Active Inference for Distributed Intelligence in the Computing Continuum

Schahram Dustdar

dustdar.prof

Thanks to my collaborators: Victor Casamayor-Pujol, Alireza Furutanpey, Boris Sedlak, Praveen Donta

The Computing Continuum

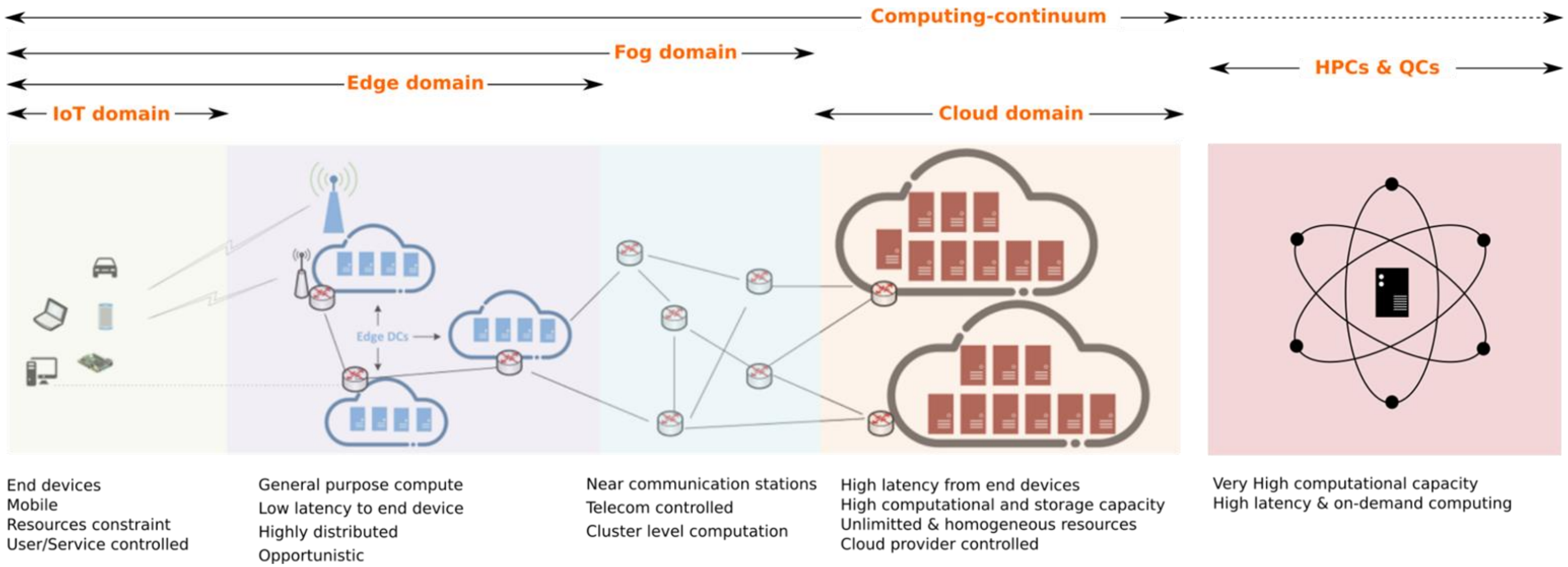
Computing fabric composed of all current computational tiers.

A seamless integration of the computing infrastructure.

Leverages the best of each tier.

Expected applications:

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



Today we have a centralized and limited visibility over the system performance, quality of service (QoS), and Quality of data.

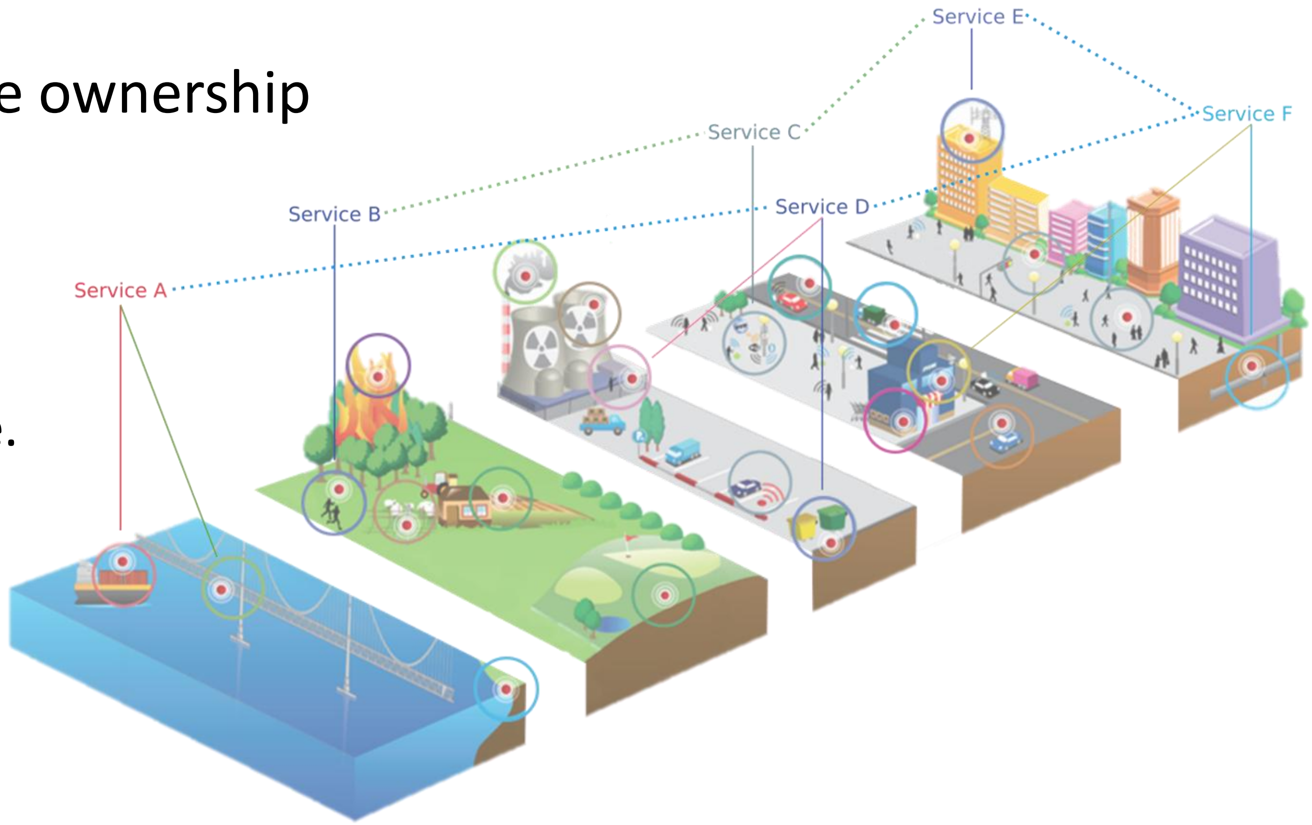
The Computing Continuum

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.

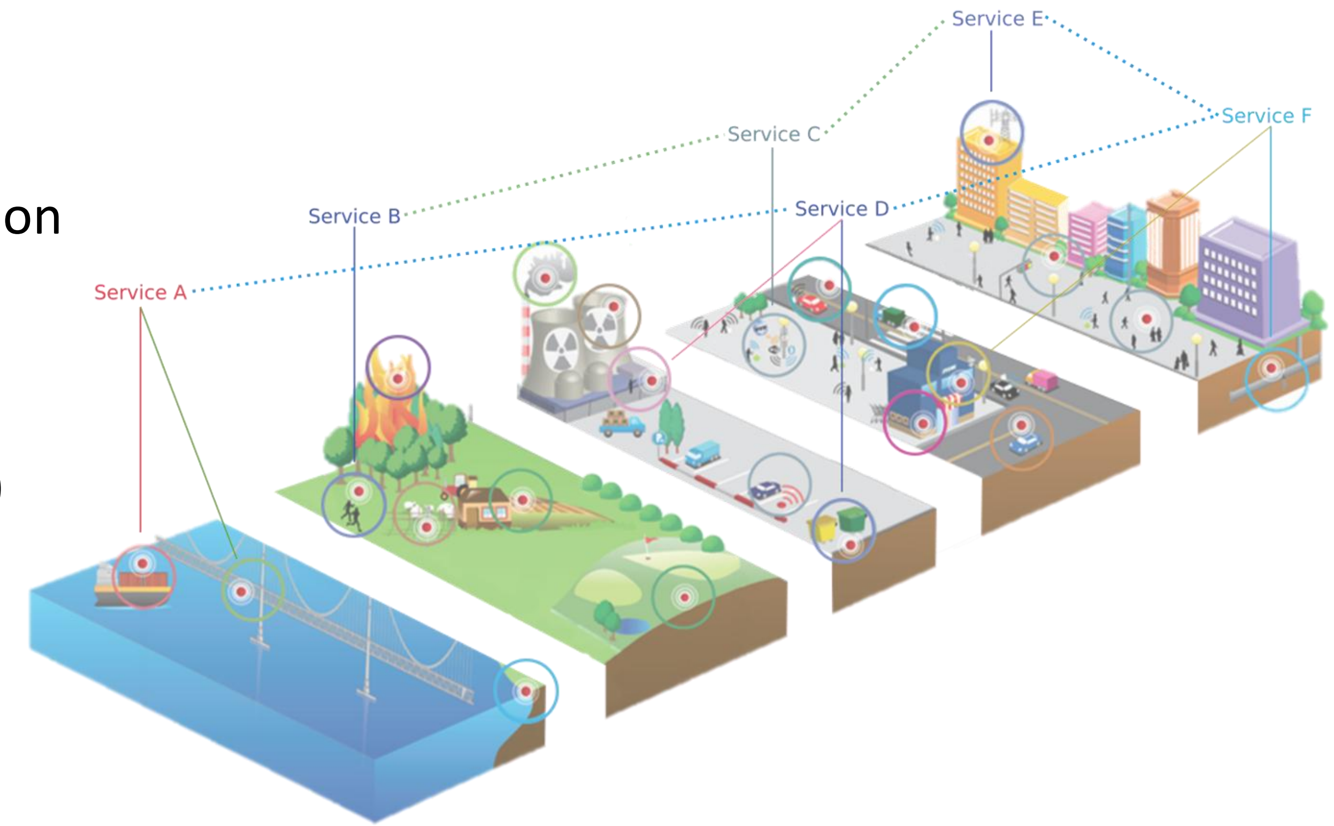
The Computing Continuum

Geographically distributed

Challenges deployment and service adaptation

Centralized governance falls short
(intensified by stricter requirements)

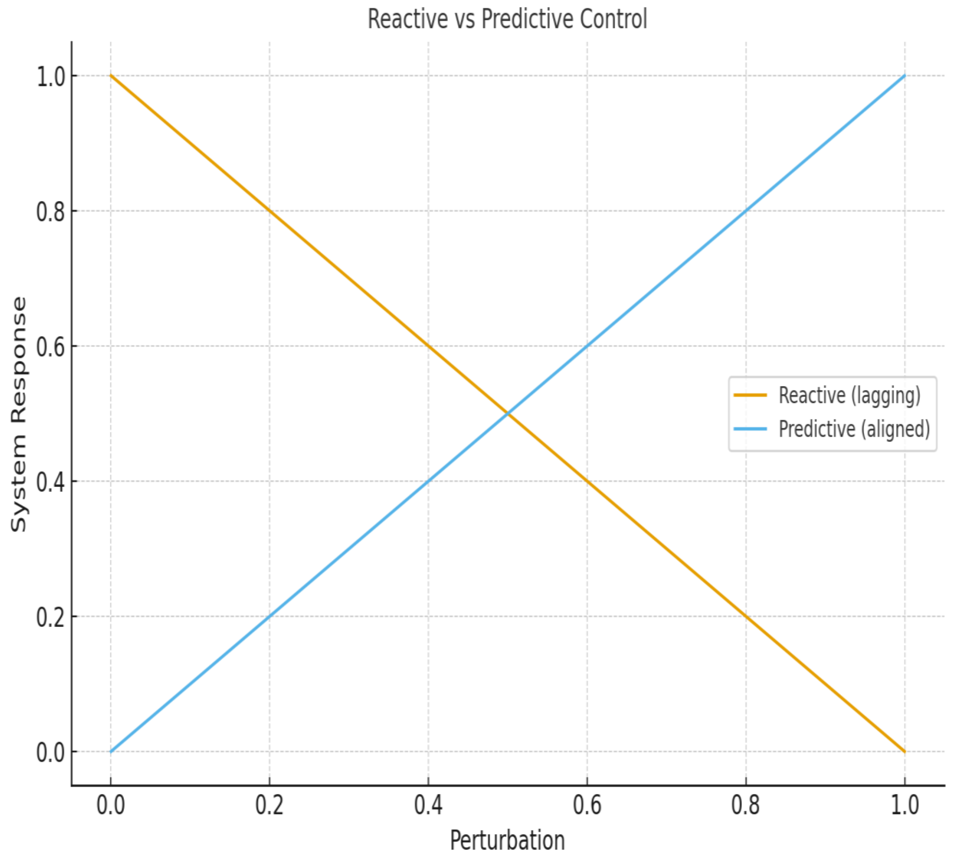
Tailored runtime adaptations (Service + HW)



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

Governance (Intelligence) in Continuum Computing

- Distributed apps span sensors, edge, fog, and cloud.
- Reactive, centralized management often fails.
- Non-stationarity breaks SLO compliance.
- Need predictive regulation over reactive.

