Distributed Intelligence through Active Inference

13 February 2024

Discussion starter for TNB, VERSES AI Inc, and DSG (TUW)

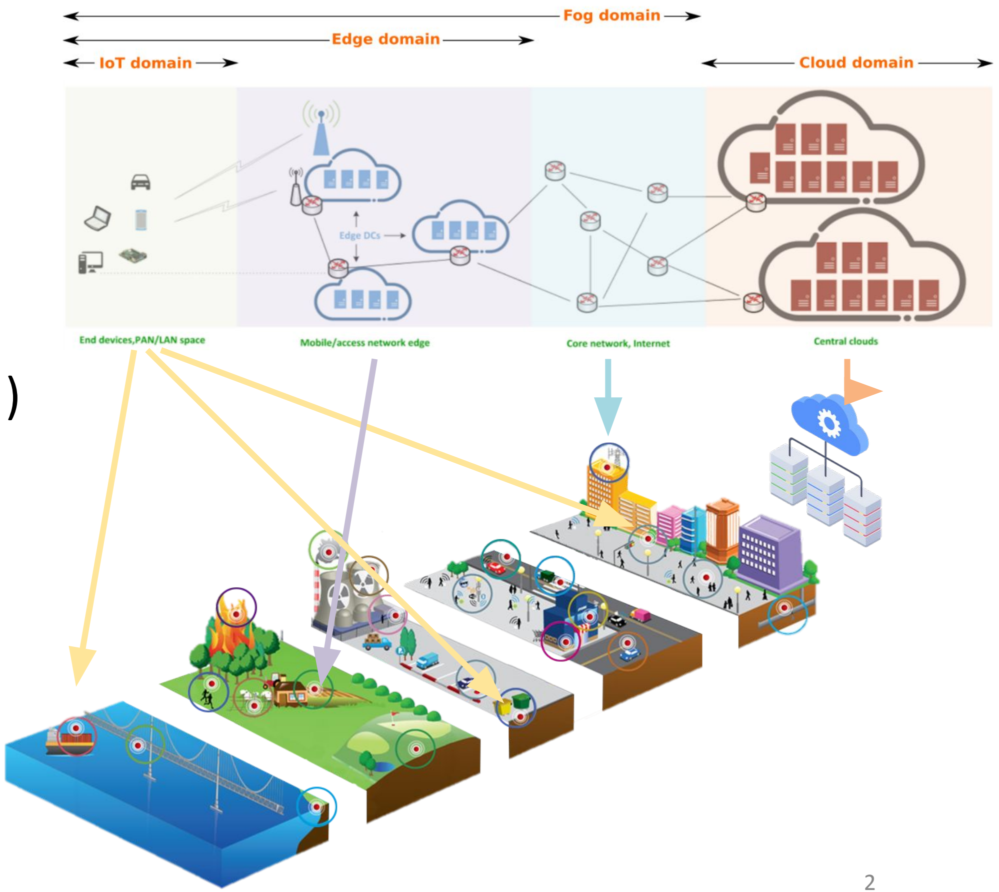
Schahram Dustdar, Boris Sedlak, Víctor Casamayor Pujol,
Praveen Kumar Donta, and Andrea Morichetta

dsg.tuwien.ac.at

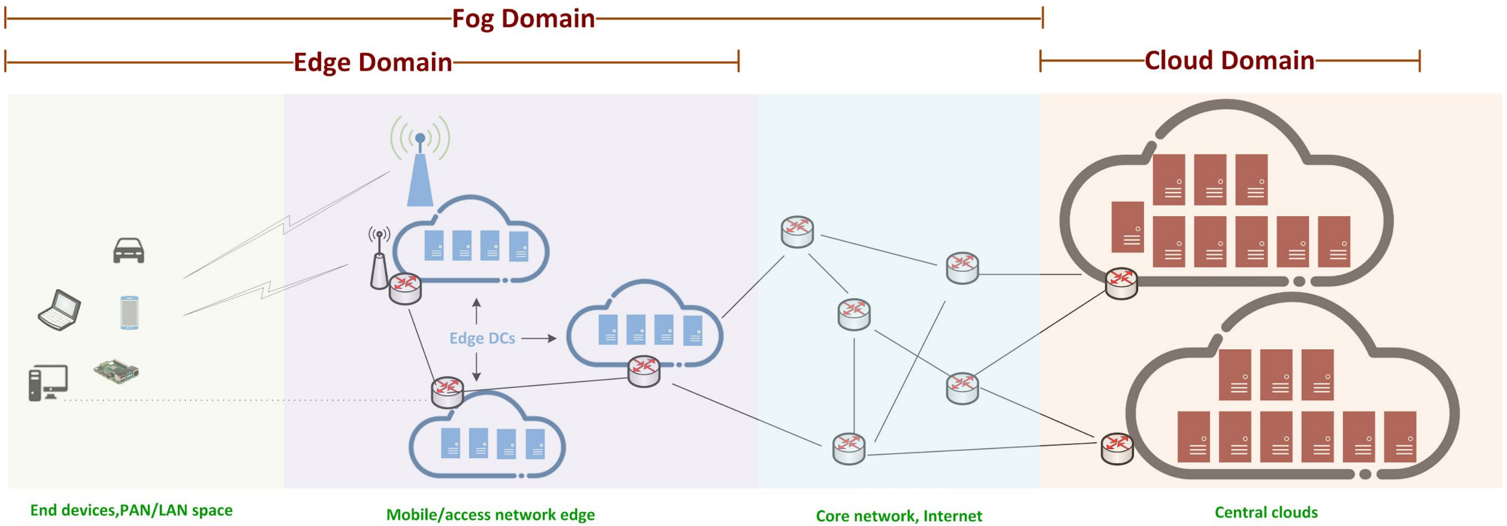
dsg DISTRIBUTED SYSTEMS GROUP

Current State

- Distributed Systems are key to our society
- Underlie our critical infrastructures and applications (Smart cities, Healthcare, Autonomous vehicles,…)
- Interconnectedness (fabric) of components (HW, SW, People) induces complexity
- We increasingly see fundamental issues we need to address



Distributed Compute Continuum: A high level view



Low reliability

Volatility

Mobility

(Mostly) Wireless connectivity

Small form factor

Battery constraints

Mobile, IoT, smart home, vehicles, …

User/Service provider controlled

Edge of the (mobile) network

Low latency to end device

Close to/collocated with 4G/5G base stations

General purpose compute infrastructure

Standards-based architectures & management/orchestration stacks

Telecom operator controlled

"Unlimited" compute/storage resources

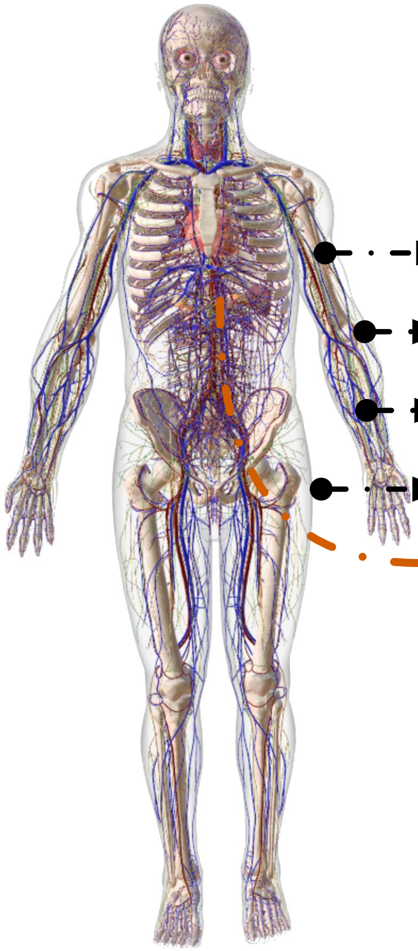
Full spectrum of cloud services

High availability

Lower cost

Higher latency vs. edge/fog

Cloud provider controlled



Human Ecosystem

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

Skeletal System

Nervous System

Cardiovascular System

Lymphatic System

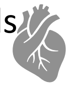
Endocrine System

Infrastructure Systems

Regulation Systems

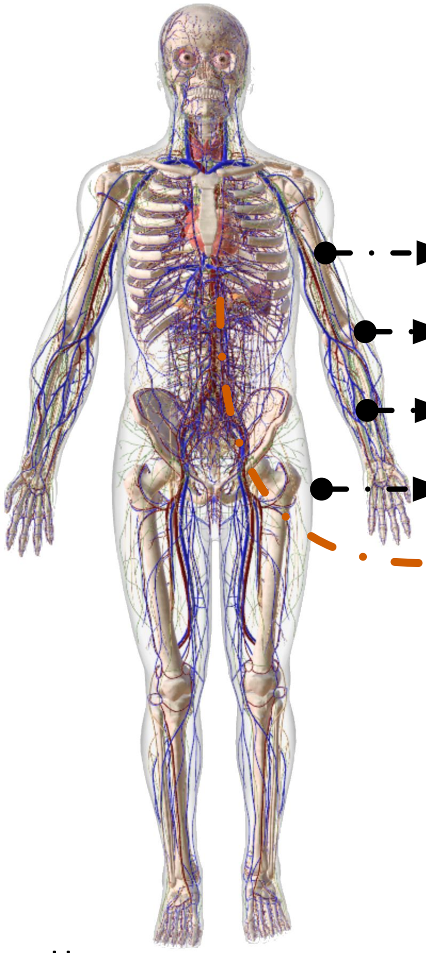
- Brain
- Spinal Cord
- Cranial Nerves
- Spinal Nerves

- 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

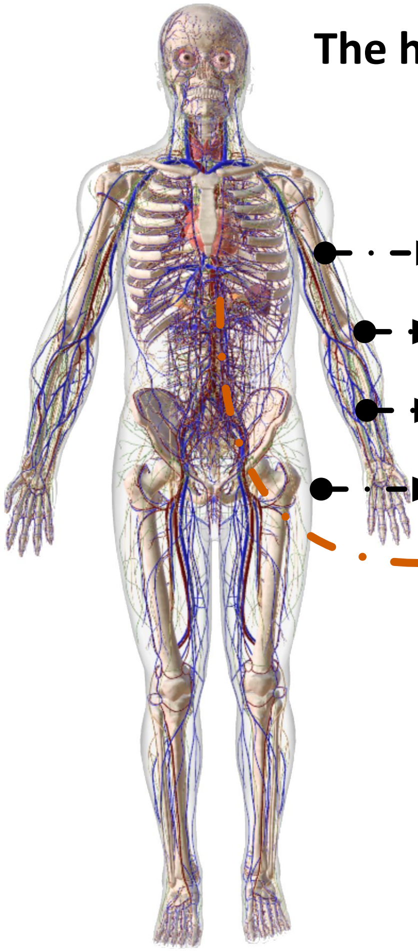
The human body is comprised of a series of complex systems, including:

