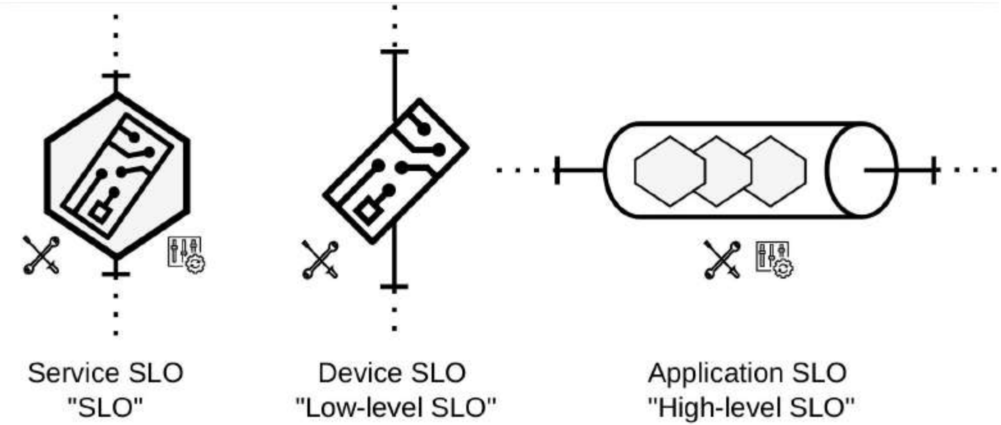
Service Level Objective as system requirements

Components: Service + host + SLO + elasticity strategies
Tailored adaptations to service and host



Service Level Objective as system requirements

Three types of SLOs



SLO area of interest

SLOs have to be **adaptive** & **explainable**.

This means data must be gathered and assessed locally.
It is crucial to select only relevant data for the SLO.

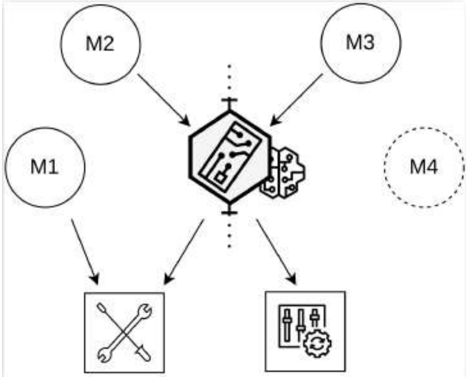
SLO area of interest

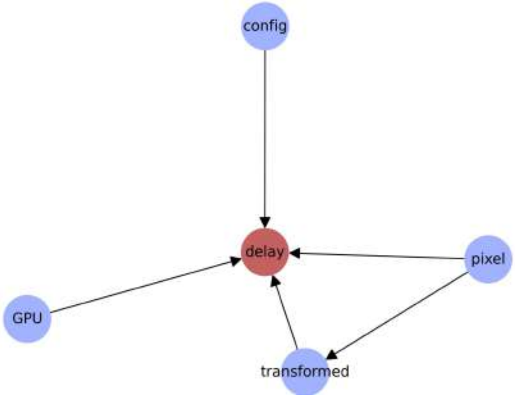
We use the Markov Blanket [1]

➔ Causal filter

➔ Model interfaces

$$P(x|MB(x), Y) = P(x|MB(x))$$





Markov Blanket for processing delay from [2]

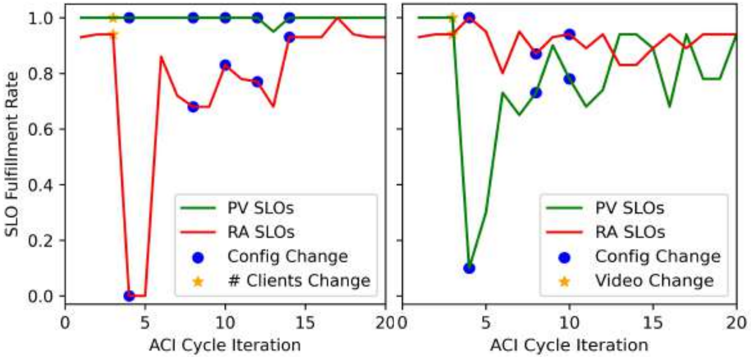
[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[2] B. Sedlak, V. C. Pujol, P. K. Donta, and S. Dustdar, “Designing Reconfigurable Intelligent Systems with Markov Blankets,” in *Service-Oriented Computing*, in Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2023, pp. 42–50. doi: 10.1007/978-3-031-48421-6_4.