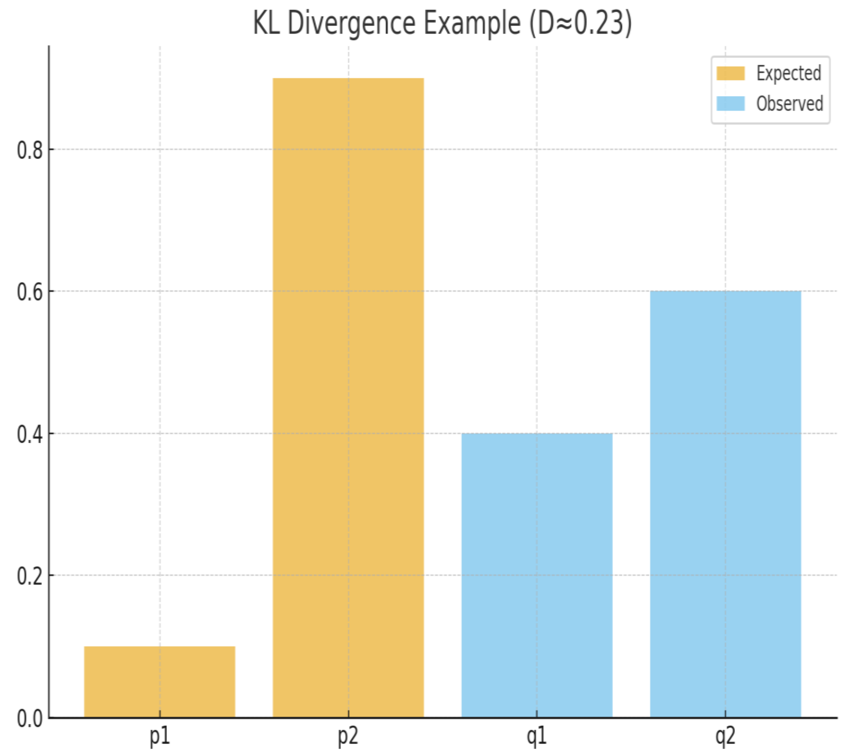


Biological Lens: Predictive Regulation

- Organisms regulate by anticipating changes.
- Homeostasis = reactive; Allostasis = predictive.
- Free Energy Principle: minimize prediction error.
- Analogy: components should anticipate loads.

Physics Lens: Fluctuation–Dissipation Theory (FDT)

- FDT links fluctuations to responses.
- Departures signal nonequilibrium dynamics.
- In computing: compare expected vs observed responses.
- Alignment = predictive equilibrium.



Predictive Equilibrium Defined

- Alignment between predicted and observed outcomes.
- Active property: sustained by adaptation.
- Combines dynamic balance, reconfiguration, predictive consistency.
- Basis for antifragility.

Mathematical Foundations: Bayes’ Theorem

Bayes’ rule updates beliefs given new observations.

- Prior p(s): belief before seeing data.
- Likelihood p(o|s): how observations arise from states.
- Evidence p(o): normalizing constant.
- Posterior p(s|o): updated belief after observation.

$$p(s|o) = \frac{p(o|s)\,p(s)}{p(o)}$$

Posterior = Likelihood × Prior / Evidence

Bayes' Theorem Example: Rain and Wet Grass

- Prior: P(Rain) = 0.3 (30% chance of rain).
- Likelihood: P(Wet|Rain) = 0.9 (grass usually wet if it rains).
- Alternative: P(Wet|¬Rain) = 0.1 (sprinkler can also make grass wet).
- Observation: Grass is wet → update belief about rain.
- Posterior: P(Rain|Wet) ≈ 0.64 (now more likely it rained).

$$P(Rain|Wet) = \frac{P(Wet|Rain)\,P(Rain)}{P(Wet)}$$

Posterior = Likelihood × Prior / Evidence

Variational Free Energy (VFE)

- Measures how well an approximate posterior Q(s) matches the true posterior.
- Minimizing F ≈ maximizing model evidence (probability of data under model).
- Balances Accuracy (fit to data) and Complexity (change in beliefs).
- Tractable alternative to exact Bayesian inference.

$$F = D_{KL}[Q(s)|P(s|o)] - \ln P(o)$$

Expected Free Energy (EFE)

Extends free energy to future outcomes.

- Evaluates policies π: sequences of actions.
- Captures both pragmatic value (rewards) and epistemic value (information gain).
- Provides solution to explore–exploit dilemma.

$$G(\pi) = E_Q[\ln Q(s|\pi) - \ln P(o, s|\pi)]$$

Kullback–Leibler (KL) Divergence

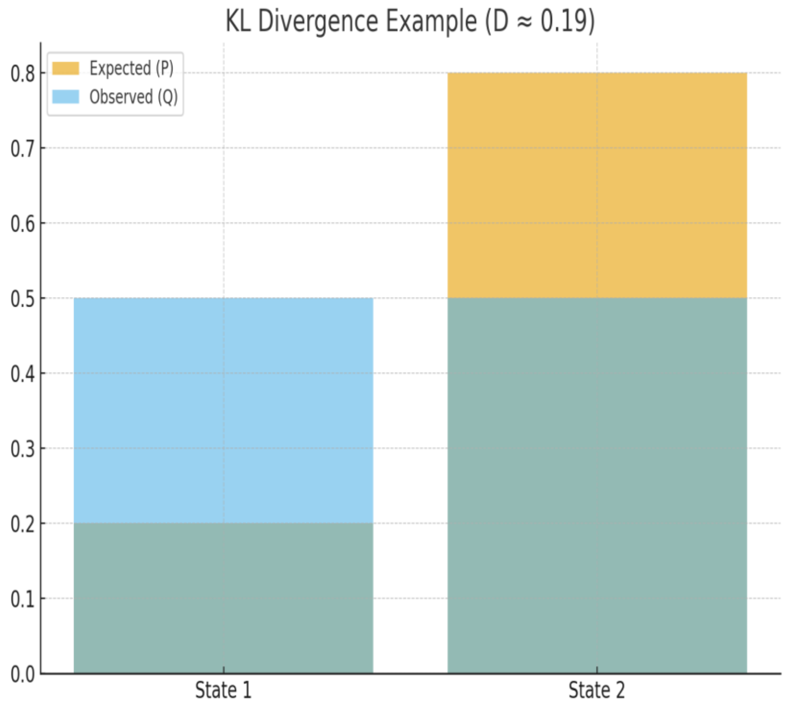
Measures dissimilarity between two probability distributions.

- D=0 when Q (posterior) and P (prior) match perfectly; larger values mean greater divergence.
- In Active Inference: diagnostic of model–world mismatch.
- Used as both early warning and learning signal.

$$D_{KL}[Q(x)|P(x)] = \sum_{X} Q(x) \ln \frac{Q(x)}{P(x)}$$

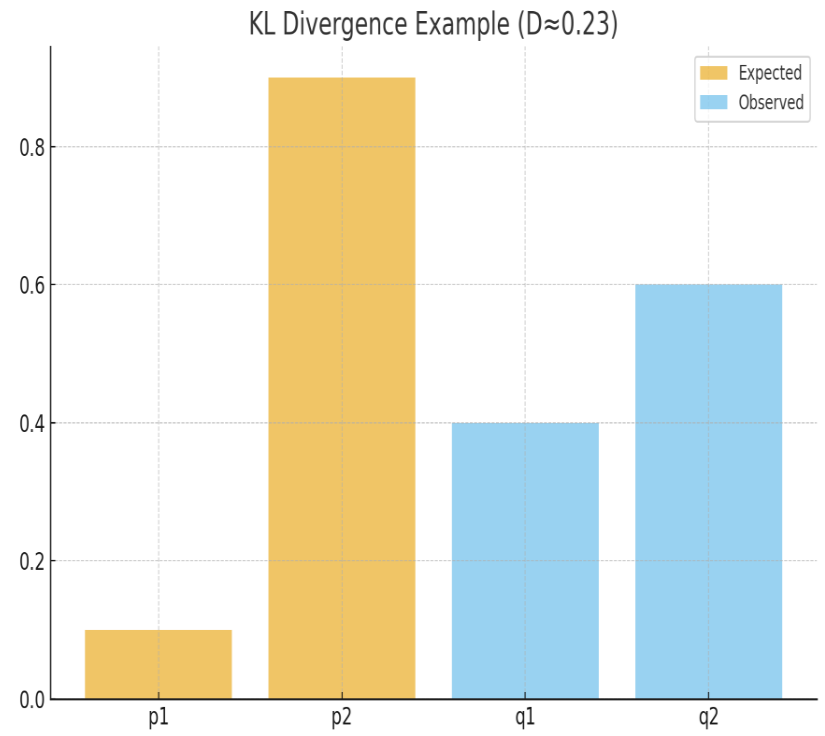
KL Divergence Example

- Suppose Expected distribution = [0.2, 0.8].
- Observed distribution = [0.5, 0.5].
- KL divergence quantifies the mismatch.
- Here, D ≈ 0.19, indicating nontrivial divergence.



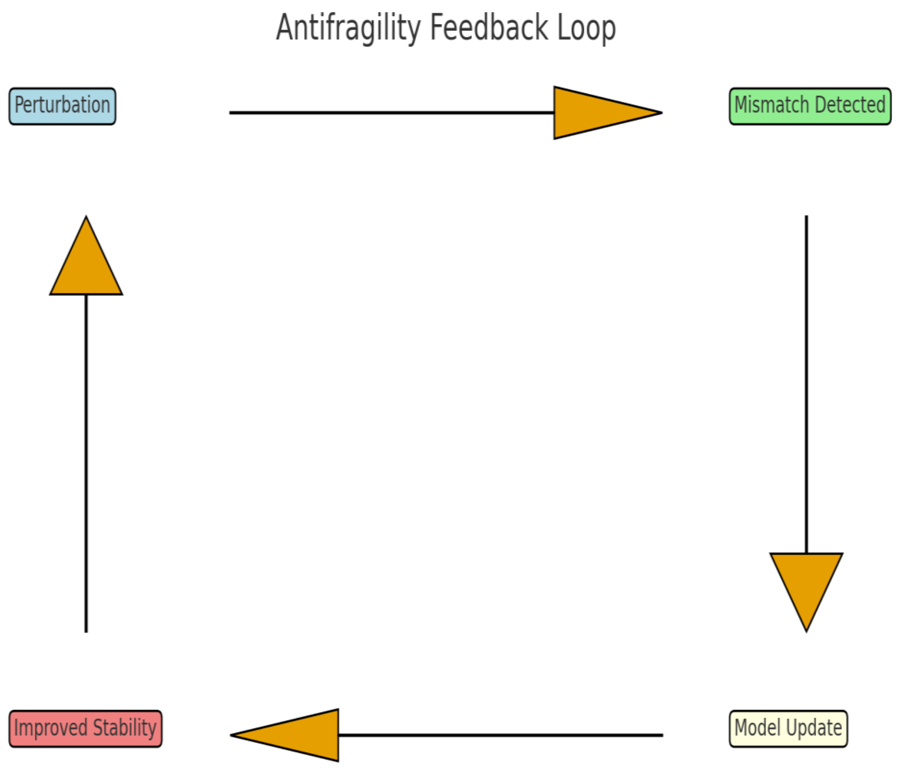
KL Divergence as Diagnostic Tool

- Distance between expected and observed behavior.
- Localizes failure causes: rewiring, coupling shifts, noise.
- Early-warning signal for model revision.
- Doubles as learning signal.



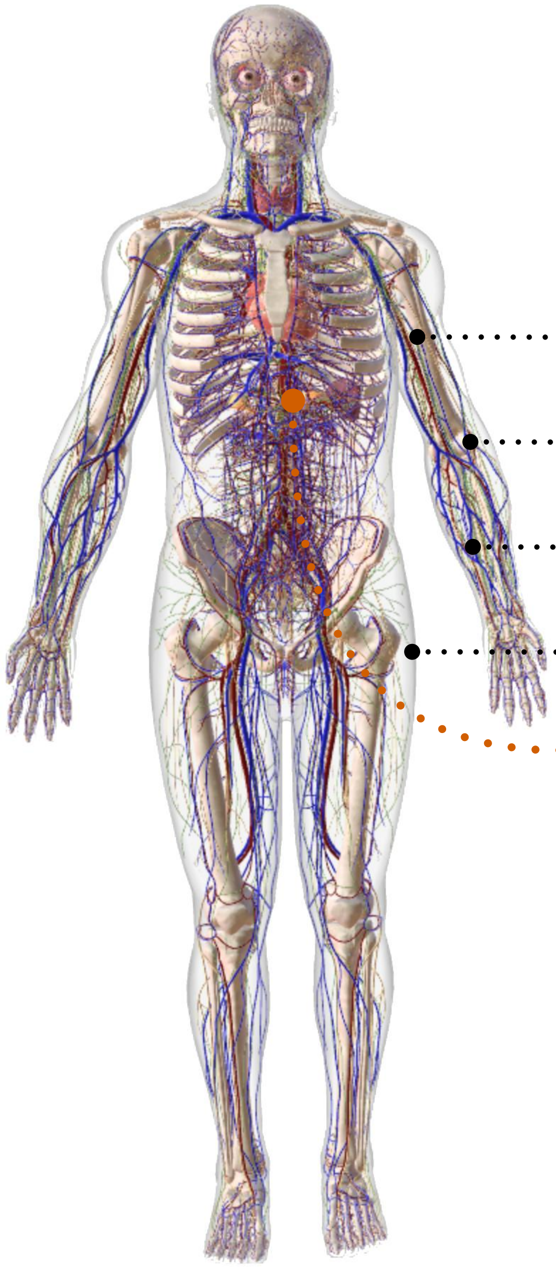
Antifragility in Distributed Systems

- Systems improve because of stress, not despite it.
- Perturbations expose mismatches → learning opportunity.
- BNs + KL provide learning gradient.
- Equilibrium supplies safety rails.



Architecture: Predictive Equilibrium in Action

- Local BNs at edge nodes predict responses.
- Fog nodes run perturbation campaigns and compute divergence.
- Cloud meta-controller aggregates signals.
- Balances exploration vs exploitation.