- 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

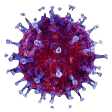
Regulation Systems

DeepSLOs

Collaborative Learning

Representation Learning

Zero Trust



- Part of the immune system
- Protects your body against foreign invaders

Distributed Computing Continuum Systems are composite complex systems

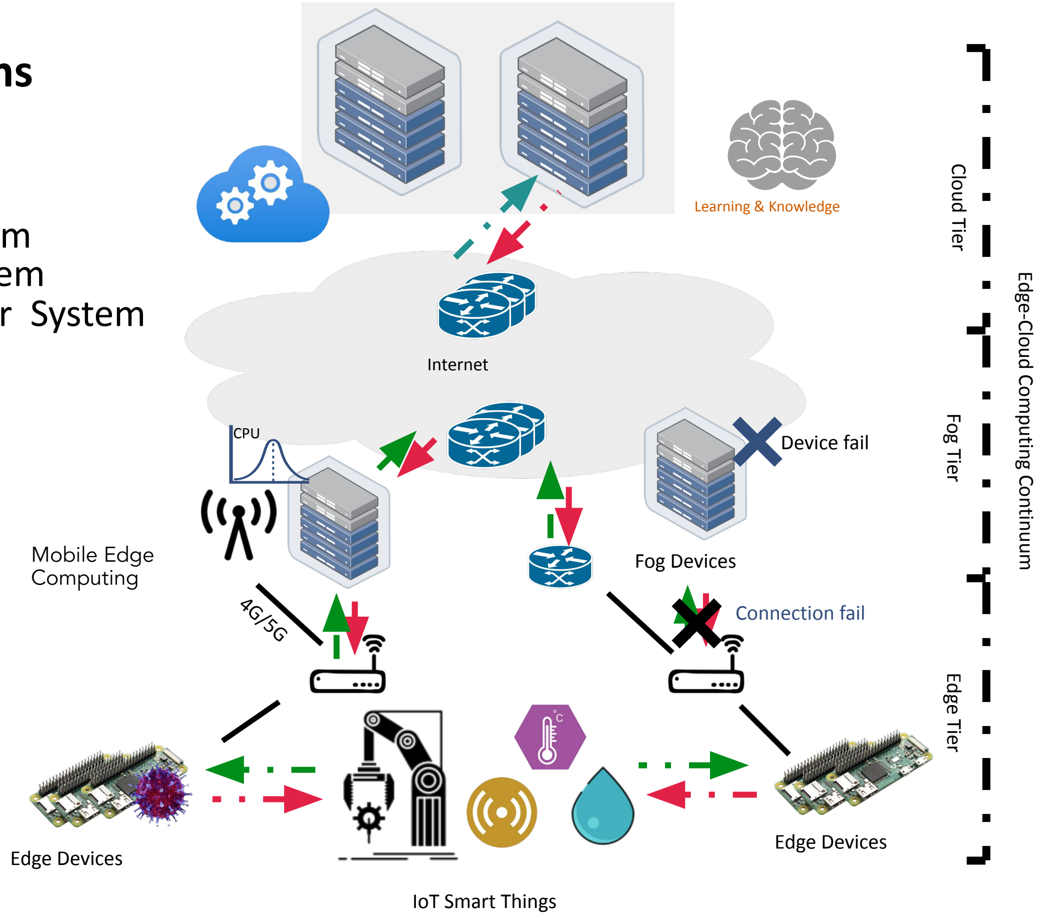
Infrastructure Systems

Skeletal System
Nervous System
Cardiovascular System
⋮

- Devices & Sensors
- Connection & Communication
- Data Flow
- Learning & Knowledge

Regulation Systems

- Elasticity Systems
- Self-adaptive Systems
- Fault-tolerance Systems
- Privacy & Security



Elasticity (Resilience)

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

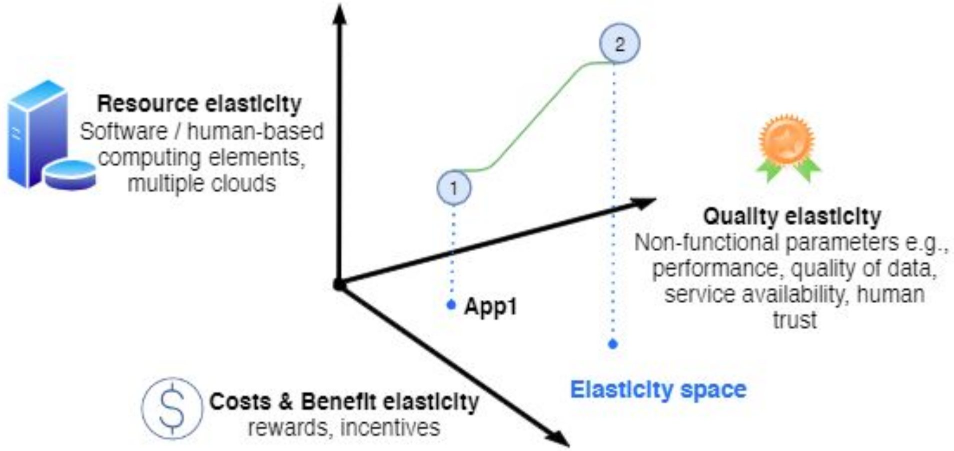
stretch when a force stresses them

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

shrink when the stress is removed

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

Elasticity > Scalability

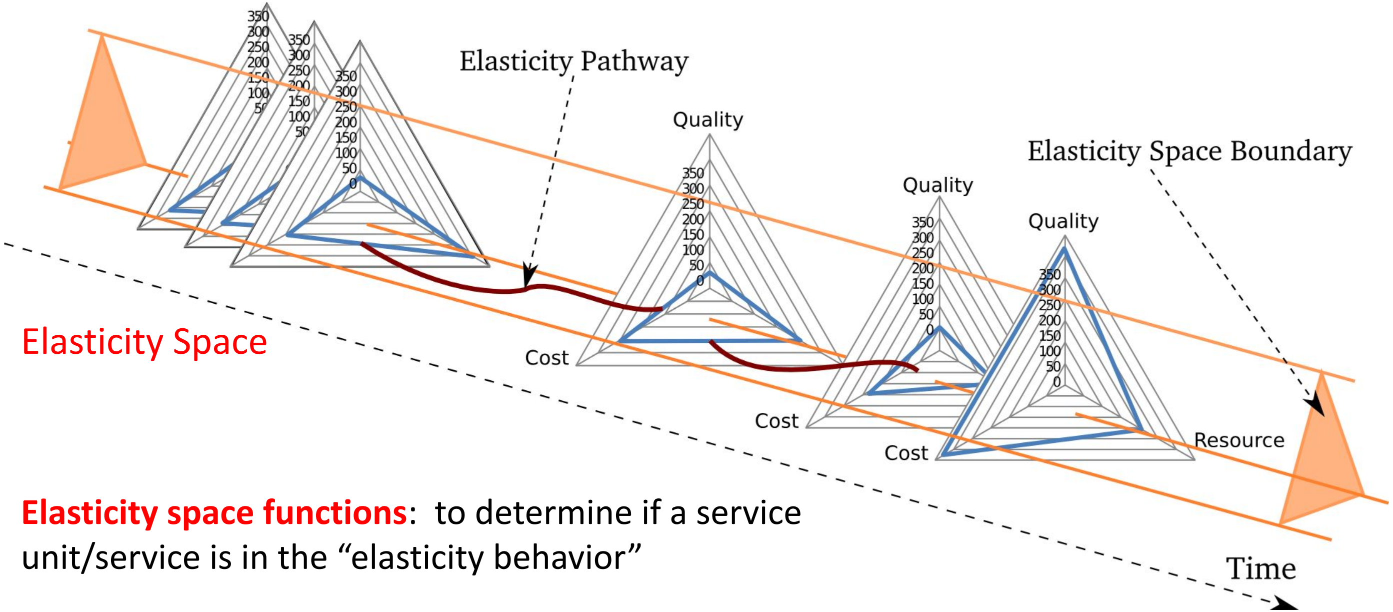


Dustdar S., Guo Y., Satzger B., Truong H. (2012) [Principles of Elastic Processes](), IEEE Internet Computing, Volume: 16, [Issue: 6](), Nov.-Dec. 2012

Elasticity Model for Edge & Cloud Services

Moldovan D., G. Copil,Truong H.-L., Dustdar S. (2013). **MELA: Monitoring and Analyzing Elasticity of Cloud Service. CloudCom 2013**

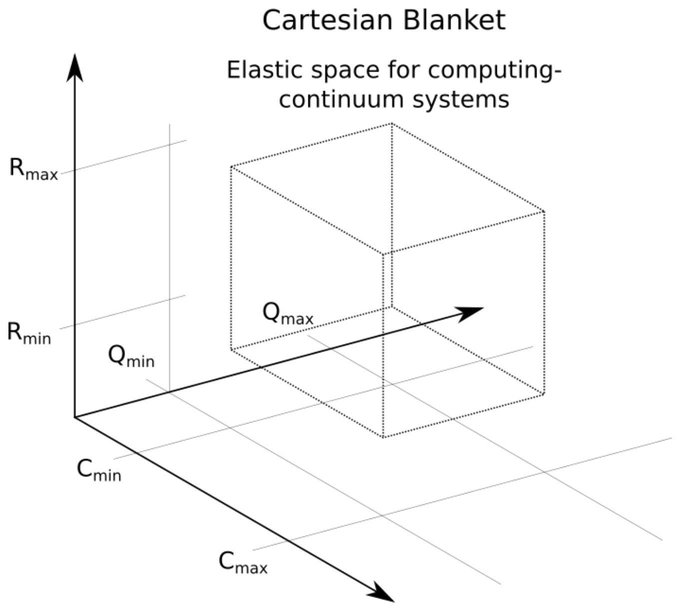
Elasticity Pathway functions: to characterize the elasticity behavior from a general/particular view



Elasticity space functions: to determine if a service unit/service is in the “elasticity behavior”

The Cartesian Blanket
Adapting elasticity in the continuum

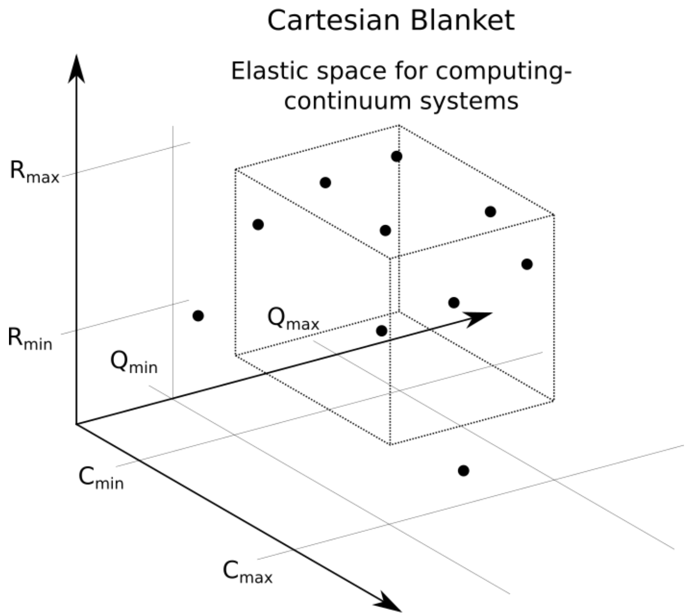
- System control based SLOs (Service Level Objectives)
- SLOs are represented as thresholds on the Cartesian space
- The system space is delimited within an hexahedron.
    - There is minimum and maximum value for each variable



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.