SLO area of interest

Autonomy: Active Inference [2] agents from the Free Energy Principle [1]

➔ Can leverage Markov Blanket models.
➔ Embed exploitation-exploration trade-off.
➔ No reward function, but preferred observations.
➔ Capacity to improve model to causal graph.



SLO fulfillment of 2 services controlled by AIF agents [3]

[1] K. Friston *et al.*, “The free energy principle made simpler but not too simple,” Jan. 2022, doi: 10.48550/arxiv.2201.06387.

[2] R. Smith, K. J. Friston, and C. J. Whyte, “A step-by-step tutorial on active inference and its application to empirical data,” *Journal of Mathematical Psychology*, vol. 107, p. 102632, 2022, doi: https://doi.org/10.1016/j.jmp.2021.102632.

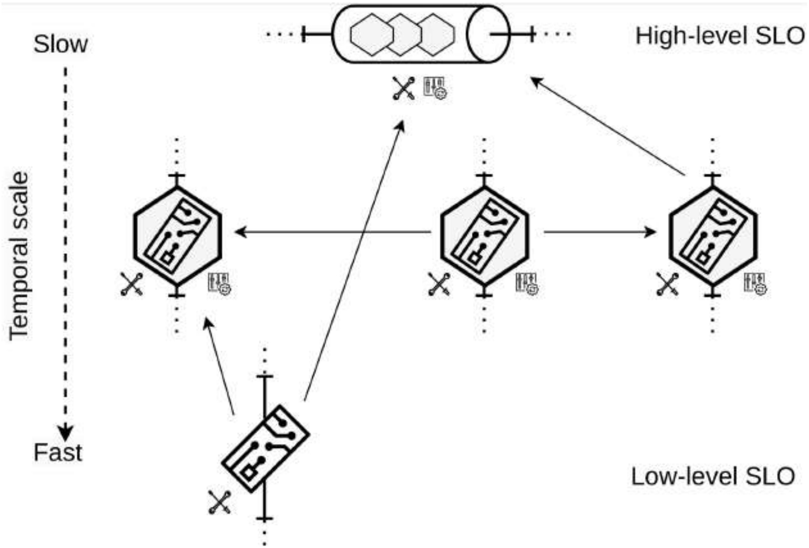
[3] B. Sedlak, V. C. Pujol, P. K. Donta, and S. Dustdar, “Equilibrium in the Computing Continuum through Active Inference,” *Future Generation Computer Systems*, May 2024, doi: 10.1016/j.future.2024.05.056.

DeepSLOs

Hierarchical structure connecting all SLOs

At design:
Address prioritization (trade-offs) between SLOs.

At runtime:
Diffuse higher-level policies towards lower-level SLOs to ensure a cohesive.
Detect conflicting SLOs



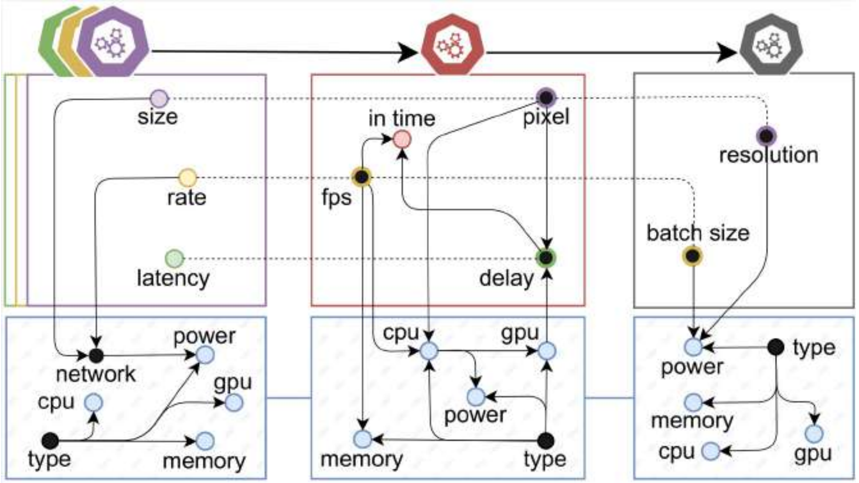
DeepSLOs

The links between the variables in different SLOs might com from different sources:

➔ Service dependencies

➔ Data analysis

➔ High-level SLO specifications

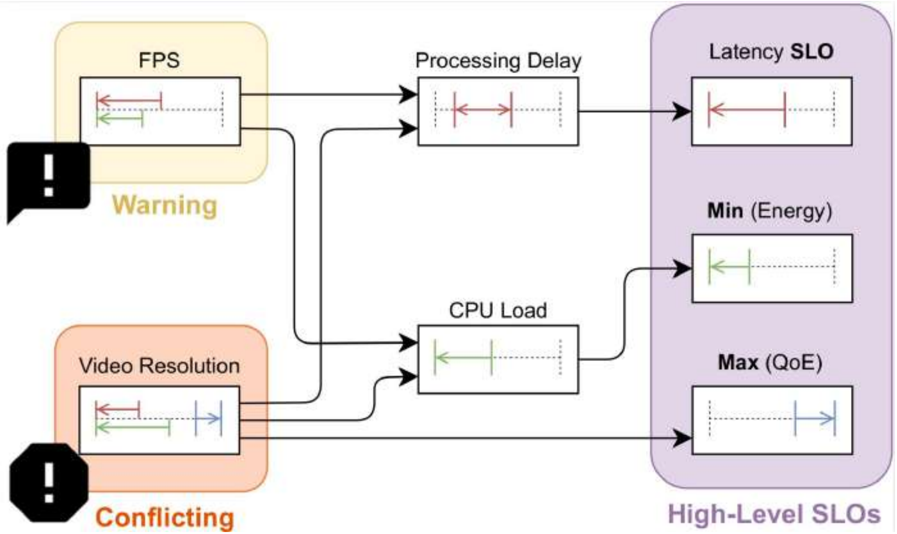


[1] B. Sedlak, V. C. Pujol, P. K. Donta, and S. Dustdar, “Markov Blanket Composition of SLOs,” IEEE International Conference on Edge Computing & Communications, July 2024, doi: TBD.

DeepSLOs

The DeepSLO as a BN can detect conflicting SLOs.

And if possible, resolve them.



[1] B. Sedlak, V. C. Pujol, P. K. Donta, and S. Dustdar, “Diffusing High-level SLO in Microservice Pipelines,” IEEE International Conference on Service-Oriented System Engineering, July 2024, doi: TBD.

Limitations & Future work

Data-centric approach.

Expert knowledge.

Communication overhead.

Design process involvement.

New ways to build DeepSLOs.

GNNs with LLMs to propose initial DeepSLO structures.

Larger and more intricate applications.

Conclusions

Multi-owned DCCS face the need of addressing individual requirements and the overall application goal.

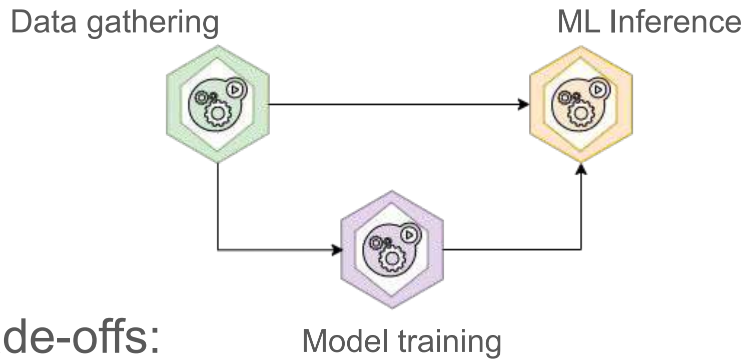
DeepSLOs allow a local and self-adapt behavior while accounting for general policies.

Further, the model brings explainable behaviors to entire applications.

Thank you for your attention

Any Questions?

Running example



A model for the entire eHealth application.