Human Ecosystem

The human body is comprised of a series of complex systems, including:

Skeletal System

Nervous System

Cardiovascular System

Lymphatic System

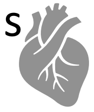
Endocrine System

Infrastructure Systems

Regulation Systems

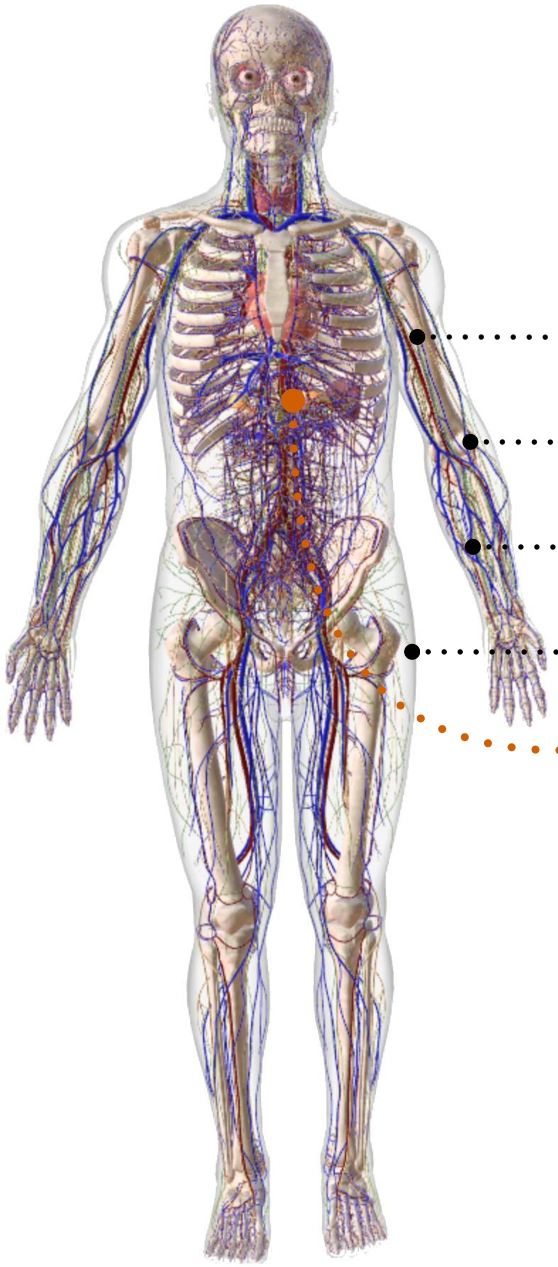
- Brain
- Spinal Cord
- Cranial Nerves
- Spinal Nerves

- Oxygen
- White Blood Cells
- Hormones
- Nutrients



Helping the body meet the demands (**40k neurons)**

Control Internal Environment, Memory and Learning (**86 billion neurons**)



Human Ecosystem

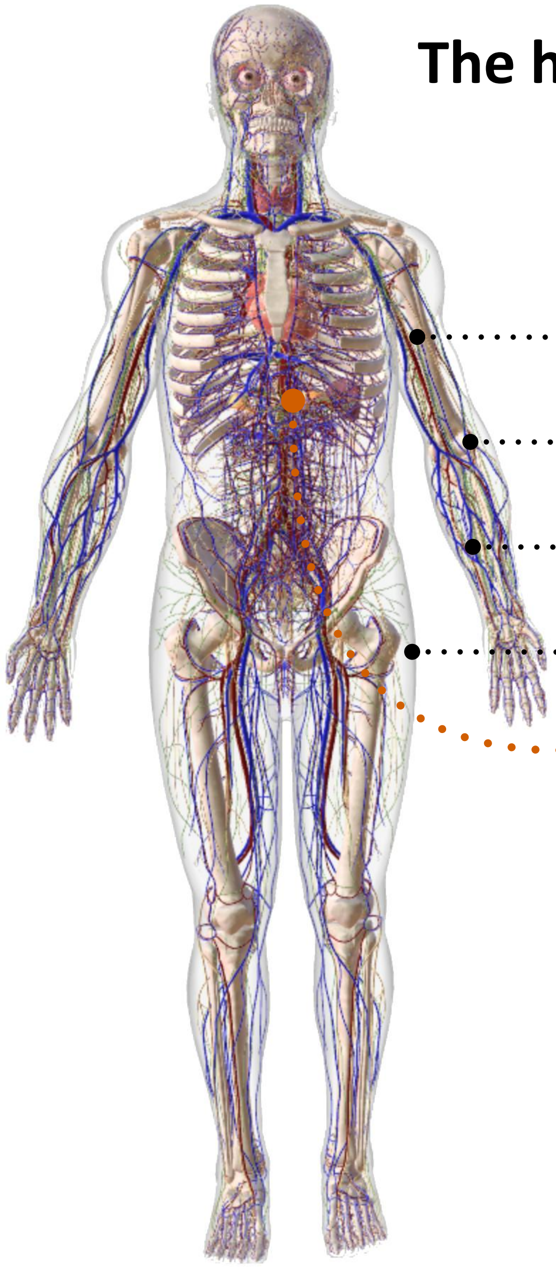
The human body is comprised of a series of complex systems, including:

- Skeletal System
- Nervous System
- Cardiovascular System
- Lymphatic System
- Endocrine System

→ Infrastructure Systems

→ Regulation Systems

The human body is comprised of a series of complex systems, including:

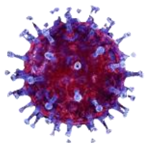


Human Ecosystem

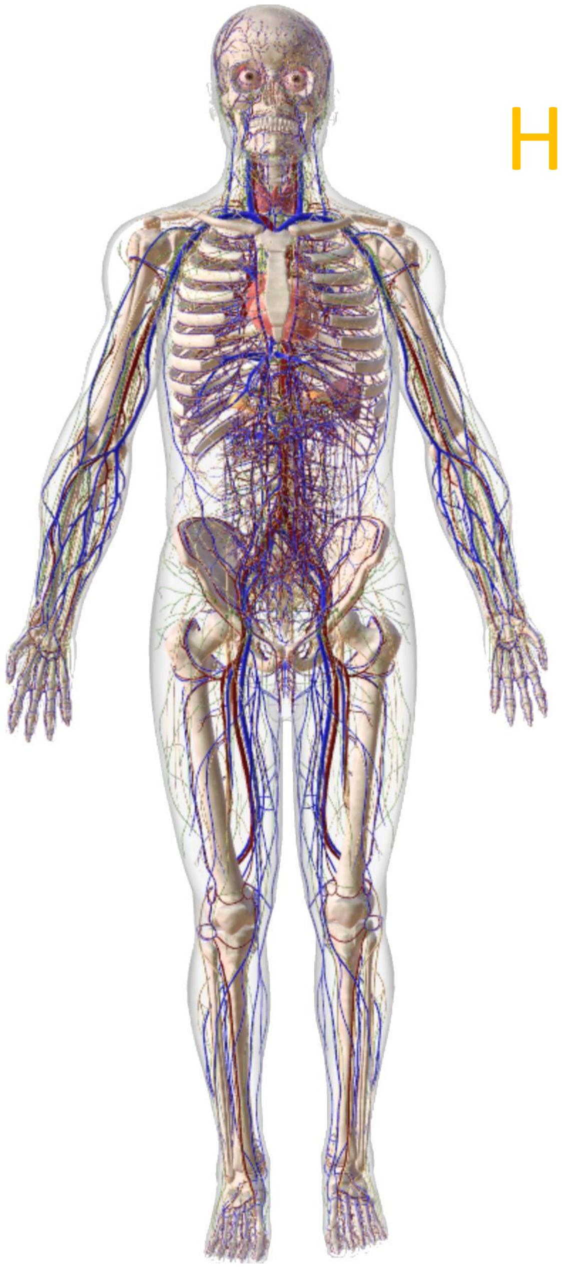
- Skeletal System
- Nervous System
- Cardiovascular System
- Lymphatic System
- Endocrine System

Infrastructure Systems → DeepSLOs, Collaborative Learning, Representation Learning

Regulation Systems → Zero Trust



- Part of the immune system
- Protects your body against foreign invaders
- Control and coordinate your body's metabolism
- Response to injury, stress, and mood



Human Ecosystem

Nervous system

Homeostasis and Resilience in DCCS

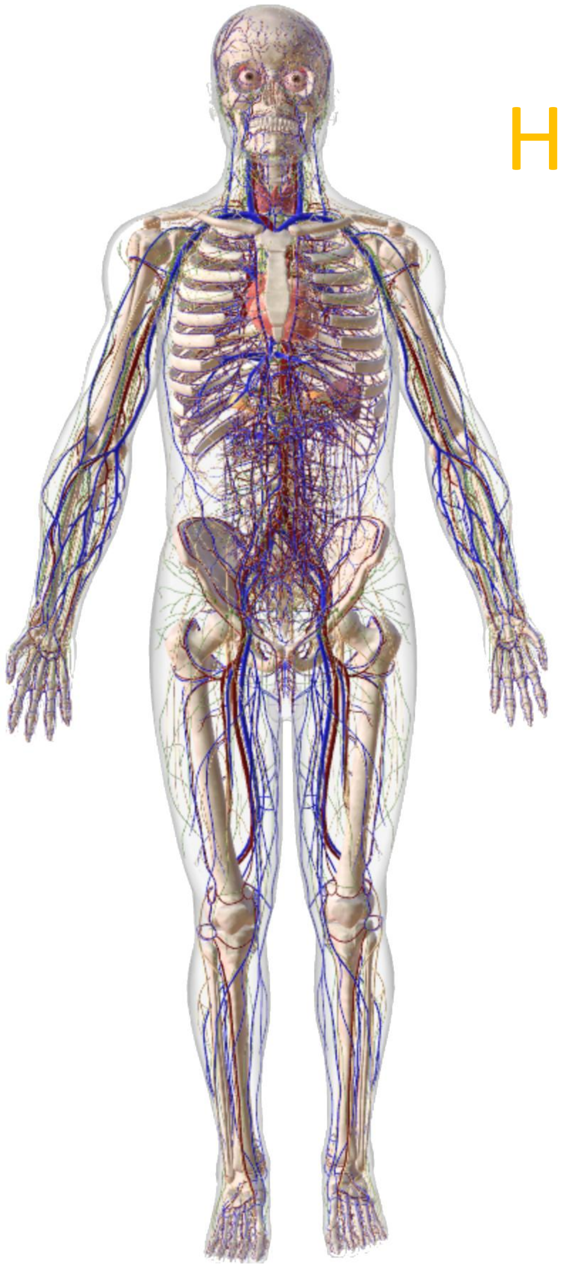
Human body self-regulates:

- Temperature
- Blood pressure
- …

Human body self-heals

Humans also learn how to maintain her/his needs satisfied.

Homeostasis and Resilience in DCCS



Human Ecosystem

Nervous system

Overall state - Top-bottom sensing.

From feeling *good-bad* to actual problem.

We also need this feature for DCCS due to their scale and interconnections.

Elasticity (Resilience)

(Physics) The property of returning to an initial form or state following deformation

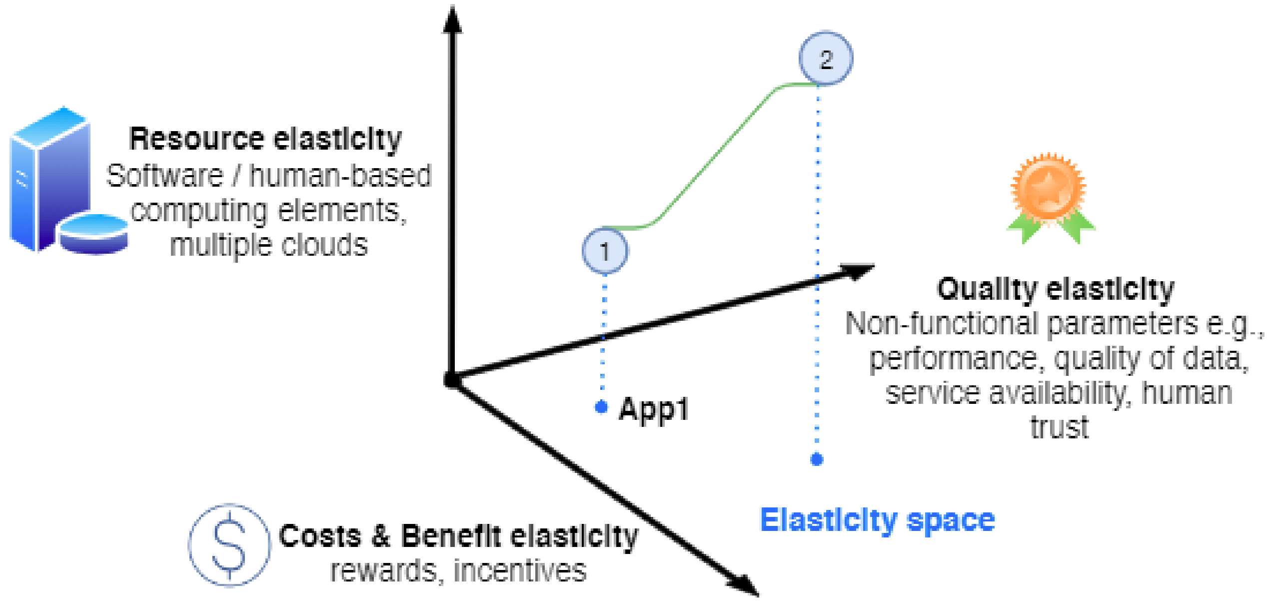
stretch when a force stresses them

e.g., ***acquire*** *new resources,* ***reduce*** *quality*

shrink when the stress is removed

e.g., ***release*** *resources,* ***increase*** *quality*

Elasticity > Scalability

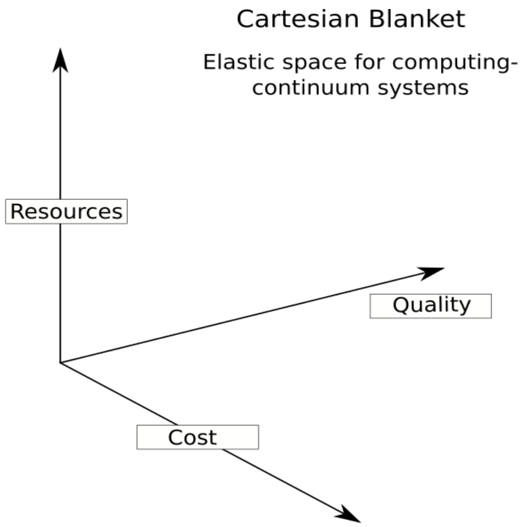


Dustdar S., Guo Y., Satzger B., Truong H. (2012) [Principles of Elastic Processes](), IEEE Internet Computing, Volume: 16, [Issue: 6](), Nov.-Dec. 2012

High-level state

Resources, Quality, Cost

- Highest-level description of system state from Cloud computing/elasticity work [1].
- DCCS have many different stakeholders with different interests, RQC can frame a common language.