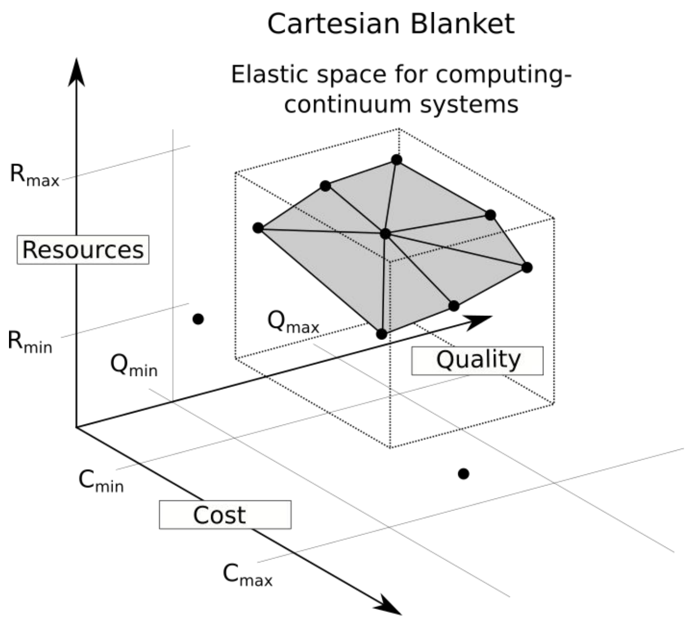


The Cartesian Blanket

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.

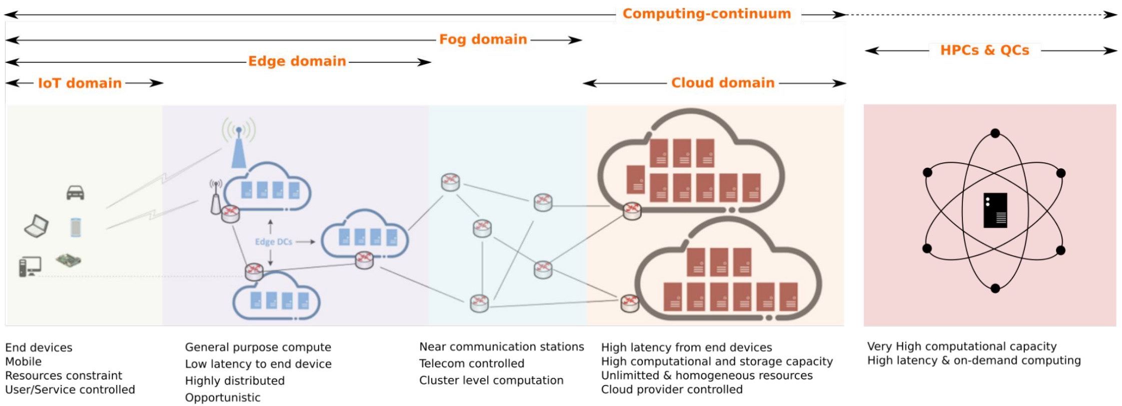
- Now we have the system represented, but



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

Infrastructure

➢ Computing continuum



➢ Application performance highly dependent on the underlying infrastructure
  ○ Heterogeneity of resources & heterogeneous distribution
  ○ Resources diverse interconnection

➢ Sustainability

Infrastructure & Applications – Modeling issues

How we model these systems? What is the “self” for the system?

Centralized vs. Agent-based

- Composability / Nested capacity / Dynamic configuration.

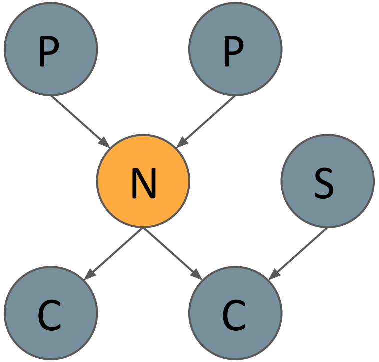
Intelligence and Behavior

- Bring *intelligence* to the underlying infrastructure
- Let’s use SLOs for that!
- But, let’s talk about them
  - Not *only* business-oriented
  - At different levels of the system
    - Devices
    - Services
    - Application
    - …
  - Mechanisms to control interactions and system components
  - Tailored elasticity strategies

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).

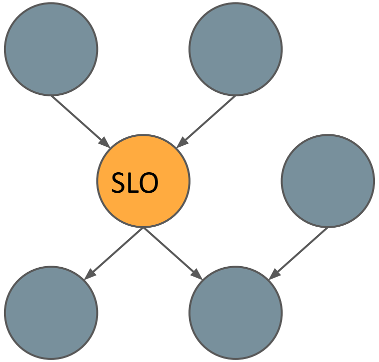


A tool for *causal* filtering.

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

Causal Inference

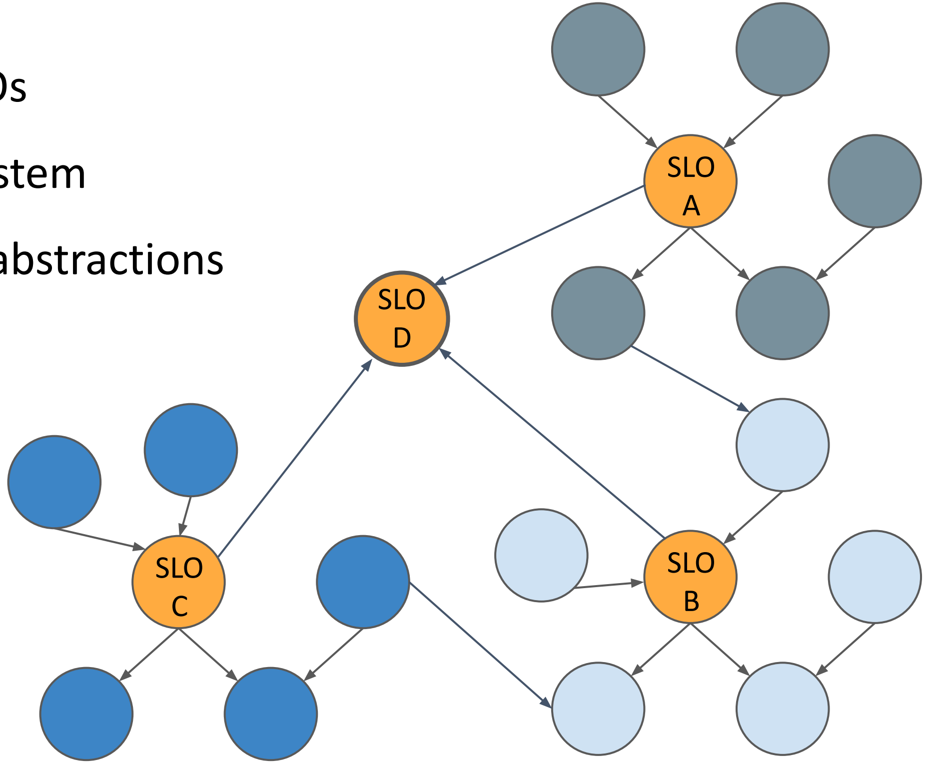
- ➢ Discover & leverage causal relationships.
- ➢ 3 Rungs on the ladder of causation. [2]
  - Observational
  - Interventional
  - Counterfactual
- ➢ Explainability capacity



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

DeepSLOs

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

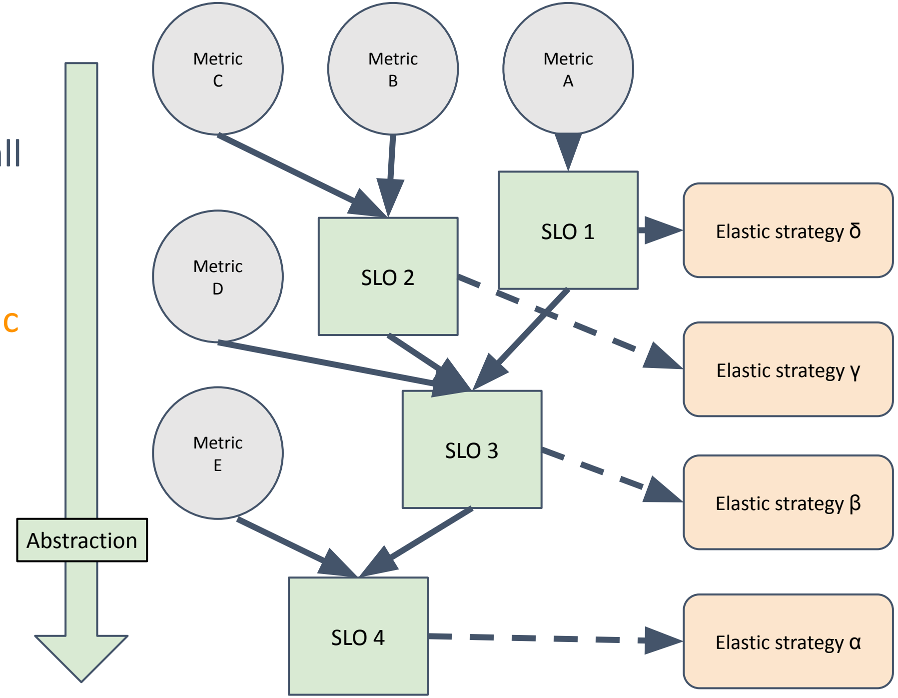


DeepSLOs

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.