Stakeholders can address design and runtime trade-offs:
Take cost SLO and QoE (user satisfaction).
Hence, the rate of GPU usage can balance cost and QoE.

The relationship between certain SLOs require specific constraints:
GPU cost can vary depending on the provider.

Finally, prioritization and agreement between stakeholders can ease trade-offs, e.g., in the case of a eHealth, QoE might be always a priority.

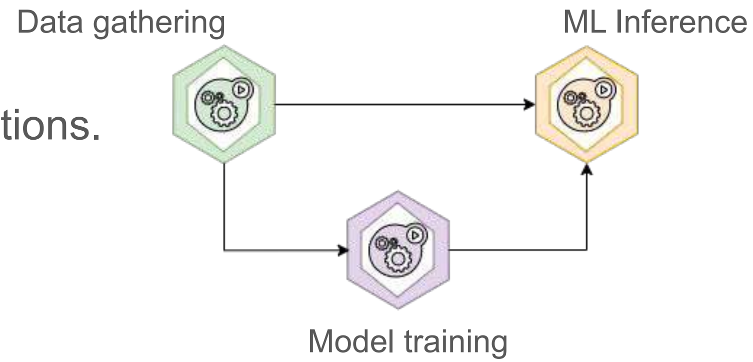
Running example

SLO: energy consumption of the SBC

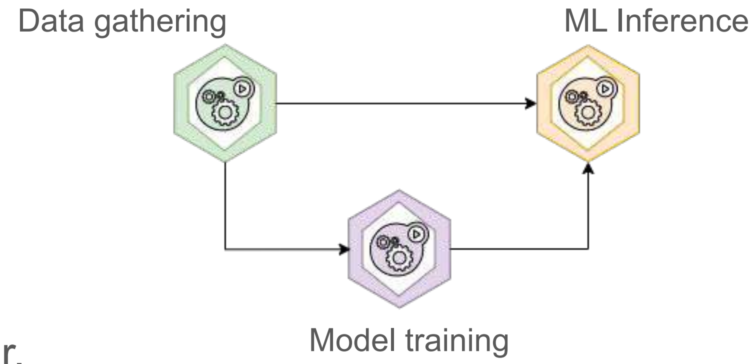
We need to track all relevant information related to the SLO.

Markov Blanket discovery techniques allow to select the relevant set out of all given data.

Active inference agents use the MB model.
Further, we can select the preferred observations
(i.e., SLO fulfillment) as the expected utility for the actions.
Finally, interventions can improve model accuracy.



Running example



Defining SLOs:

Inference processing time: T_{min} T_{max}

Host, an SBC, power consumption: less 8W per hour.

Regional policy requires privacy enhancements, shifting the processing time to T'_{min} T'_{max}

Types of SLOs:

A low-level SLOs: power consumption less than 8W per hour.

A high-level SLO: total system's running cost or the success rate of the alarm system.

Elastic strategies (beyond scaling):

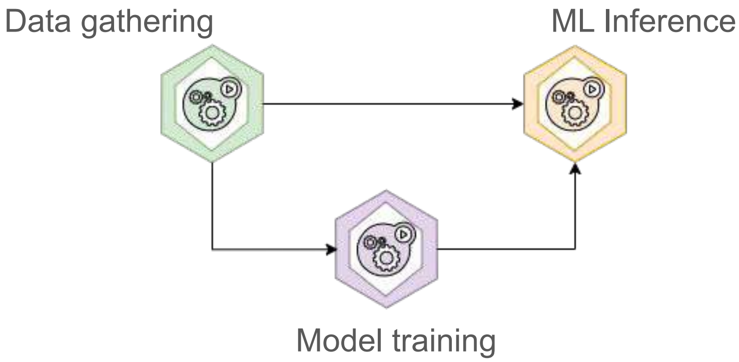
Switch ON/OFF the GPU at the Edge server for the training (Energy vs Cost vs Performance)

Offload training to Cloud (requiring further privacy enhancement (Privacy vs Cost)

Running example

eHealth application that remotely gather data from the patient and alerts (by means of ML inference) the health care services if attention is needed.

We take a piece of such application: a service gathering data, another training the model, and a service inferring the patient health condition.


Data gathering
ML Inference
Model training