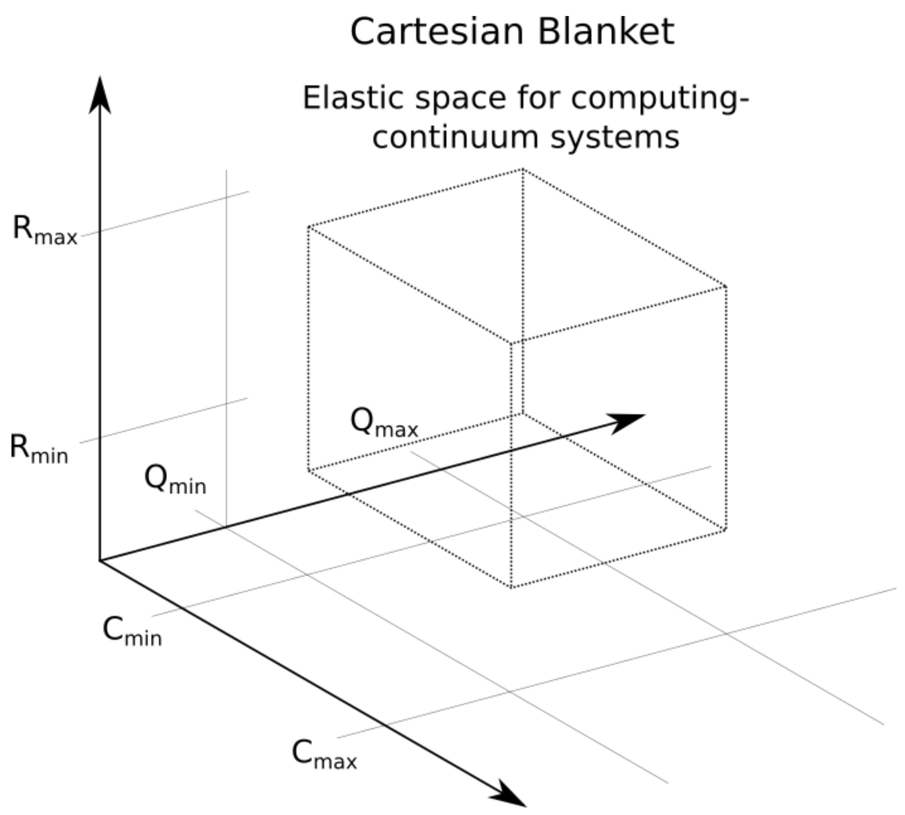
Operational equilibrium

- Defined as an operational mode of the application, from the highest level state.
- Any system can have several operational equilibria, leading to different configurations of the underlying infrastructure

[1] S. Dustdar, Y. Guo, B. Satzger, and H. L. Truong, "Principles of elastic processes," *IEEE Internet Computing*, vol. 15, no. 5, pp. 66–71, Sep. 2011, doi: 10.1109/MIC.2011.121.

The Cartesian Blanket
Adapting elasticity in the continuum

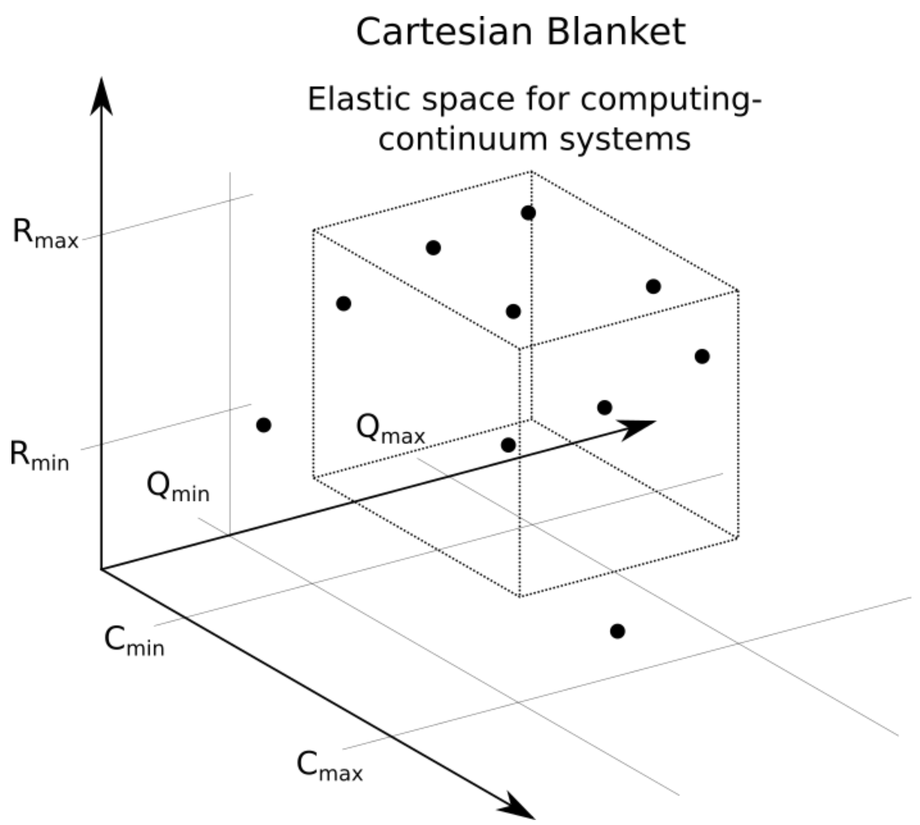
- System control based SLOs (Service Level Objectives)
- SLOs are represented as thresholds on the Cartesian space
- The system space is delimited within an hexahedron.
  - There is minimum and maximum value for each variable



The Cartesian Blanket

Adapting elasticity in the continuum

- The space is constraint to the actual infrastructure characteristics; not homogenous.
- The infrastructure is represented as points, not unlimited.
- The only valid infrastructure is the one **inside** the hexahedron.



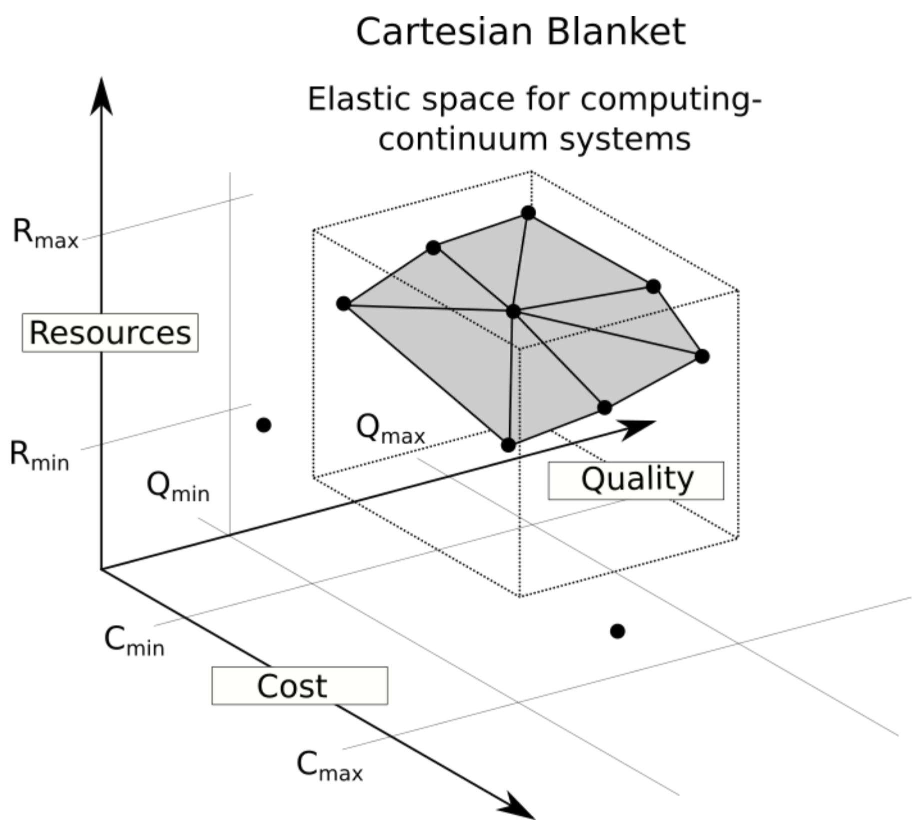
The Cartesian Blanket

Adapting elasticity in the continuum

- The system space possible configurations can be visualized as a stretched blanket over the infrastructure points.
  - Assuming linear interpolation on the space between the infrastructure components.

- Now we have the system represented, but

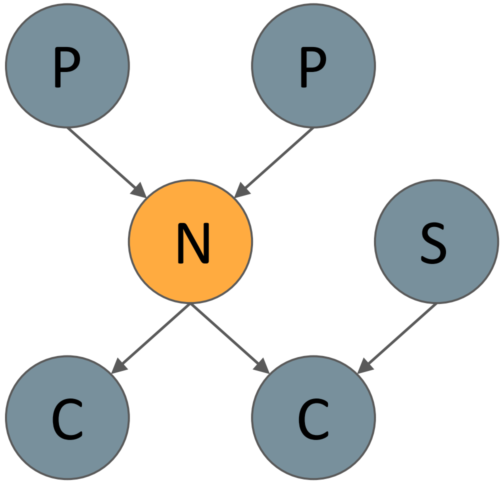
How can this representation help on the design and management of the distributed computing continuum systems?



Markov Blanket

The Markov Blanket of a random variable is the subset of nodes that provide enough information to statistically infer its value. Concept from Judea Pearl [1].

In a Bayesian Network, the Markov Blanket of a node (N) is composed of the parents (P), the children (C) and the co-parents of the children (S).



A tool for *causal* filtering.

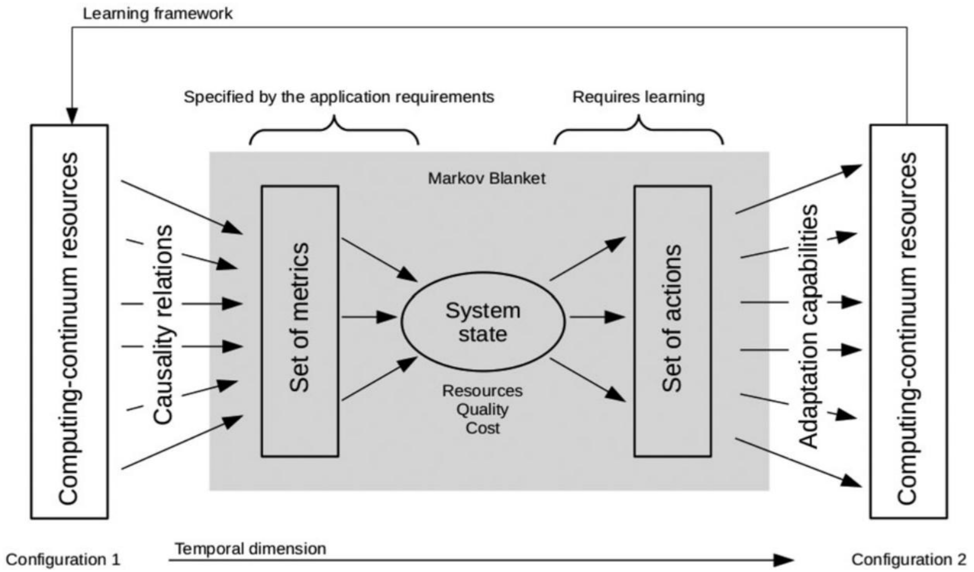
[1] Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc.