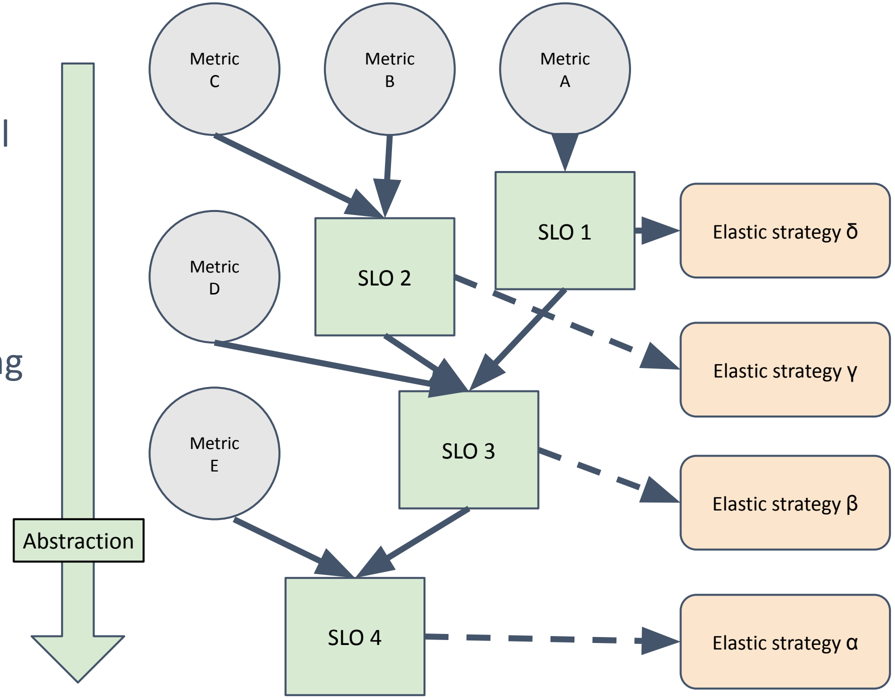
DeepSLOs

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

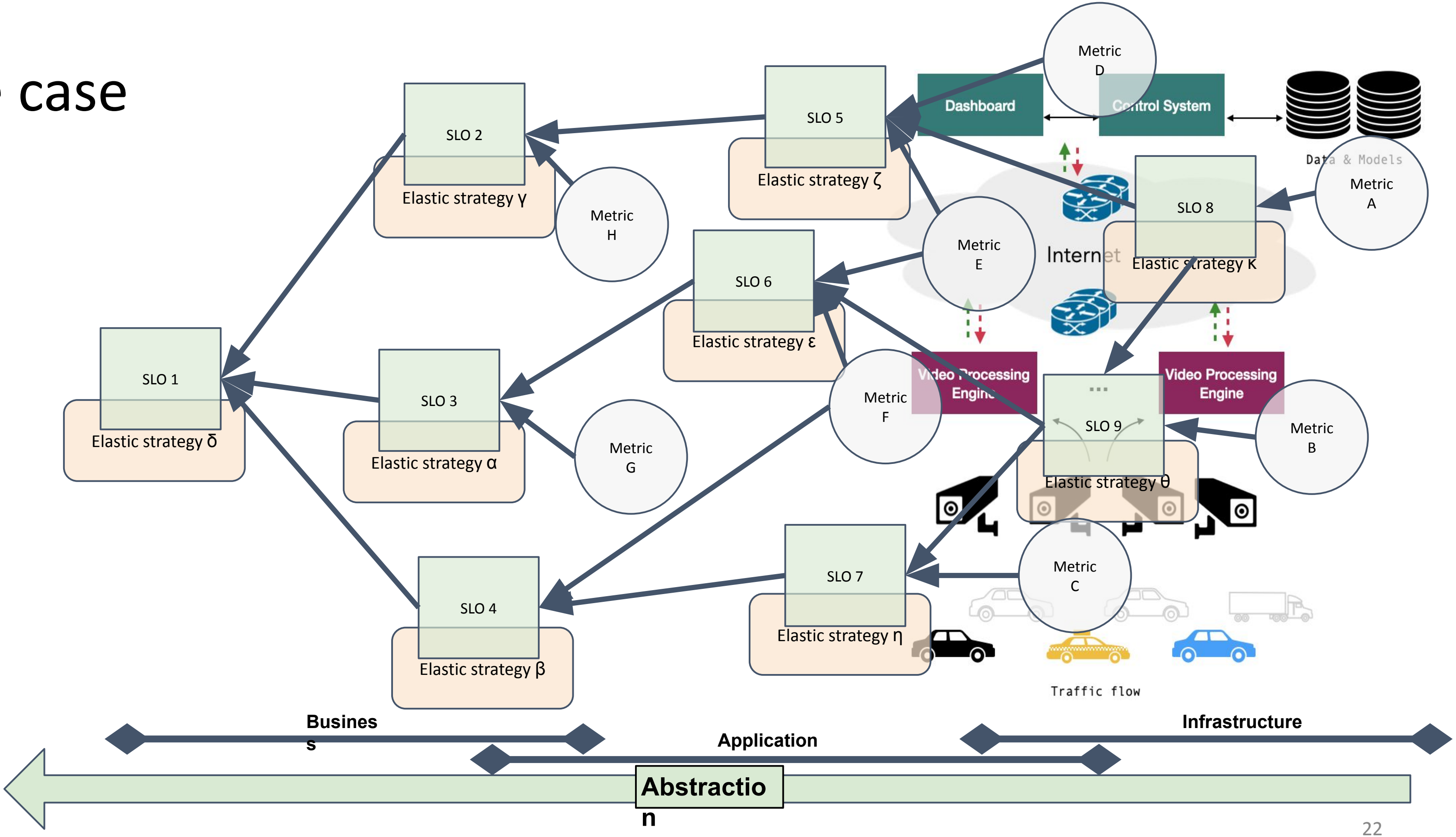
3. A complete DCCS can be mapped with several DeepSLO that connect at their highest level, allowing each DeepSLO to properly propagate towards the infrastructure the shared objectives.

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

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

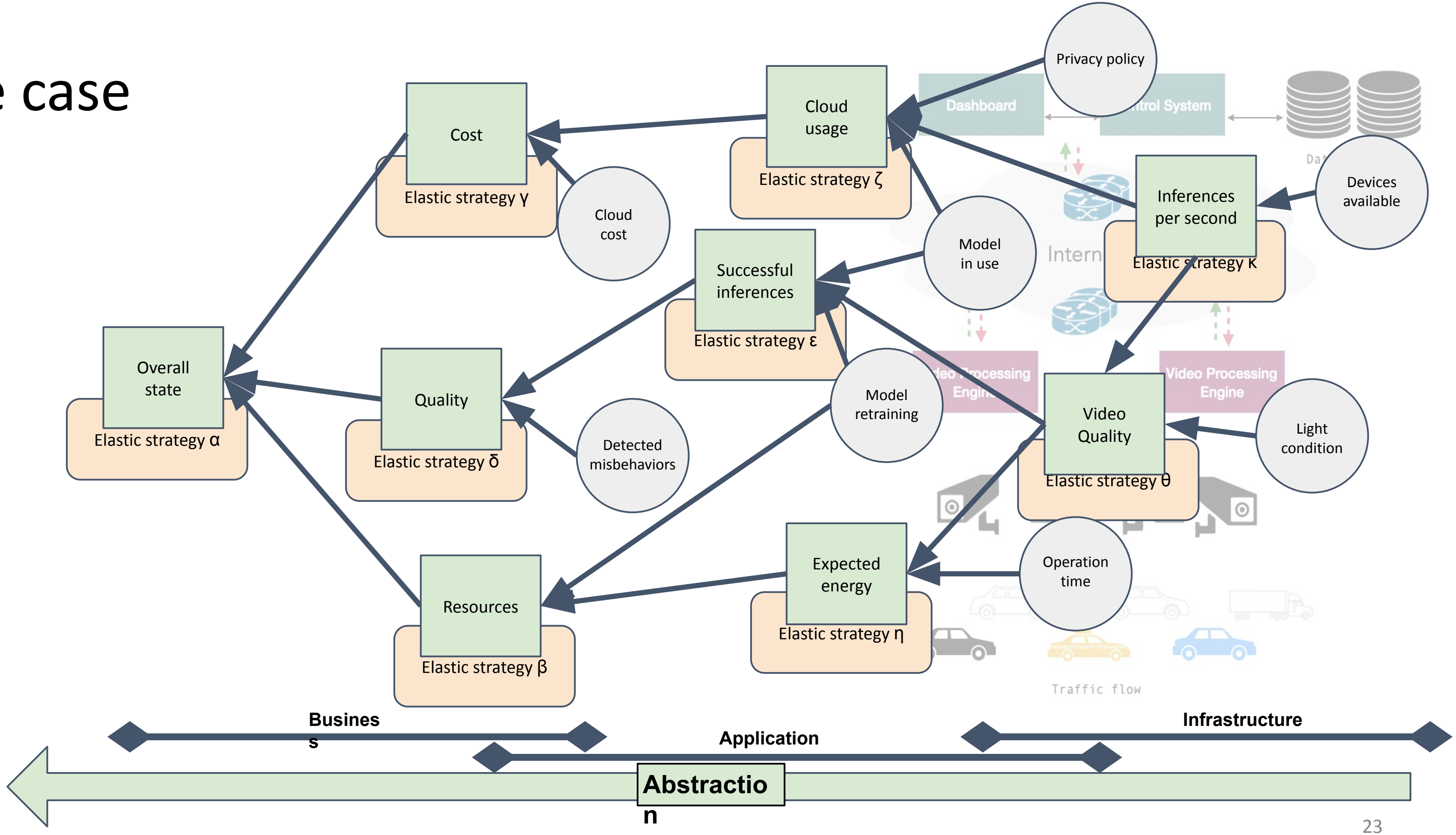


Use case



SLO 1
Elastic strategy δ
SLO 2
Elastic strategy γ
SLO 3
Elastic strategy α
SLO 4
Elastic strategy β
SLO 5
Elastic strategy ζ
SLO 6
Elastic strategy ε
SLO 7
Elastic strategy η
SLO 8
Elastic strategy κ
SLO 9
Elastic strategy θ
Metric A
Metric B
Metric C
Metric D
Metric E
Metric F
Metric G
Metric H
Dashboard
Control System
Data & Models
Internet
Video Processing Engine
Video Processing Engine
Traffic flow
Business
Application
Infrastructure
Abstraction

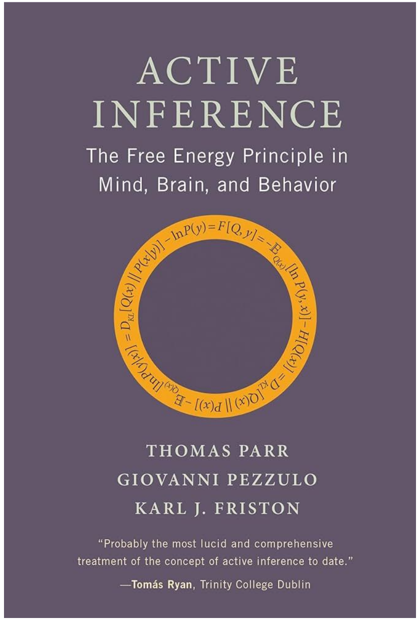
Use case



Overall state
Elastic strategy α
Cost
Elastic strategy γ
Cloud cost
Cloud usage
Elastic strategy ζ
Privacy policy
Dashboard
ntrol System
Dat
Inferences per second
Elastic strategy κ
Devices available
Model in use
Intern
Successful inferences
Elastic strategy ε
Quality
Elastic strategy δ
Detected misbehaviors
Model retraining
Video Processing Engine
Video Quality
Elastic strategy θ
Light condition
Resources
Elastic strategy β
Expected energy
Elastic strategy η
Operation time
Traffic flow
Business
Application
Infrastructure
Abstraction

Approach towards AIF

- Exchange opinions to advance PhD
- Main resources for Active Inference [1-5]
- **Verses whitepaper** [1] as a key vision
- Active Inference for **intelligent** systems



[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

Preliminary Work

Goal: Explain that the CC paper builds upon the two papers we wrote before, where we apply similar principles. This is the fusion of all that.