Markov Blanket (MB)

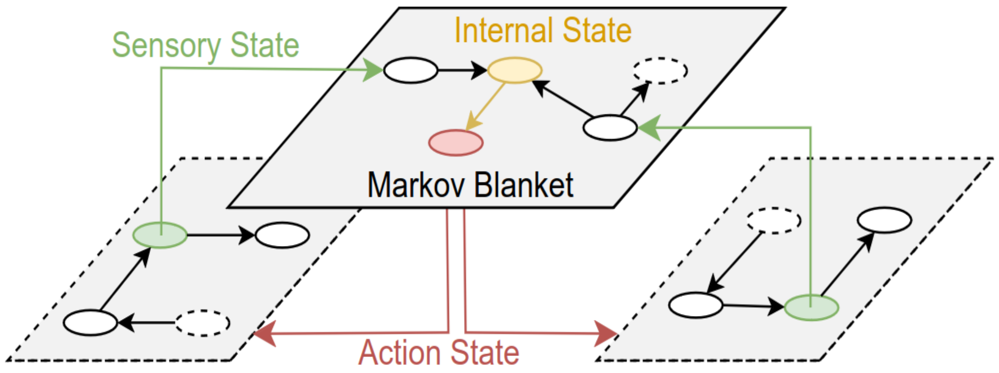
Interactions between **systems** (e.g., human in world) can be expressed through MBs – fulfill Markov property; allow modeling reactive behavioral models for elasticity

Creates **formal boundary** between a system and external states – limits scope of variables that determine **internal state**; discard remaining information to reduce dimension

Provides clear interfaces for **sensory**- and **action states**; policy (e.g., scaling) as a mapping between these states



Behavioral Markov blanket of a system [4]

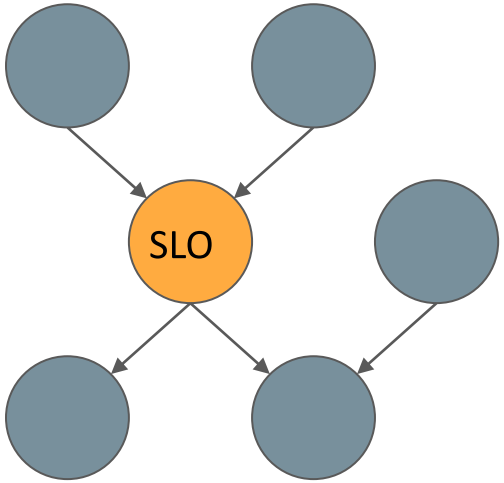


Action-perception cycle between multiple entities [5]

[4] Dustdar, Casamayor Pujol, and Dustdar; On Distributed Computing Continuum Systems (2023)
[5] Sedlak, Casamayor Pujol, Donta, and Dustdar; Markov Blanket Composition of SLOs (2024)

Causal Inference

- ➢ Discover & leverage causal relationships.
- ➢ 3 Rungs on the ladder of causation. [2]
  - Observational
  - Interventional
  - Counterfactual
- ➢ Explainability capacity



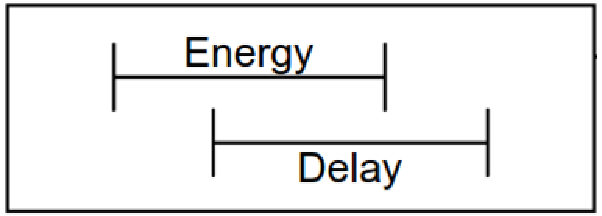
[2] J. Pearl and D. Mackenzie (2018), The Book of Why: The New Science of Cause and Effect. USA: Basic Books, Inc.,

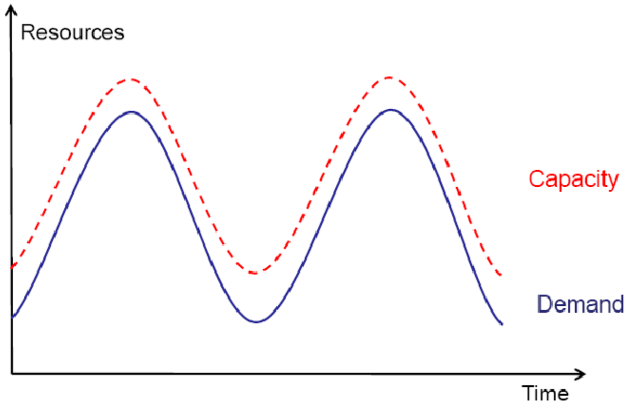
Service Level Objectives

Service Level Objectives (SLOs) specify requirements that must be ensured throughout operation (e.g., latency < *t*). Focused mainly on performance, narrows the scope

Elasticity Strategies scale a system according to current demand; e.g., if performance is insufficient, allocate more resources, change quality, adapt costs. However, what if this does not fulfill SLOs?

Edge Computing allows to decrease latencies for IoT applications; can use load-balancing mechanisms to direct load, but only scale resources up to a local limited



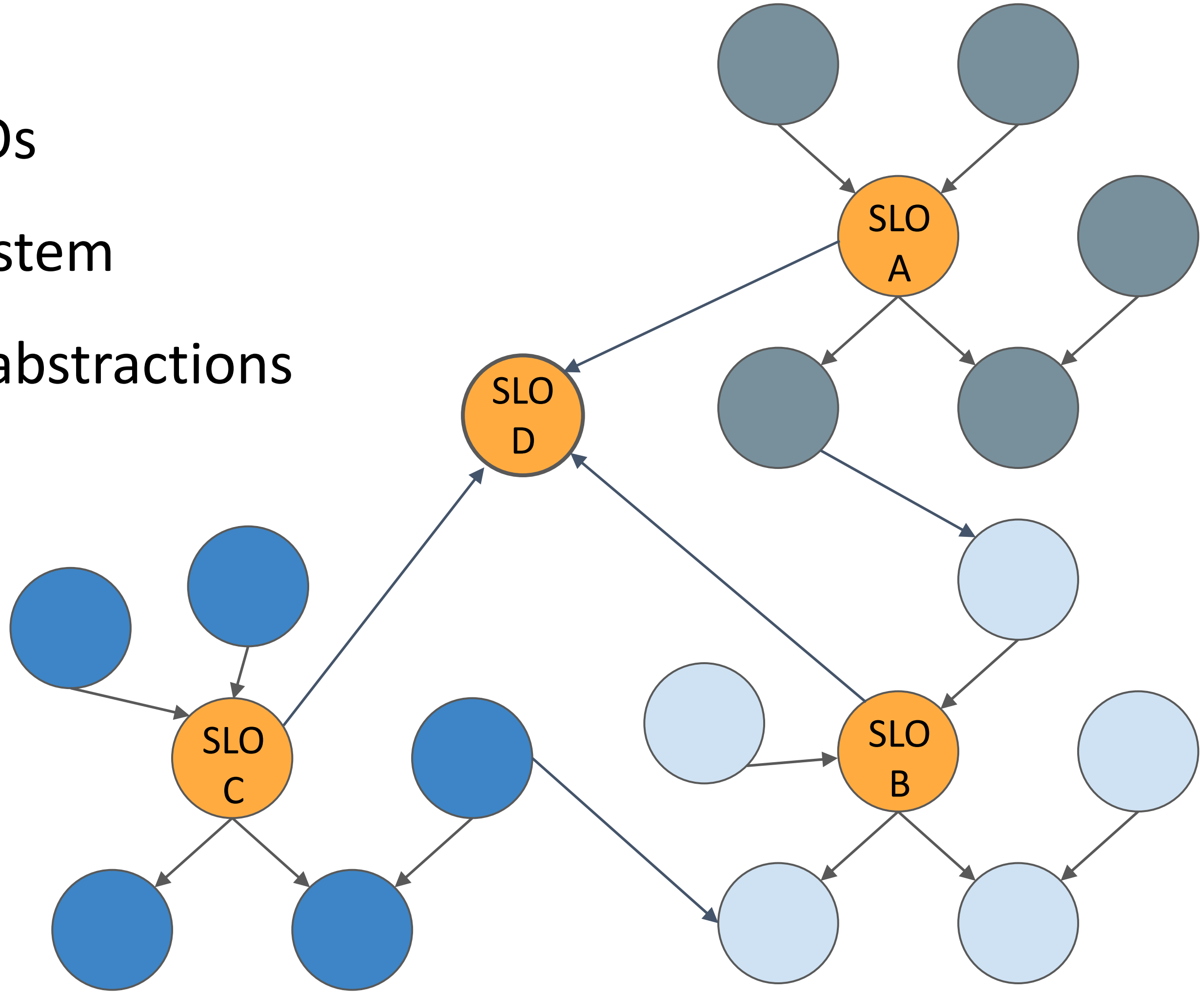


Elasticity allocates the right amount of resources [2]

[2] Ricciardi et al., **Saving Energy in Data Center Infrastructures** (2011)

DeepSLOs

- A construct we envision relating SLOs
- Provides a complete view of DCC system
- Allows aggregation towards higher abstractions



Problem Summary

Intricacy of requirements
Large-scale distributed systems are complex and their correct function requires more flexible ways to ensure SLOs

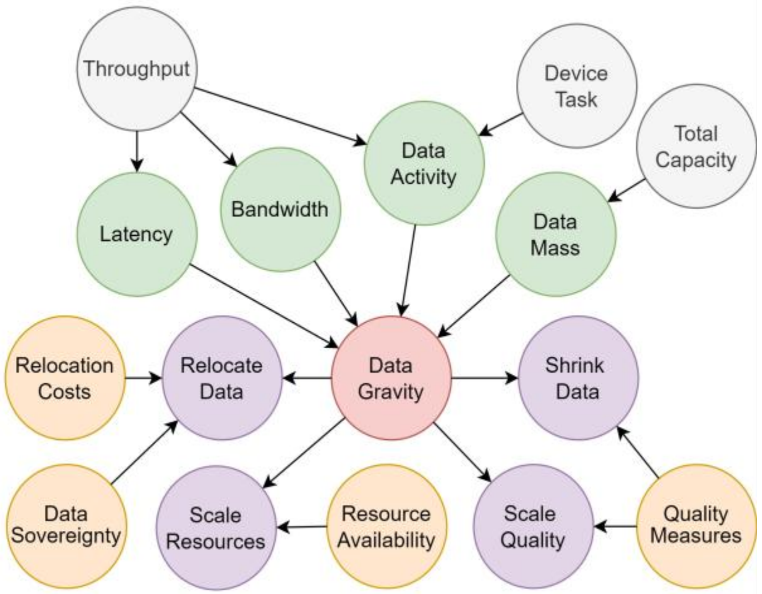
→ Composable behavioral models

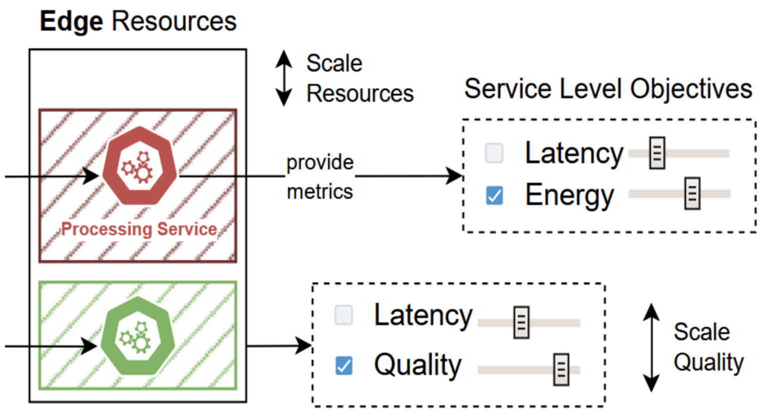
Resource limitations on Edge
Resources are scarce at the Edge and it might often not be possible to offload, scale vertically, or horizontally

→ Multi-dimensional elasticity strategies

ML algorithms as blackbox
Low trust in ML-based orchestration mechanisms (incl. autoscalers) that cannot be verified empirically

→ Causality-based service adaptation





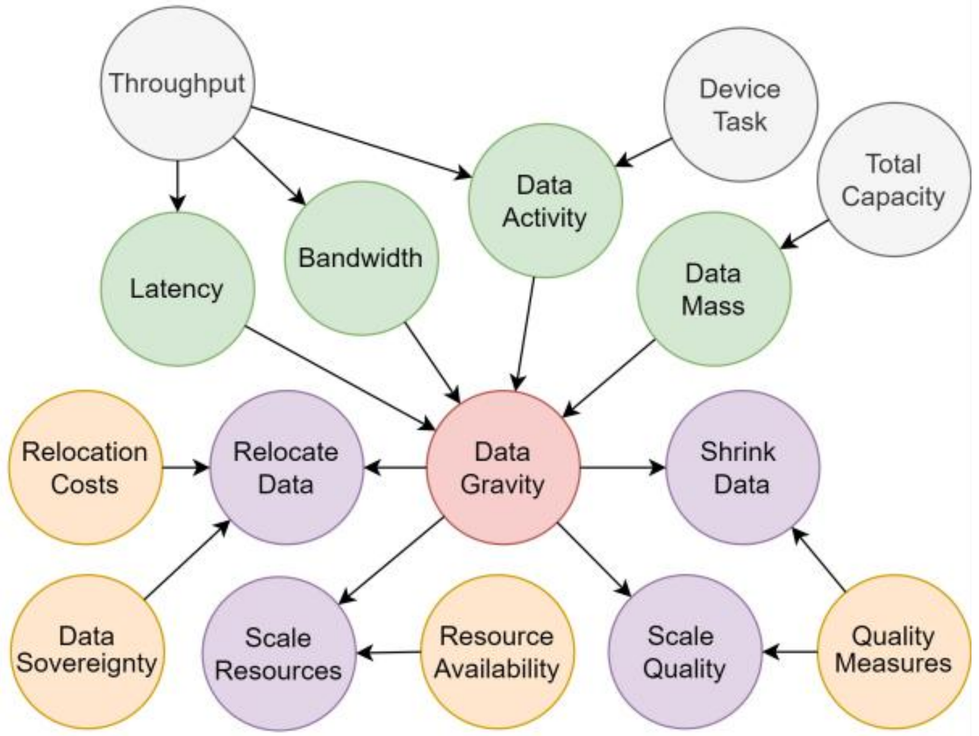
SLOs and Behavioral Models

MB: Expresses how to evaluate a composite SLO and how to react according to the current device context

Behavioral model
Internal state (●) evaluates objectives and how these relate to external sensory inputs (●);

can interact with the world through action, i.e., elasticity strategies (●),

which are influenced by contextual factors (●)



Example of a behavioral model for data gravity [3]