- 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

{b.sedlak,v.casamayor,pdonta,dustdar}@dsg.tuwien.ac.at

Active Inference on the Edge: A Design Study*

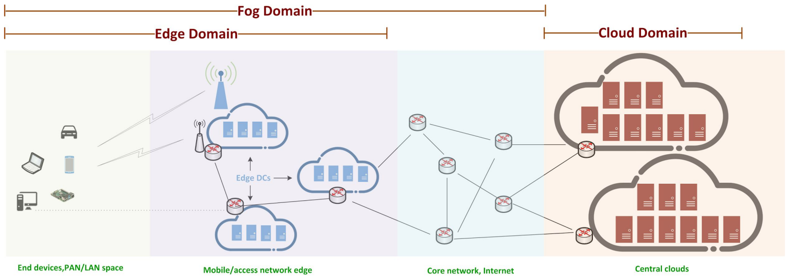
Boris Sedlak, Victor Casamayor Pujol, Praveen Kumar Donta, and Schahram Dustdar

[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

Paper Introduction

- 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

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

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 broken down 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%.

Keywords: Active Inference, Computing Continuum, Scalability, Edge Intelligence, Transfer Learning, Equilibrium

1. Introduction

Computing Continuum (CC) systems as envisioned in [1, 2, 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 Level Objectives (SLOs), may thus be broken down into smaller parts that are ensured by the respective components. To contribute to high-level goals, each device optimizes its service according to its scope. This allows SLOs to span the entire CC, also called Deep SLOs [4]. While it is one challenge to segregate and disseminate SLOs, ensuring them is another. Requirements are versatile and may change over time, every component must itself discover how its SLOs are related to its actions. For this to happen, the device could use Machine Learning (ML) techniques to discover causal relations between its environment and SLO fulfillment [5]. This promotes the usage of Active Inference (ACI) [6], a concept from neuroscience that describes how the brain continuously predicts and evaluates sensory information to model real-world processes. Given these causal models, components could adjust their environment according to preferences (i.e., SLOs).

Ensuring SLOs autonomously (i.e., evaluating the environment to infer adaptations) makes components intelligent [7]; any system (or subsystem) 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 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 by a fog node; whenever the Edge is scaled up with new devices (or device types), existing SLO-compliance models can be exchanged within the cluster. While each tier has its own SLOs, their tools for adaptation can have a different scale, e.g., fog nodes would be able to shift computations within clusters, from devices that fail their SLOs. Such operations can consider environmental impacts (e.g., network issues), but also heterogeneous device characteristics.

To realize this vision, we present our framework for collaborative edge intelligence. Guided by ACI, individual edge devices gradually develop a causal understanding of how to ensure their SLO. This knowledge is federated through a device cluster; edge devices of arbitrary types reuse existing models to ensure their SLOs. Thus, the entire Edge becomes spanned

*Corresponding author

Email addresses: boris.sedlak@dsg.tuwien.ac.at (Boris Sedlak), v.casamayor@dsg.tuwien.ac.at (Victor Casamayor Pujol), pdonta@dsg.tuwien.ac.at (Praveen Kumar Donta), dustdar@dsg.tuwien.ac.at (Schahram Dustdar)

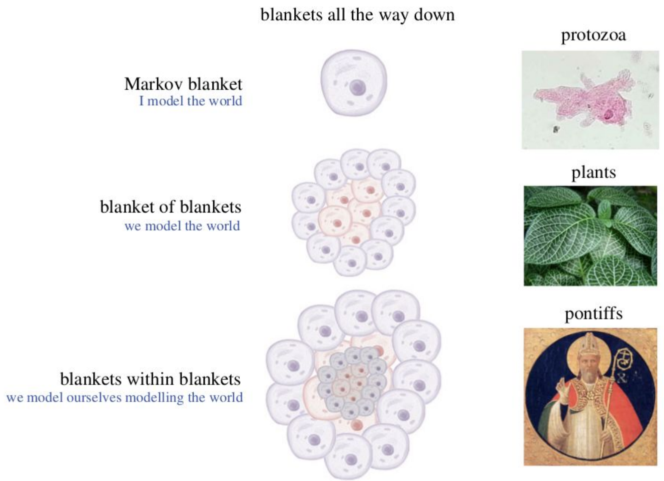
Preprint submitted to Future Generation Computer Systems *December 28, 2023*

[8] Equilibrium in the Computing Continuum through Active Inference, pending at FGCS, https://doi.org/10.48550/arXiv.2311.16769

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]

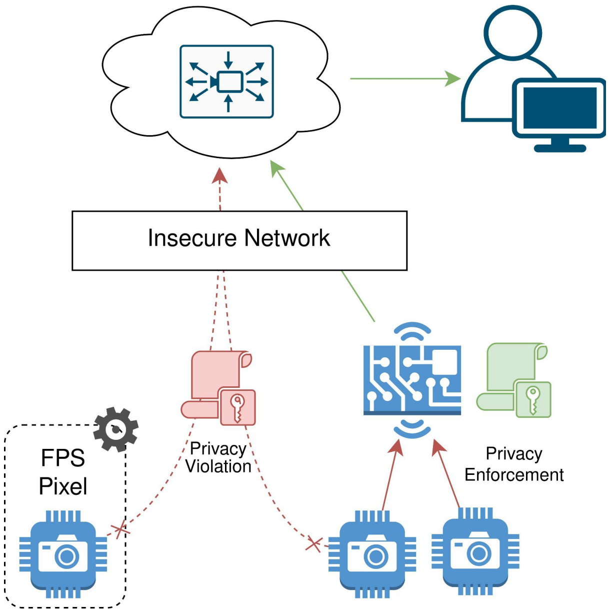
[3] Kirchhoff et al., The Markov blankets of life: autonomy, active inference and the FEP, https://doi.org/10.1098/rsif.2017.0792

Running Example

- Reflected in most of the architecture

- Use Case
  Distributed video processing architecture where IoT streams are transformed on **edge devices** to preserve individual’s privacy. After privacy enforcement, distribute streams over **cloud**.

- Hierarchical network structure
  IoT devices provide streams to edge devices; streams processed locally at edge devices; video stream properties are **configurable**



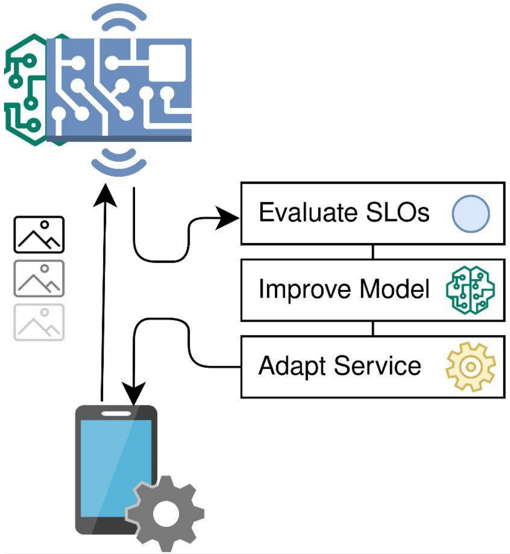
Collaborative Edge Intelligence Framework

3 major contributions in interplay:

Collaborative Edge Intelligence Framework

3 major contributions in interplay:

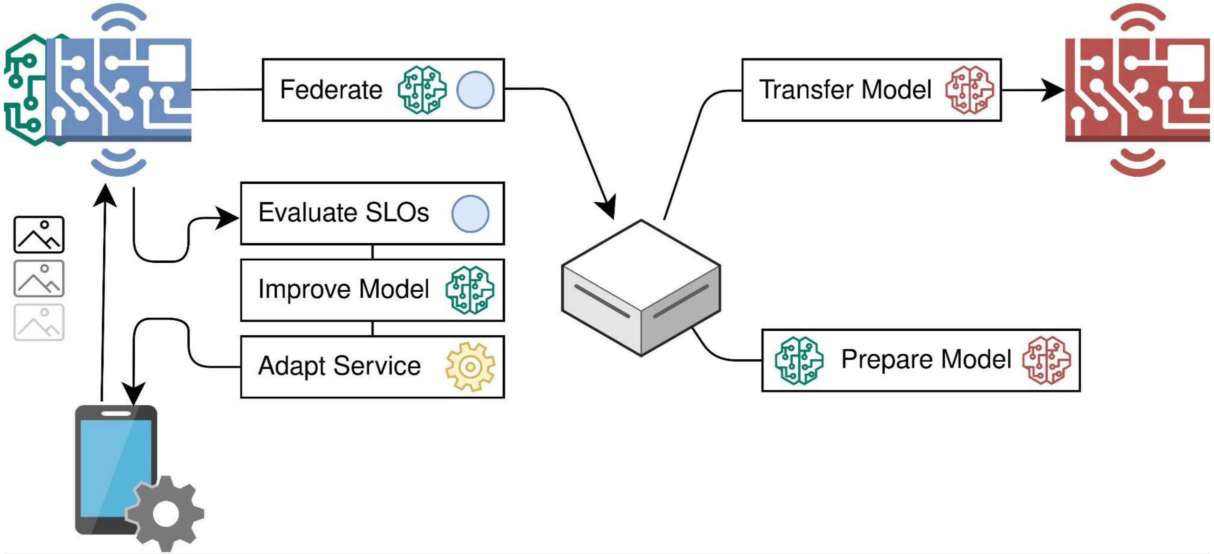
1. Continuous model accuracy and local SLO fulfillment



Collaborative Edge Intelligence Framework

3 major contributions in interplay:

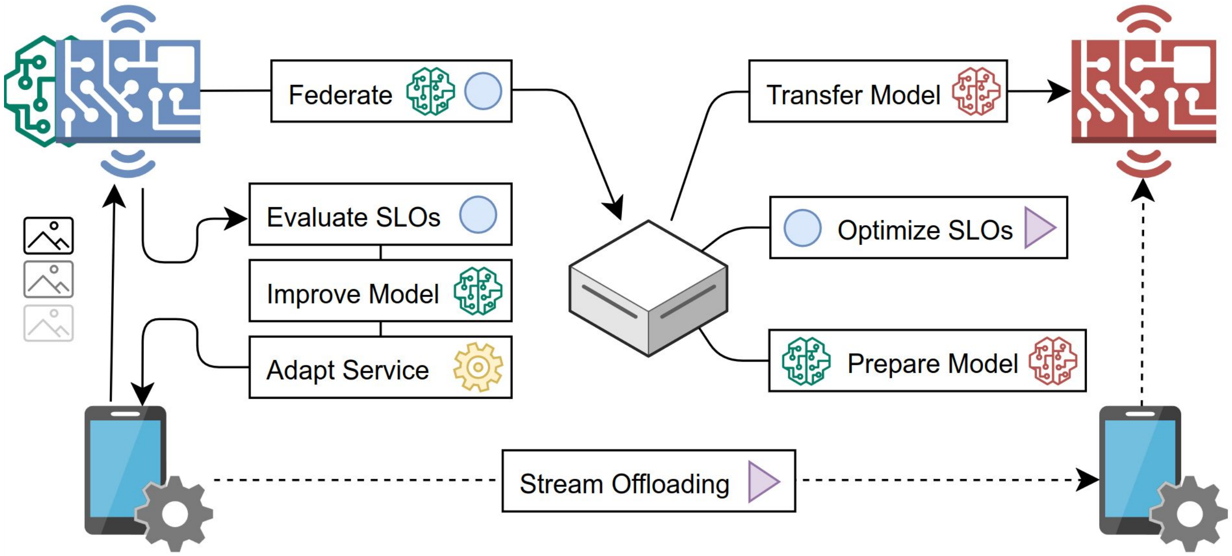
1. Continuous model accuracy and local SLO fulfillment
2. Federation and combination of models