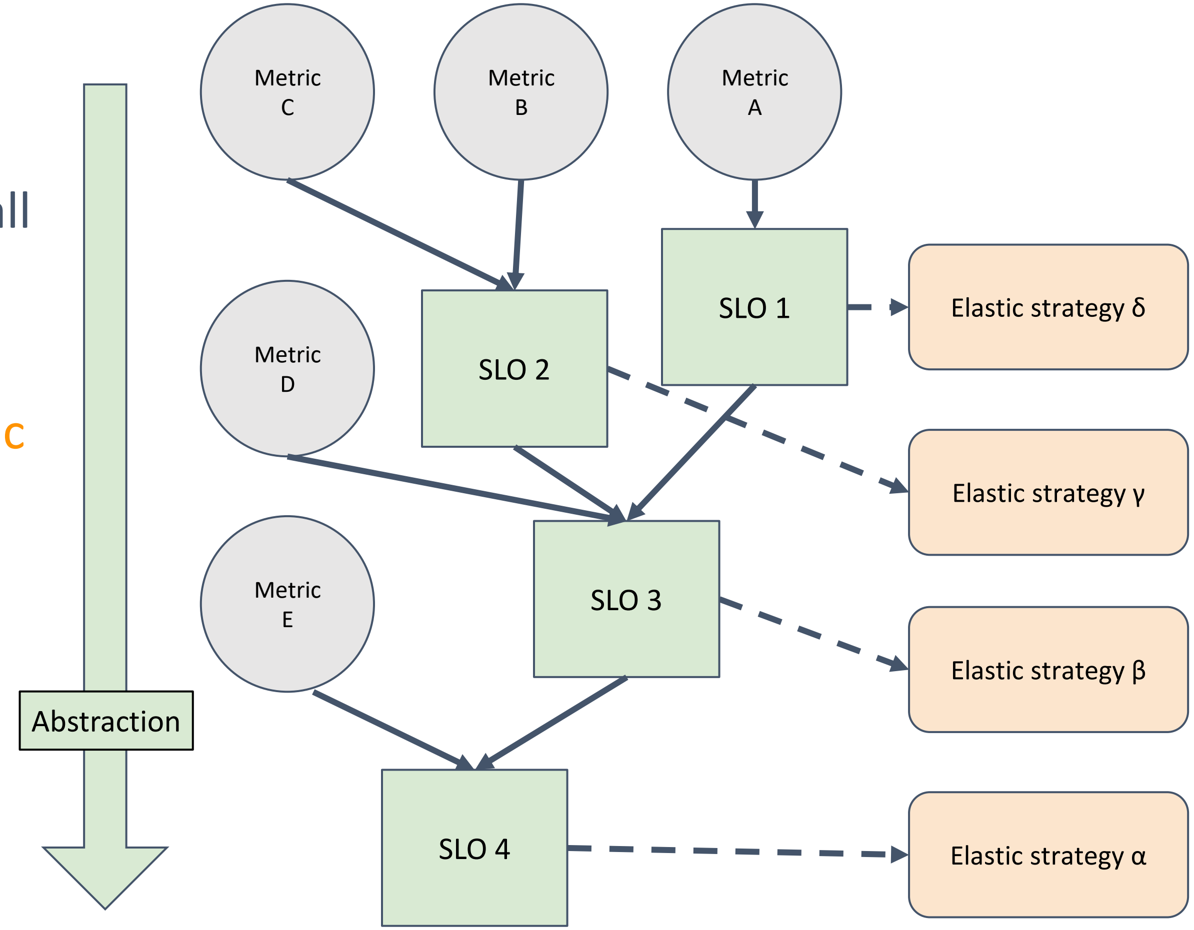
[3] Sedlak et al., **Controlling Data Gravity and Data Friction: From Metrics to Multidimensional Elasticity Strategies** (2023)

DeepSLOs

DeepSLOs as a hierarchically structured set of SLOs that relate causally and purposefully, holistically integrating all system needs.

1. A single DeepSLO can be in charge of an autonomic component of the system, providing ad-hoc objectives and elastic strategies at different abstraction levels, and mapping into the infrastructure.

2. Horizontal relations are within the same level of abstraction, vertical relations incorporate purpose and lead to different abstraction levels.



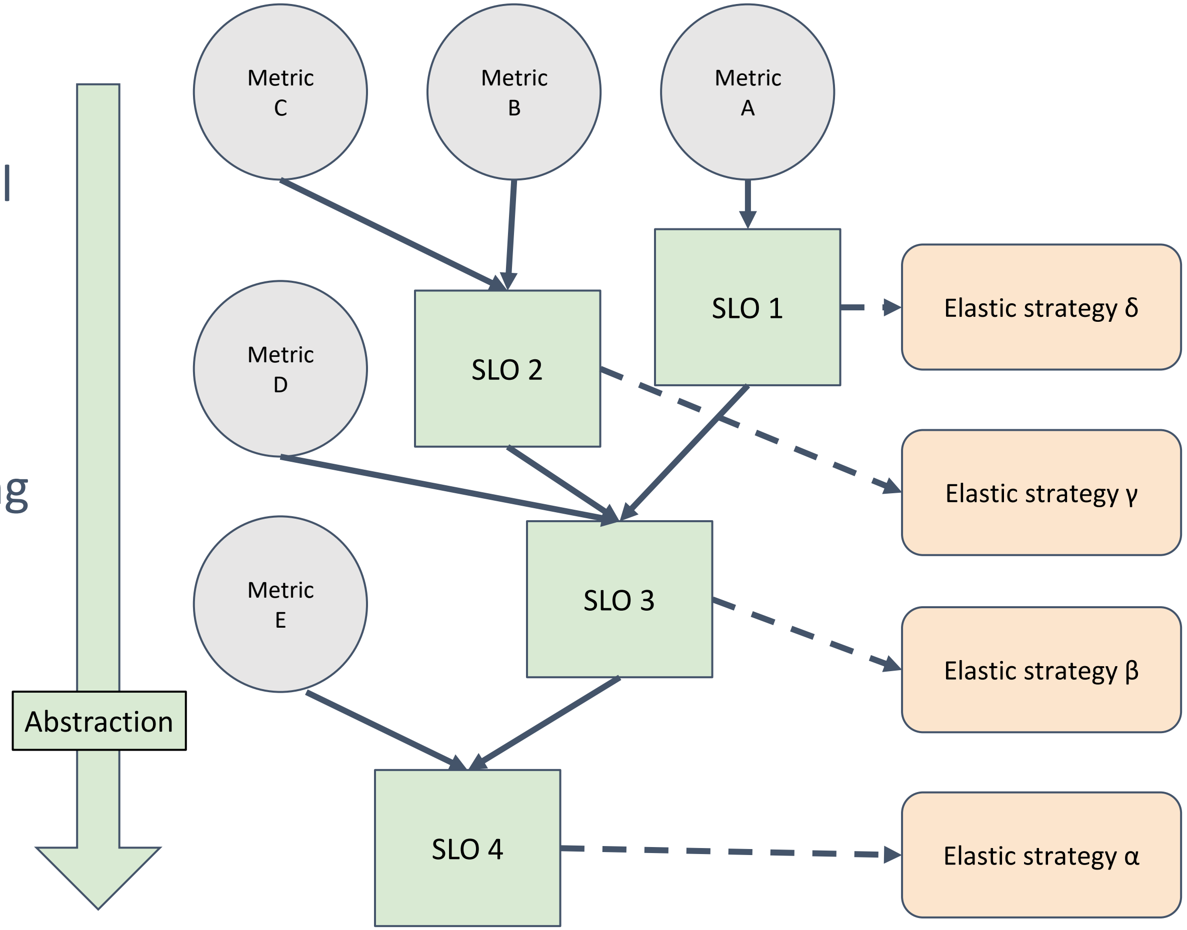
DeepSLOs

DeepSLOs as a hierarchically structured set of SLOs that relate causally and purposefully, holistically integrating all system needs.

3. A complete DCCS can be mapped with several DeepSLO that connect at their highest level, allowing each DeepSLO to properly propagate towards the infrastructure the shared objectives.

4. They provide a framework to solve the *multiple elasticity strategy problem*.

5. Integrate transversal features such as privacy, security, energy-efficiency, reliability…



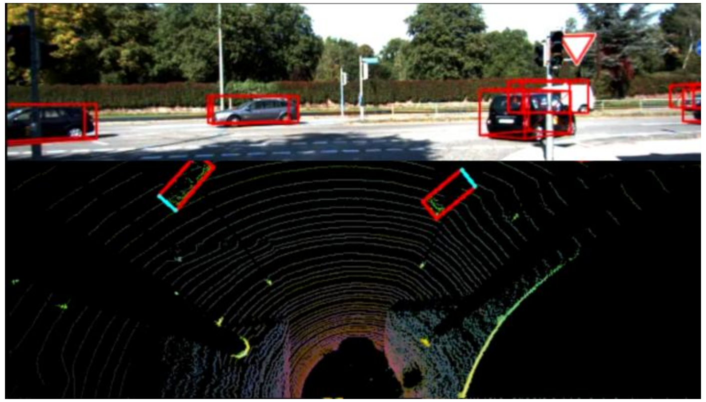
Stream Processing Scenarios

Commonly addressed use cases revolve around continuous **stream processing**, in case **time-critical** adaptations are required, this poses a higher need for sophisticated adaptation mechanisms.

Video Processing (Yolo V8)

Object detection in a video stream using Yolo [6]

Mobile Mapping (Lidar)



Creating a mobile map from binaries using Lidar [6]

QR Scanner (OpenCV)

QR code scanning in a video using OpenCV [6]

[6] Sedlak et al., **Adaptive Stream Processing on Edge Devices through Active Inference** (Scheduled for 2025)

Elastic Quality

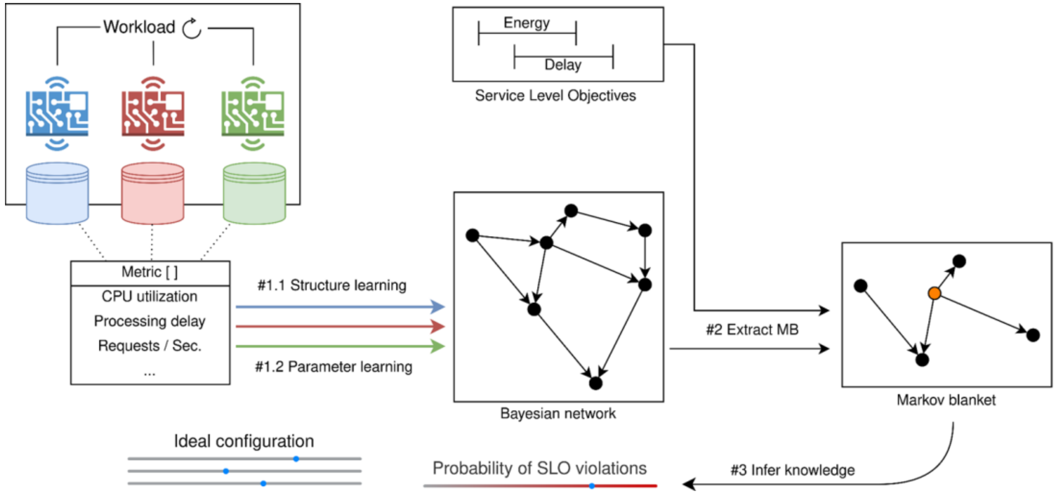
Problem: with limited resources, find alternative and effective strategies to ensure processing SLOs; use MB to create **interpretable** representation of service behavior; include relevant metrics and actions

Resulting model contains:

- Target objectives (i.e., SLOs)
- Factors influencing/depending SLOs
- Optimal system configuration

3-Step basic methodology for providing this model through (**1**) Bayesian Network Learning (BNL), (**2**) Markov Blanket (MB) extraction, and (**3**) Inference.

Jetson Xavier [7]

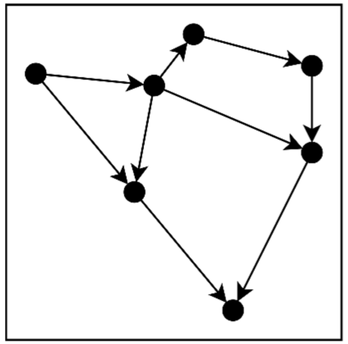


[7]

[7] Sedlak et al., **Designing Reconfigurable Intelligent Systems with Markov Blankets** (2023)

Elastic Quality (cont.)

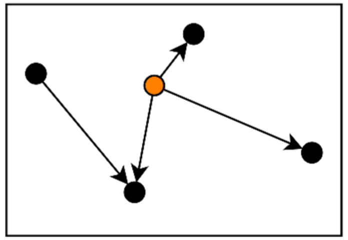
Bayesian Network Learning



Bayesian network

- **Structure Learning**
  Hill-Climb Search (HCS)
  Directed Acyclic Graph (DAG)
- **Parameter Learning**
  Max. Likelihood Estimation
  Conditional Prob. Table (CPT)

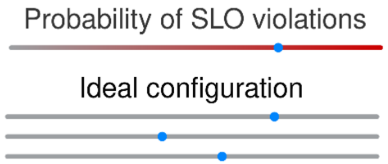
Markov Blanket Selection



Markov blanket

- **Causality filter**
  Extract a variable subset Create system interface
- Identify variables that impact **SLO fulfillment**

Knowledge Extraction



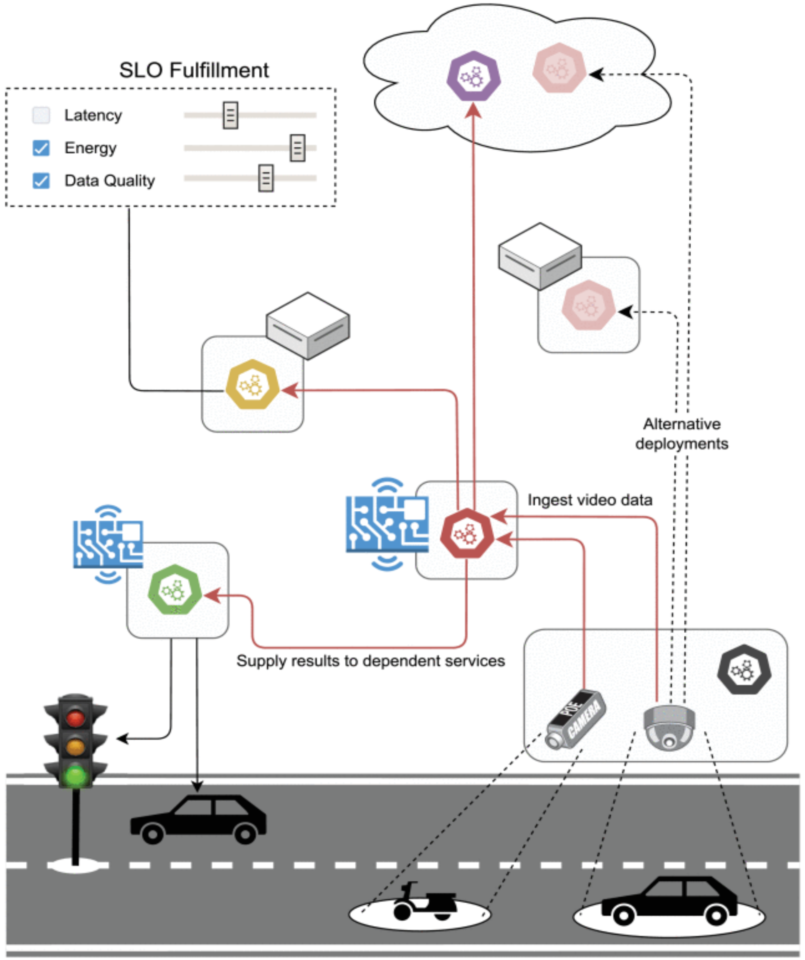
- P(SLO < x) for different variable combinations
- Find **Bayes-optimal** system configuration
- E.g., estimate impact of GPU, energy cons.

Generalizing Approach

Transitive Requirements [5]
SLOs by stream consumers determine the service quality that each “link” has to provide; **compose** MBs of dependent services to find implications and optimize deployment

Spanning CC with SLOs [6]
Microservice architectures composed of various services with SLOs for user-facing layer, e.g., latency or quality; infer lower-level SLOs and parameters for influential services

SLO-Aware Offloading [7]
Offloading a task to a resource-restricted device jeopardizes SLO fulfillment of existing services; estimate the implication to global SLO fulfillment to find suitable device hosts

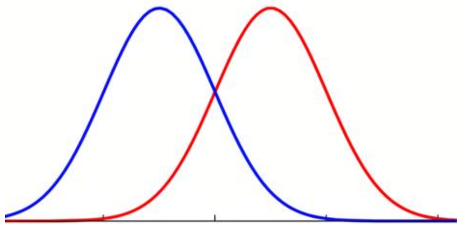


Optimizing the deployment of microservice pipelines according to the SLOs posed for each service [5]

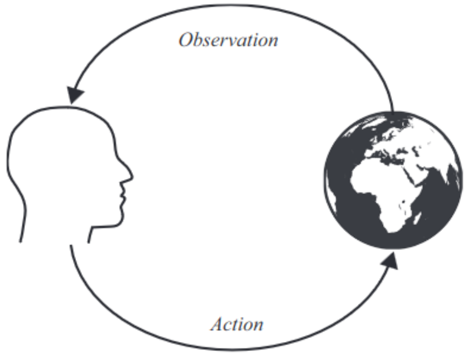
[6] Sedlak et al., **Diffusing High-level SLO in Microservice Pipelines** (2024)
[7] Sedlak et al., **SLO-Aware Task Offloading Within Collaborative Vehicle Platoons** (2024)

Refining Approach

Known Shortcomings
(**1**) BNL requires large amounts of training data in upfront;
(**2**) if discrete, must visit all possible states (e.g., scaling actions);
(**3**) over time, models get distorted due to variable drifts



Active Inference
Concept from **neuroscience** developed by Friston et al. [7,8]; allows agents to interact with their environment by learning the underlying **generative models** to persist over time



Action-perception cycle [7]

[7] Parr, Pezzulo, and Friston; Active Inference: The Free Energy Principle in Mind, Brain, and Behavior (2022)
[8] Friston et al., **Designing ecosystems of intelligence from first principles** (2024)

Active Inference

Describes how systems maintain their states and make predictions about their environment to minimize free energy.

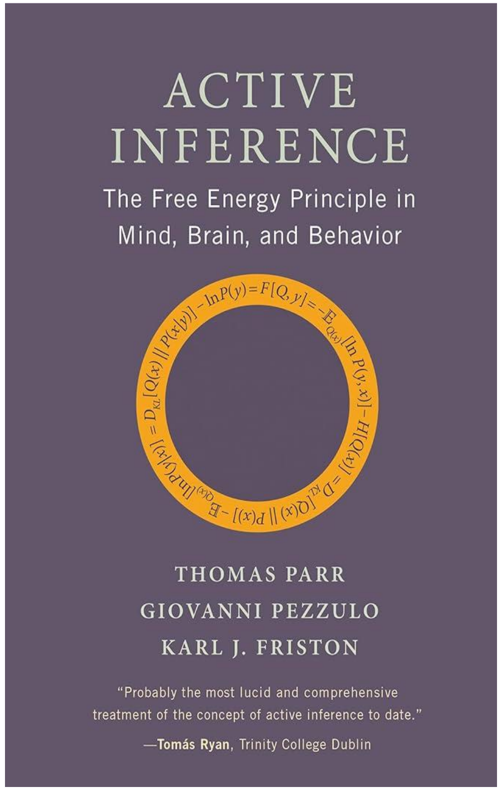
Active Inference Framework

1. **Objective**: Minimize free energy (a measure of uncertainty) to maintain homeostasis and predict environmental changes.

2. **Core Concepts**:
   - Free Energy Principle: Systems minimize the difference between predicted and actual sensory inputs.
   - Bayesian Inference: Use of probabilistic models to update beliefs about the state of the world based on new data.
   - Generative Models: Systems use models to generate predictions about sensory inputs and outcomes.

3. **Mechanisms**:
   - Perception: Involves updating beliefs about the state of the environment based on sensory inputs.
   - Action: Involves selecting actions that minimize expected free energy by reducing prediction errors.
   - Learning: Adjusting the parameters of generative models to improve future predictions and actions.

Active Inference is a framework that integrates perception, action, and learning through Bayesian inference and generative models to minimize prediction errors and free energy. It has significant applications in fields like robotics, machine learning, and cognitive computing, where systems need to predict, adapt, and learn from their environment efficiently.

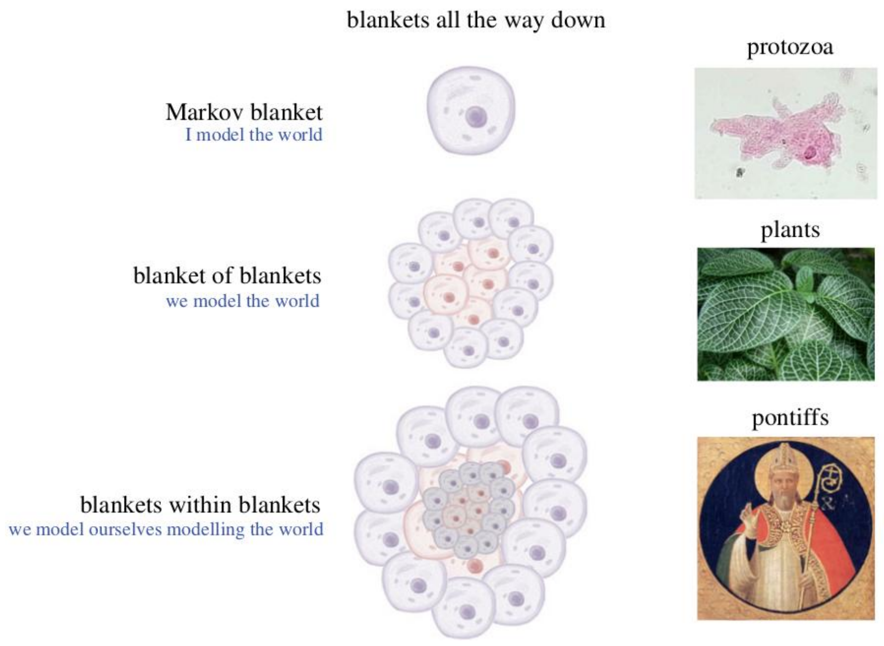


[1] Friston et al.,Designing Ecosystems of Intelligence from First Principles, https://doi.org/10.48550/arXiv.2212.01354
[2] Friston, Life as we know it, https://doi.org/10.1098/rsif.2013.0475
[3] Palacios et al., On Markov blankets and hierarchical self-organisation, https://doi.org/10.1016/j.jtbi.2019.110089
[4] Kirchhoff et al., The Markov blankets of life: autonomy, active inference and the FEP, https://doi.org/10.1098/rsif.2017.0792
[5] Parr et al., Active Inference: The Free Energy Principle in Mind, Brain, and Behavior, https://doi.org/10.7551/mitpress/12441.001.0001

Research Scope

Intersection between distributed service assurance and Active Inference:

- **Structural causal models**
  - **Causality** to tame large scale networks
  - Revealing and managing dependencies
- **Self-evidenced cellular structures**
  - Evaluate continuously how to fulfill SLOs
  - Based on empirical values (i.e., metrics)
- **Homeostasis – Equilibrium**



[3]

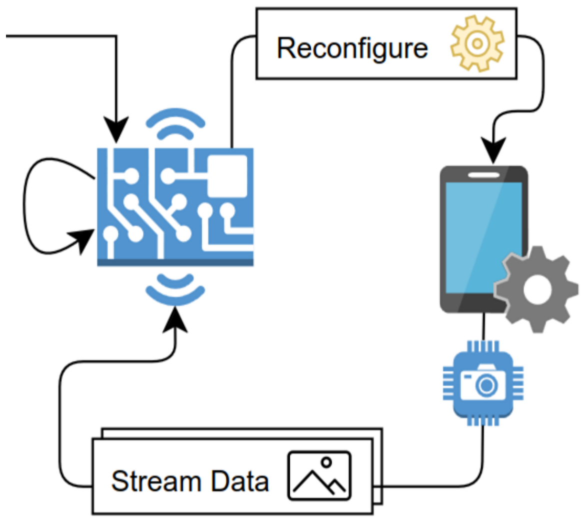
[3] Kirchhoff et al., The Markov blankets of life: autonomy, active inference and the FEP, https://doi.org/10.1098/rsif.2017.0792

Active Inference applied

1. Mapping between neuroscience and distributed computing systems [6,15,16]; understanding processing requirements (i.e., SLOs) as a form of **homeostasis,** e.g., cell temperature

2. Create autonomous components that identify how to ensure requirements and resolve them independently, clear modelling between higher-level and low-level components

3. Simplify service orchestration in large-scale distributed systems, such as Computing Continuum; **encapsulation** and decentralized decision-making of individual components



Ensure internal requirements [15]

[15] Sedlak et al., **Active Inference on the Edge: A Design Study** (2024)
[16] Sedlak et al., **Equilibrium in the Computing Continuum through Active Inference** (2024)

AIF Architecture in a Nutshell

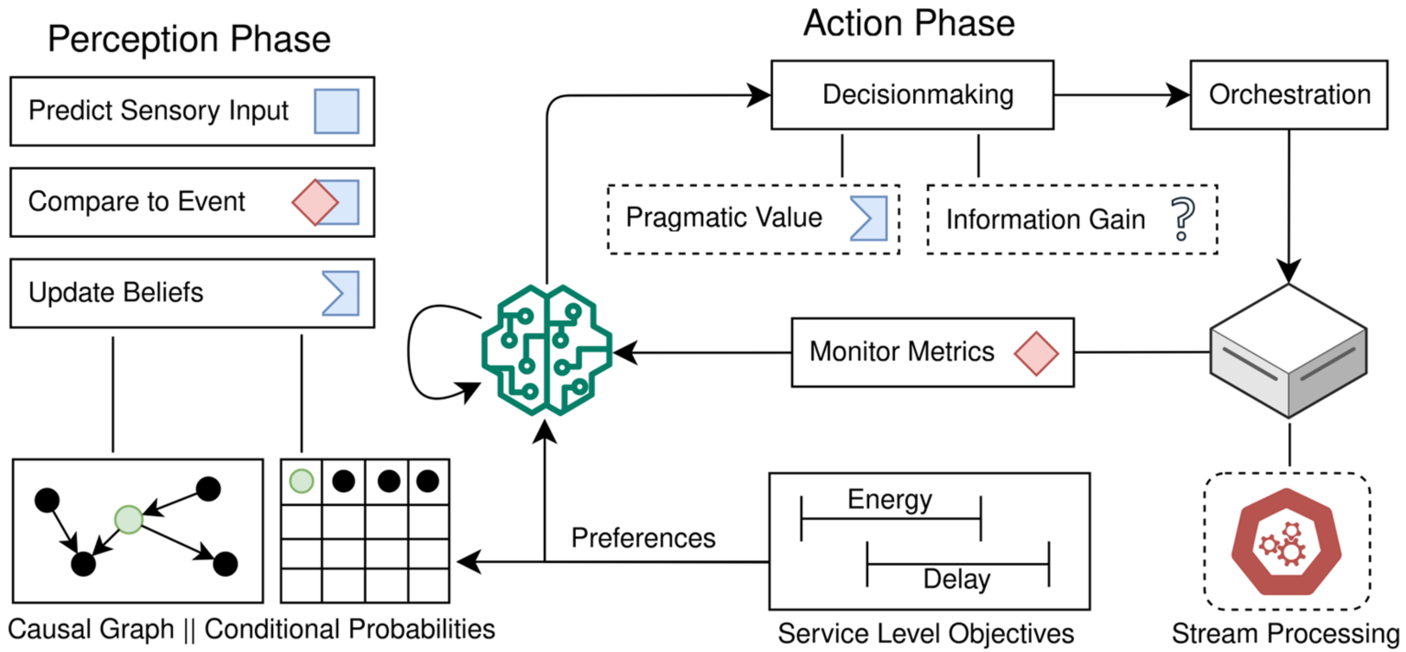
Approach

(1) **Specify** ideal runtime behavior through SLOs

(2) **AIF agents** perceive their environment and enact on it

(3) **Perception** predicts the expected SLO fulfillment and adjusts the generative model