Collaborative Edge Intelligence Framework

3 major contributions in interplay:

1. Continuous model accuracy and local SLO fulfillment
2. Federation and combination of models
3. Collaboration between cellular structures

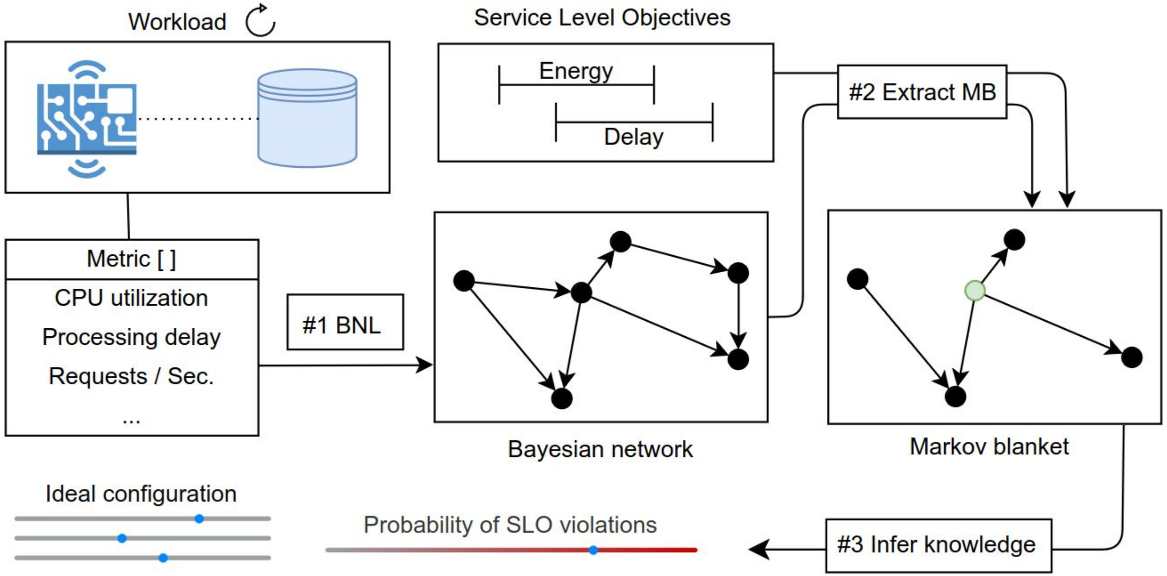


Contribution Structure

1. Continuous model accuracy and local SLO fulfillment
   a. Static BNL and Inference
   b. Continuous BNL and Inference (**AIF**)
2. Federation and combination of models
3. Collaboration between cellular structures

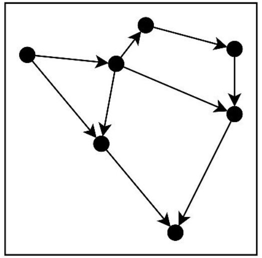
1a – Static BNL and Inference

- Basic mechanism for assuring SLOs at individual devices
- **Requires training data in upfront and is prone to data shifts**
- Evaluates possible configurations through a 3-step method



1a – Static BNL and Inference (2)

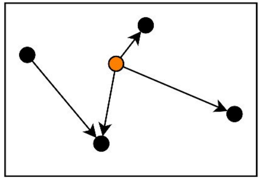
Bayesian Network Learning (BNL)



Bayesian network

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

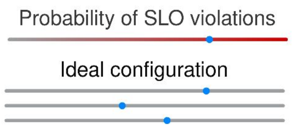
Markov Blanket (MB) Selection



Markov blanket

- ❑ Causality filter [1,4]
- ❑ Identify variables that have an impact on **SLO fulfillment**

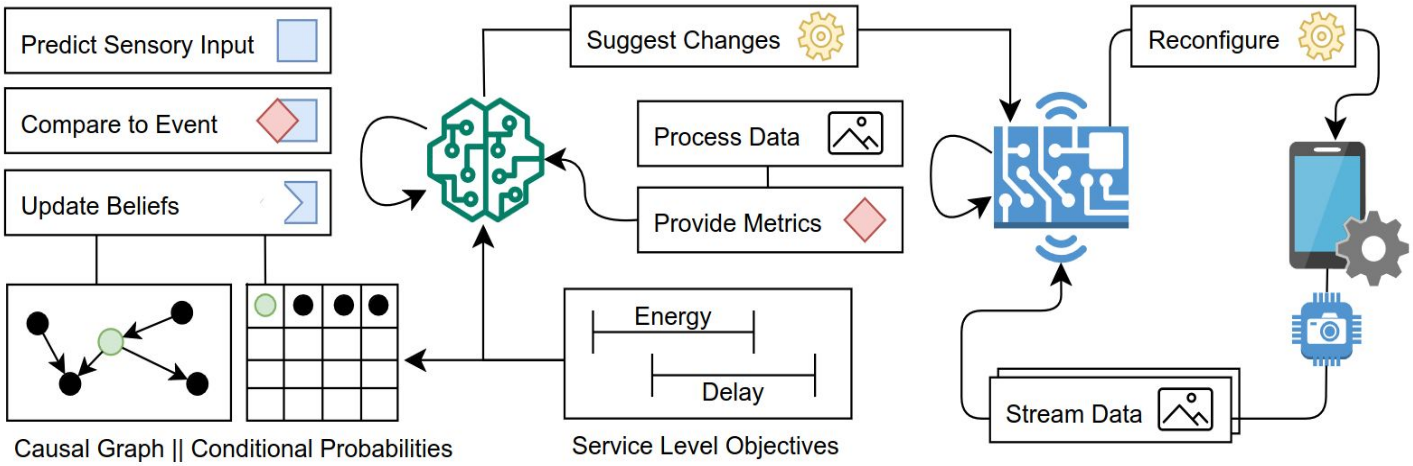
Knowledge Extraction



- ❑ P(SLO < x) for all variable combinations
- ❑ Find **Bayes-optimal** system configuration

1b – Continuous BNL and Inference

- **AIF agent** → Equilibrium-Oriented SLO Compliance (**EOSC**) model
- Agent uses SLOs as **preferences** during continuous adaptation
- BN trained incrementally from incoming observations
- Beliefs updated according to prediction errors



1b – AIF Agent Behaviour

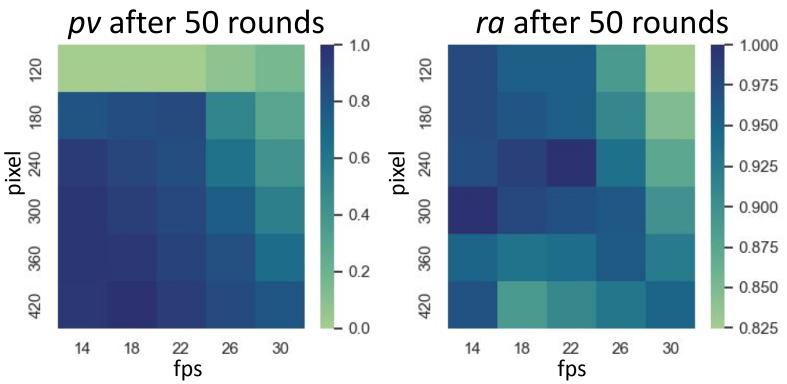
Determined by three factors:

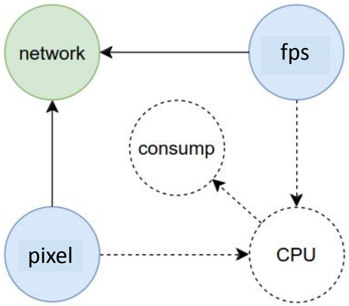
- Pragmatic value (*pv*)
  Summarizes **QoE** SLOs (e.g., resolution)
- Risk assigned (*ra*)
  Summarizes **QoS** SLOs (e.g., network limit)

pv & *ra* calculated as **separate factors** from MBs; configurations rated according to SLO fulfillment; **interpolation** between known configurations

- Information gain (*ig*)
  Continued on the next slide

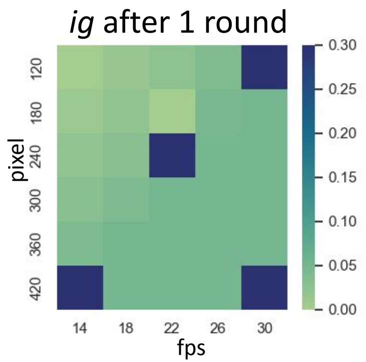
$$u_c = pv_c + ra_c + ig_c$$





1b – AIF Agent Behaviour (2)

- Information gain (*ig)*
  - Favors configurations that promise **model improvement**
  - Summarizes surprise for observations included in the **MB**
  - Hyperparameter (*e*) allows exploring designated areas

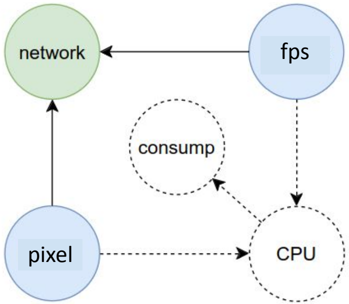


AIF agent cycle:

1. Calculate **surprise** for current batch of observations
2. Retrain structure (or parameters) depending on surprise
3. Calculate behavioral factor for **empirically evaluated** configs
4. **Interpolate** between known configurations in 2D (or 3D) space
5. Choose the highest-scoring (device) configuration

Agent gradually develops **understanding** how to ensure SLOs

$$ig(c) = e + \left(\frac{\tilde{\mathfrak{S}}_c}{\bar{\mathfrak{S}}}\right) \times 100$$



2 – Knowledge Exchange

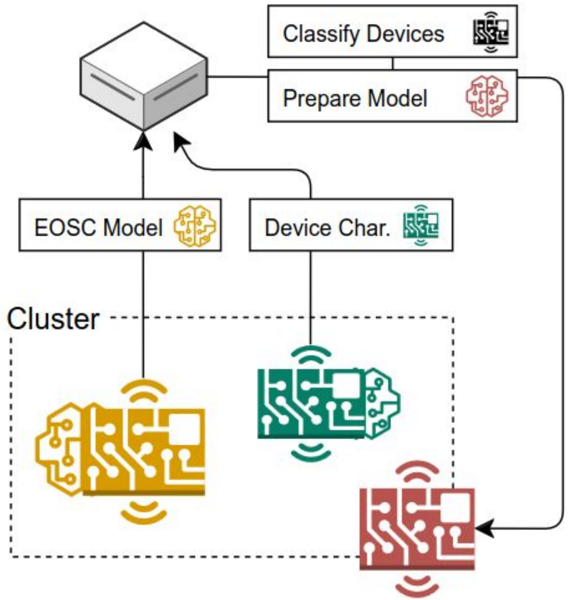
Extend from single devices to the CC

Heterogeneity among the Edge

- Impedes simple transfer learning of models
- Low model accuracy ⟶ high surprise

- Requires a **cluster leader** (fog node or edge)
- EOSC models collected at a leader node
- Model selection according to hardware char.
- Merging models to provide tailor-fit one

Fast onboarding (= horizontal scaling) of devices

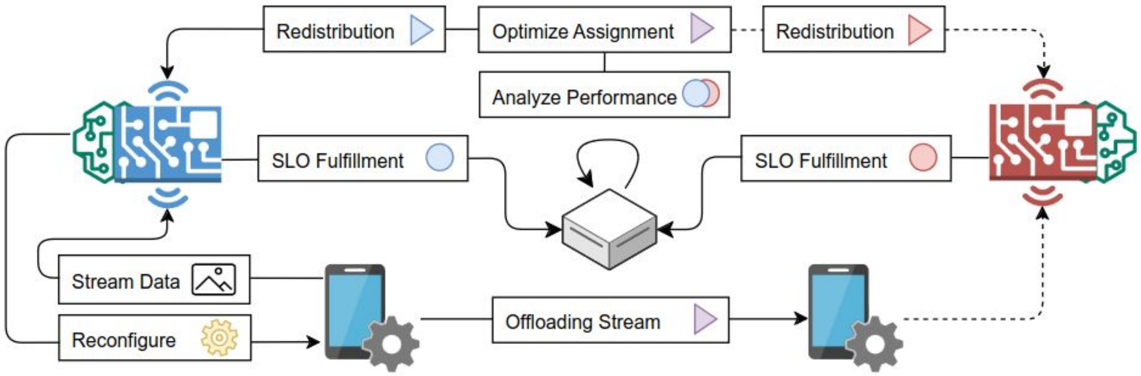


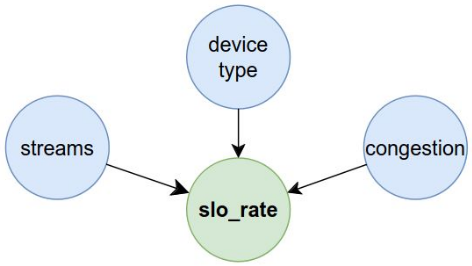
3 – Collaborative Scaling

Limited action scope of devices

- Individual devices restricted to local scope to resolve SLO violations

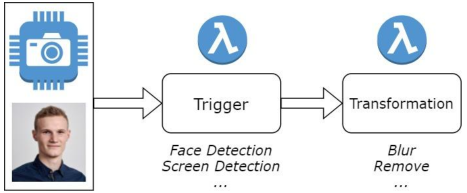
- Leader node collecting **environmental metrics** (e.g., network congestion)
- Incorporated to causal model, contrasted against local SLO fulfillment (**AIF**)
- Emerging structures allows optimizing cluster-wide SLO fulfillment
    - E.g., redistribute clients between impacted devices





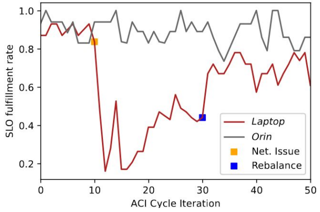
Evaluation - Overview

- Use Case
  Distributed video processing architecture where streams are transformed on **edge devices** to preserve privacy of individuals.



[5]

- Implementation
  Prototype including video transformations and the collaborative edge intelligence framework.



- Evaluation Scope
  Targeting each contribution with different aspects.

[5] Sedlak, B., Murturi, I., Donta, P.K., Dustdar, S.: A Privacy Enforcing Framework for Transforming Data Streams on the Edge. IEEE Emerging Topics in Computing (2023)

Evaluation - Use Case

BNL comprises metrics from various sources (e.g., IoT client or edge device);
Extended with target conditions (i.e., SLOs) to create the **EOSC** model:

Model training takes 11 (3) metrics

Table 1: List of metrics captured by the devices, which are turned into variables by ACI

Name	Origin	Unit	Description	Param
pixel	IoT	num	number of pixel contained in a frame	Edge
fps	IoT	num	number of frames received per second	Edge
bitrate	IoT	num	number of pixels transferred per second	No
cpu	Edge	%	utilization of the device CPU	No
memory	Edge	%	utilization of the system memory	No
streams	Edge	num	number of IoT devices providing data	Fog
consumption	Edge	W	energy pulled by the device	No
network	Edge	num	data transferred over network interface	No
delay	App.	ms	processing time per video frame	No
success	App.	T/F	if a pattern (i.e., face) was detected	No
distance	App.	num	relative object distance between frames	No
slo_rate	Edge	%	combined SLO Fulfillment rate ($pv \times ra$)	No
device_type	Edge	enum	physical device type	No
congestion	Edge	num	network congestion that increases latency	No

SLOs from model variables

Table 2: Extracted SLOs and their classification.

SLO	Condition	Tier	Type
network	$network < 1.6$ MB/s	Edge	QoS
in_time	$delay < 1/fps$	Edge	QoS
success	$success = True$	Edge	QoE
distance	$distance < 50$	Edge	QoE
slo_rate	$\max(slo\_rate)$	Fog	Both

Parameters allow **configuring** a component's environment

Evaluation - Implementation

Python prototype for which we provide:

- Github repository
- Docker container

https://www.nvidia.com/en-sg/autonomous-machines/embedded-systems/jetson-xavier-nx/

Evaluation included a variety of edge devices:

Table 3: List of devices used for implementing and evaluating the presented methodology

Full Device Name	ID	Price[6]	CPU	RAM	GPU	p [1,4]	g [0,2]	Σ
ThinkPad X1 Gen 10	*Laptop*	1800 €	Intel i7-1260P (16 core)	32 GB	————	Very High (4)	None (0)	4
Nvidia Jetson Orin	*Orin*	500 €	ARM Cortex A78 (6 core)	8 GB	Volta (383 core)	High (3)	High (2)	5
Nvidia Jetson Nano	*Nano*	150 €	ARM Cortex A57 (4 core)	4 GB	————	Low (1)	None (0)	1
Nvidia Jetson Xavier	*Xavier*	300 €	ARM Carmel v8.2 (6 core)	8 GB	————	Medium (2)	None (0)	2
Jetson NX GPU	*NX*	300 €	ARM Carmel v8.2 (6 core)	8 GB	Amp (1024 core)	Medium (2)	Low (1)	3

Devices combined within a cluster and classified relatively to each other

Evaluation - Aspects

We motivated, evaluated, and provided the results for 13 aspects:

A-1: Do MBs reduce the complexity of inference?
A-2: What is AIF's operational overhead?
A-3: How long require AIF agents to ensure SLOs?
A-4-1: Are the produced Bayesian networks interpretable?
A-4-2: Is the behavior of AIF agents explainable?
A-5: What is the operational impact of including BNL in the AIF cycle?
A-6: Can changes in variable distribution be handled?
A-7: Can SLOs be modified during runtime?

K-1: What is the SLO fulfillment rate of transferred models?
K-2: Can knowledge transfer achieve any speedup?
K-3: Do tailored models have lover surprise compared to existing models?

S-1: How is the load distributed among resource-constrained devices?
S-2: Can intelligent CC structures optimize local SLO fulfillment?

Evaluation - Aspects (Filtered)

We motivated, evaluated, and provided the results for 13 aspects:

A-1: Do MBs reduce the complexity of inference?
A-2: What is AIF's operational overhead?
A-3: How long require AIF agents to ensure SLOs?
A-4-1: Are the produced Bayesian networks interpretable?
A-4-2: Is the behavior of AIF agents explainable?
A-5: What is the operational impact of including BNL in the AIF cycle?
A-6: Can changes in variable distribution be handled?
A-7: Can SLOs be modified during runtime?

K-1: What is the SLO fulfillment rate of transferred models?
K-2: Can knowledge transfer achieve any speedup?
K-3: Do tailored models have lover surprise compared to existing models?

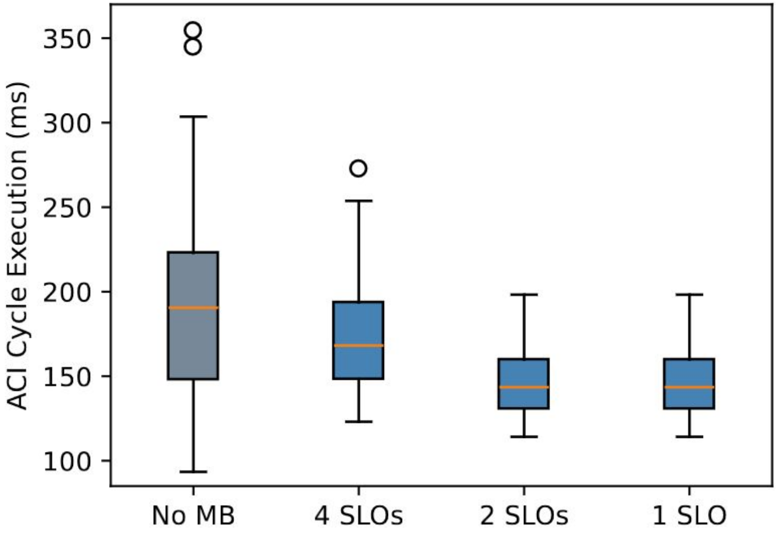
S-1: How is the load distributed among resource-constrained devices?
S-2: Can intelligent CC structures optimize local SLO fulfillment?

A-1: Do MBs reduce the complexity of inference?