(4) **Action** phase reconfigure local processing environment to minimize FE and fulfill SLOs



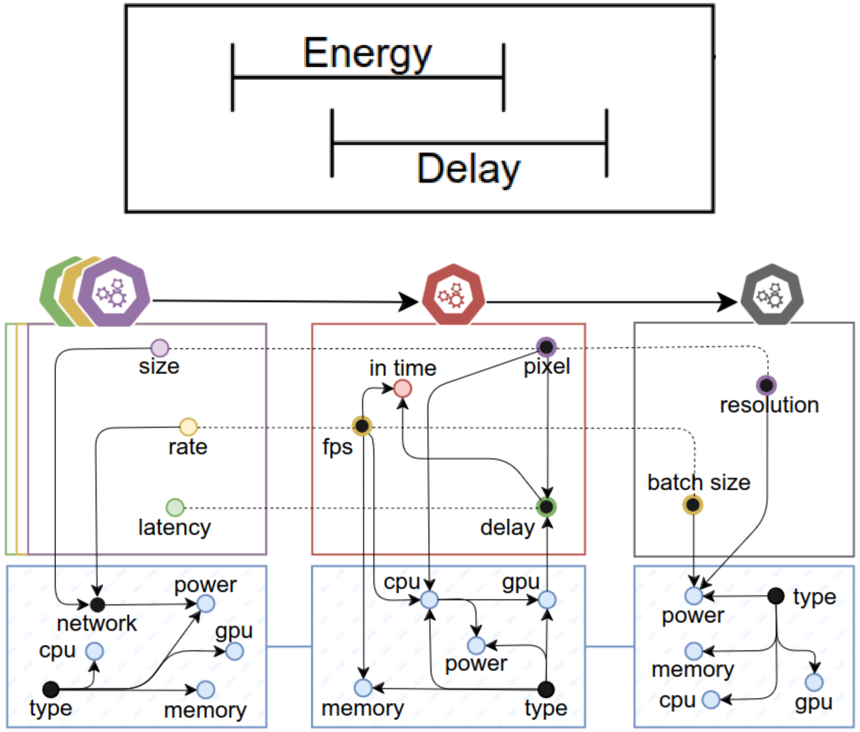
Action and perception cycles performed by the AIF agent to create an accurate model and shape the world [6]

Summary

IoT & Edge create countless applications for human benefit; pose challenges due to resource limits, for which the Computing Continuum can be a remedy

Processing SLOs must be continuously ensured; presented mechanisms designed to ensure SLO fulfillment and scale services in multiple dimensions

Active Inference as natural fit with MB & behavioral models; extend the methodology for maintaining generative models accurate and react dynamically



Preliminary Work

- Local Requirements assurance by employing BN and MB [6] →

  "Static Bayesian Network Learning"

- Design Study for AIF agents in distributed systems [7]


Designing Reconfigurable Intelligent Systems with Markov Blankets

Boris Sedlak, Victor Casamayor Pujol, Praveen Kumar Donta, and Schahram Dustdar

Distributed Systems Group, TU Wien, 1040 Vienna, Austria


Active Inference on the Edge: A Design Study*

Boris Sedlak, Victor Casamayor Pujol, Praveen Kumar Donta, and Schahram Dustdar

Distributed Systems Group, Vienna University of Technology (TU Wien), Vienna 1040, Austria.

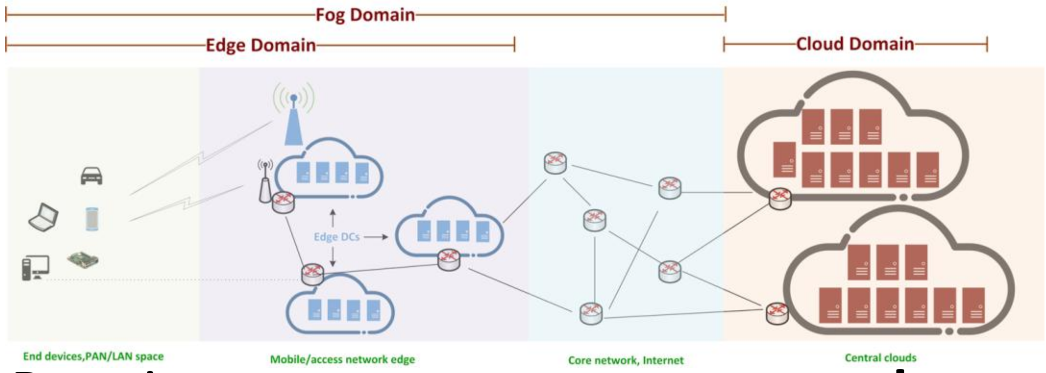
Distributed Intelligence in the Computing Continuum with Active Inference, Casamayor V., Sedlak, B., Salvatori, T., Friston, K., Dustdar, S., under review

[6] Designing Reconfigurable Intelligent Systems with Markov Blankets, ICSOC 2023, https://doi.org/10.1007/978-3-031-48421-6_4

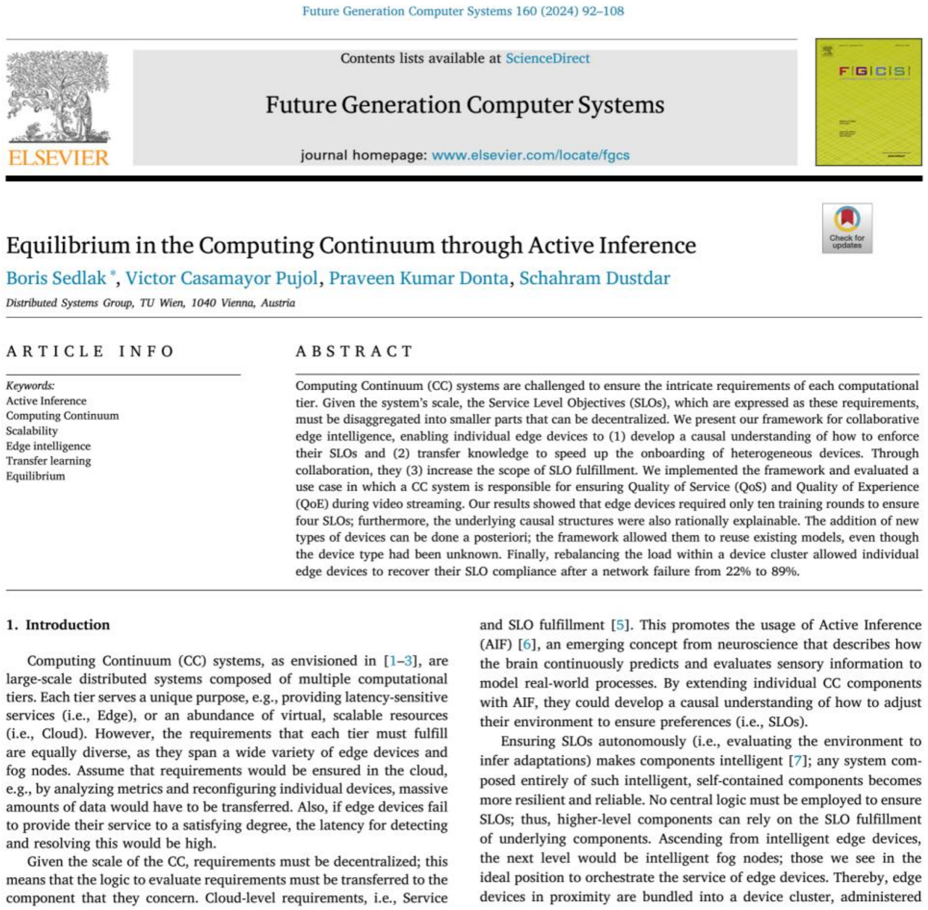
[7] Active Inference on the Edge: A Design Study, pending at IEEE PerconAI 2024, https://doi.org/10.48550/arXiv.2311.10607

Equilibrium in the CC through Active Inference

- Core problem stems from **CC architecture**
- Impossible to centrally evaluate requirements
- Heterogeneity and context-dependence



- Requires components to operate **decentralized**
- Devices unaware of how to fulfill their SLOs
- Active Inference can provide this knowledge



Future Generation Computer Systems 160 (2024) 92–108

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Equilibrium in the Computing Continuum through Active Inference

Boris Sedlak *, Victor Casamayor Pujol, Praveen Kumar Donta, Schahram Dustdar

Distributed Systems Group, TU Wien, 1040 Vienna, Austria

ARTICLE INFO

Keywords:
Active Inference
Computing Continuum
Scalability
Edge intelligence
Transfer learning
Equilibrium

ABSTRACT

Computing Continuum (CC) systems are challenged to ensure the intricate requirements of each computational tier. Given the system's scale, the Service Level Objectives (SLOs), which are expressed as these requirements, must be disaggregated into smaller parts that can be decentralized. We present our framework for collaborative edge intelligence, enabling individual edge devices to (1) develop a causal understanding of how to enforce their SLOs and (2) transfer knowledge to speed up the onboarding of heterogeneous devices. Through collaboration, they (3) increase the scope of SLO fulfillment. We implemented the framework and evaluated a use case in which a CC system is responsible for ensuring Quality of Service (QoS) and Quality of Experience (QoE) during video streaming. Our results showed that edge devices required only ten training rounds to ensure four SLOs; furthermore, the underlying causal structures were also rationally explainable. The addition of new types of devices can be done a posteriori; the framework allowed them to reuse existing models, even though the device type had been unknown. Finally, rebalancing the load within a device cluster allowed individual edge devices to recover their SLO compliance after a network failure from 22% to 89%.

1. Introduction

Computing Continuum (CC) systems, as envisioned in [1–3], are large-scale distributed systems composed of multiple computational tiers. Each tier serves a unique purpose, e.g., providing latency-sensitive services (i.e., Edge), or an abundance of virtual, scalable resources (i.e., Cloud). However, the requirements that each tier must fulfill are equally diverse, as they span a wide variety of edge devices and fog nodes. Assume that requirements would be ensured in the cloud, e.g., by analyzing metrics and reconfiguring individual devices, massive amounts of data would have to be transferred. Also, if edge devices fail to provide their service to a satisfying degree, the latency for detecting and resolving this would be high.

Given the scale of the CC, requirements must be decentralized; this means that the logic to evaluate requirements must be transferred to the component that they concern. Cloud-level requirements, i.e., Service and SLO fulfillment [5]. This promotes the usage of Active Inference (AIF) [6], an emerging concept from neuroscience that describes how the brain continuously predicts and evaluates sensory information to model real-world processes. By extending individual CC components with AIF, they could develop a causal understanding of how to adjust their environment to ensure preferences (i.e., SLOs).

Ensuring SLOs autonomously (i.e., evaluating the environment to infer adaptations) makes components intelligent [7]; any system composed entirely of such intelligent, self-contained components becomes more resilient and reliable. No central logic must be employed to ensure SLOs; thus, higher-level components can rely on the SLO fulfillment of underlying components. Ascending from intelligent edge devices, the next level would be intelligent fog nodes; those we see in the ideal position to orchestrate the service of edge devices. Thereby, edge devices in proximity are bundled into a device cluster, administered

[8] Equilibrium in the Computing Continuum through Active Inference, FGCS 160 (2024), 92-108

Research line - Model

Markovian models

- Markov blanket (DAG)
- Markov fields (non directed graphs)
- Markov chains

Deep neural networks

- Federated learning
- Graph neural networks

Agent based

- Active inference
- Reinforcement learning

- How to deal with a multimodal environment?

Incorporate data from video sources, results from video processing units, quality of the predictions, overall system cost…

- How to model relations?

The shortage of computing power on an edge device will affect overall control system, but how much?

- How to treat abstraction?

Include concepts of cost or quality along with basic infrastructure metrics, i.e. number of drivers detected at the phone and GPU usage in the same framework.

- How to obtain enough data?

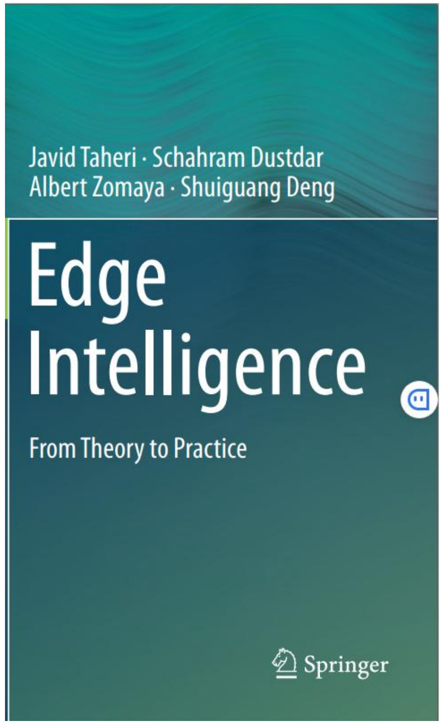
Large, hyper-distributed and open systems. How to know the system is accurate?

- And many more… How to deal with IID data? How to tackle uncertainty?

Conclusions

1. Distributed Computing Continuum from IoT->Edge->Fog->Cloud
2. Distributed Intelligence
3. SLOs, Markov Blankets, Active Inference

Thanks for your attention



Prof. Schahram Dustdar

IEEE Fellow | EAI Fellow | I2CIC Fellow | AAIA Fellow

Member *Academia Europaea*

President of the AIIA (Asia-Pacific Artificial Intelligence Association)

ACM Distinguished Scientist | ACM Distinguished Speaker

TCI Distinguished Service Award by the IEEE Technical Committee on the Internet (TCI)

IEEE TCSVC Outstanding Leadership Award in Services Computing

IEEE TCSC Award for Excellence in Scalable Computing

IBM Faculty award