- Setup
  Modify the AIF agent to calculate behavior factors (i.e., **surprise,** etc) for a reduced number of SLOs with or without MB

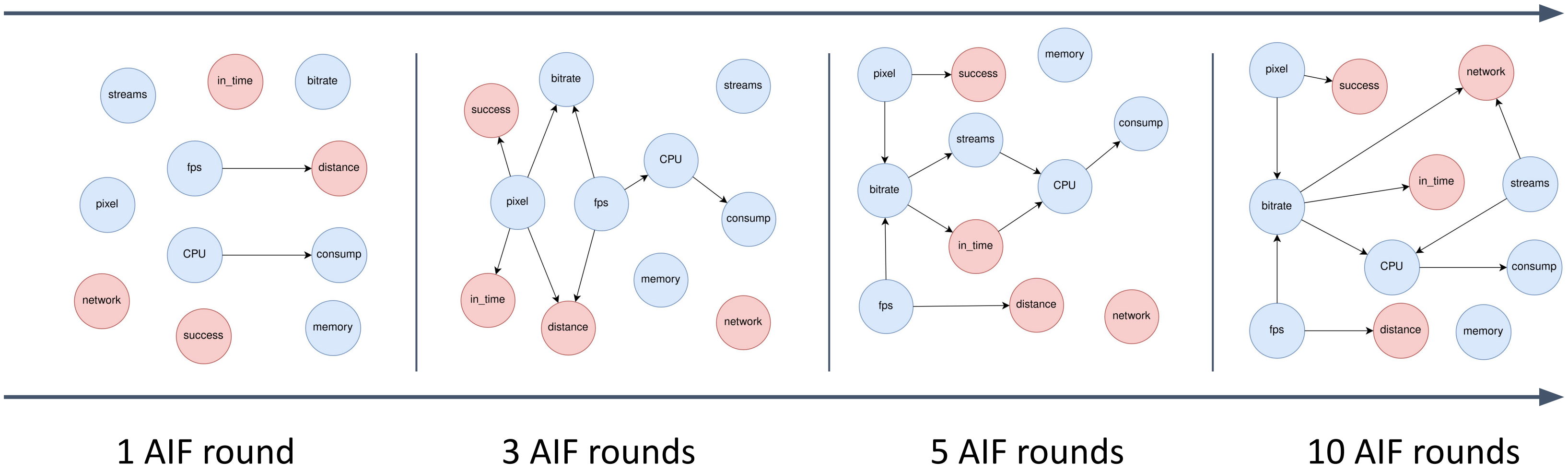
- Result
  Applying MBs reduced the median inference time of 4 SLOs from 197ms to 151ms

- Implication
  MB provided a decreased **system view**



A-4-1: Are the produced Bayesian networks interpretable?

- **Setup**
  Train the EOSC model from scratch and extract the BN after X rounds
- **Result**
  Dependencies **gradually** revealed:
- **Implication**
  AIF can be used to identify **causal relations** according to current and upcoming observations. Results are intuitively comprehensible.

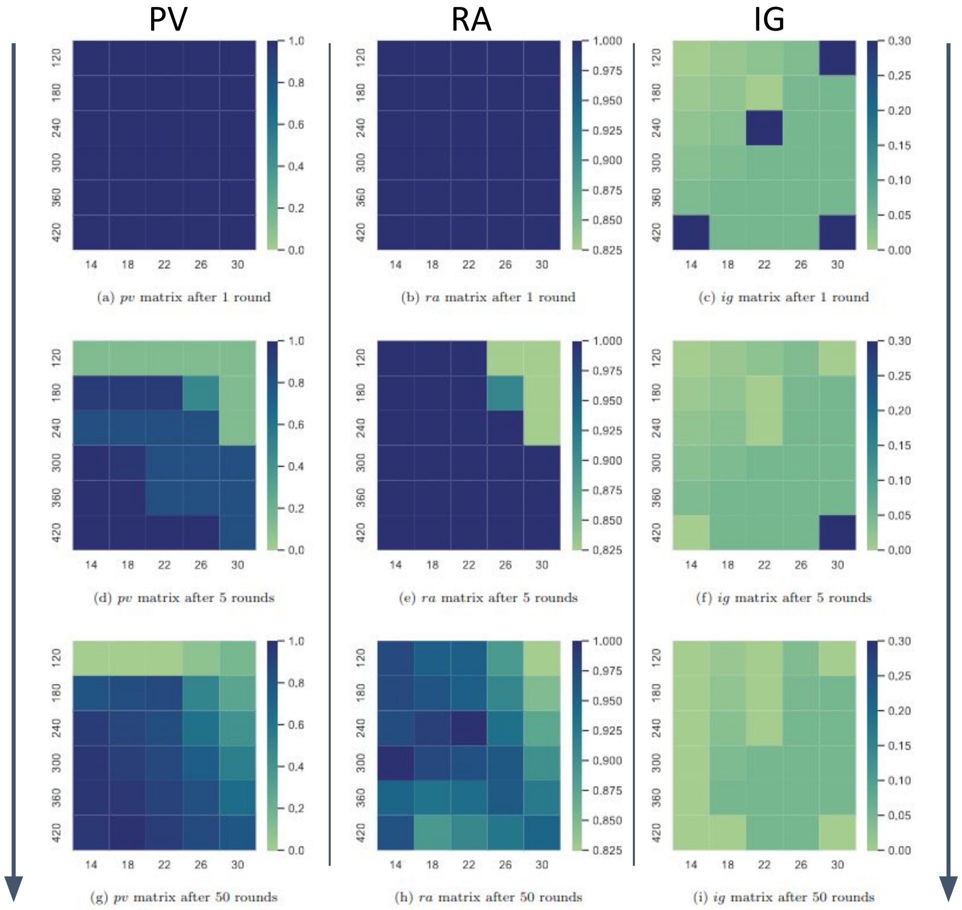


A-4-2: Is the behavior of AIF agents explainable?

- Setup
  Train the EOSC model from scratch and extract the agent’s behavioral factors after X rounds

- Result
  Develops clear preferences

- Implication
  Allows to **empirically debug** the behavior and **fine-tune** agent by adjusting hyperparameters

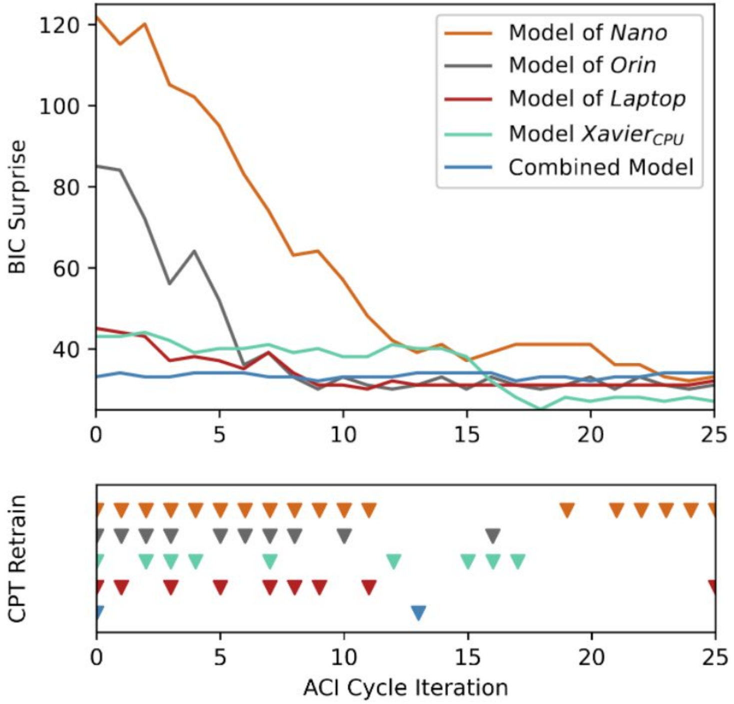


K-3: Do tailored models have lover surprise compared to existing models?

- Setup
  Federate EOSC models within the cluster, select and ***combine*** models for joining edge device; track retraining.

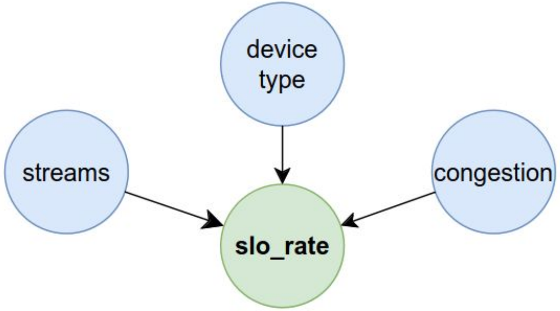
- Result
  Tailor-made model reported the lowest **surprise**, although remaining models improved through **retraining.**

- Implication
  Surprise can be decreased by choosing a (best-)fitting device model .

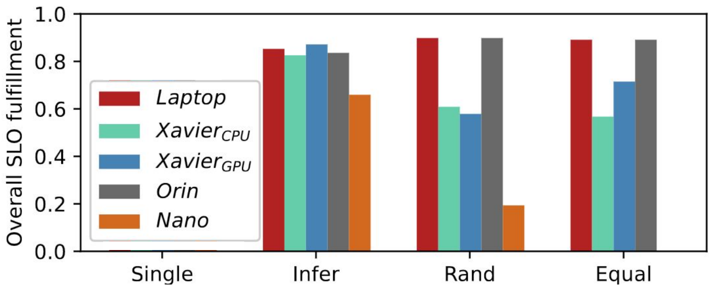


S-1: How is load distributed among resource-constrained devices?

- Setup
  Cluster-wide EOSC model that describes **SLO fulfillment** depending on *device types* and the number of processed *streams*. **Infers** optimal client assignment.



- Result
  Cluster-wide SLO fulfillment was improved from 0.60 (*E or R*) to 0.81 (*I*)



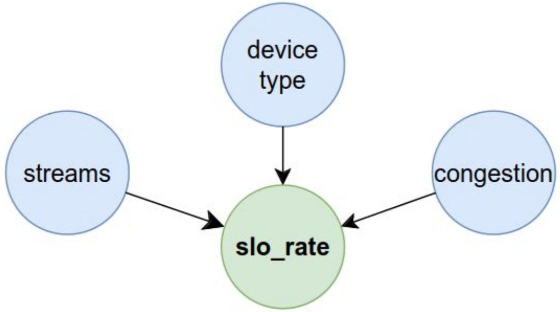
- Implication
  Leader node considered environmental factors to optimize a target variable (i.e., SLOs).

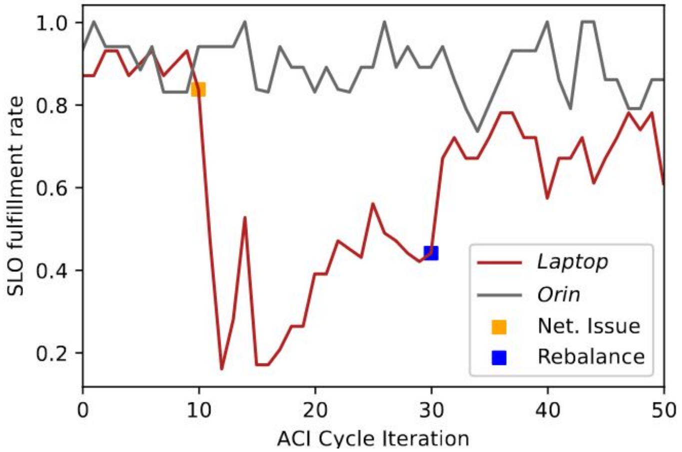
S-2: Can intelligent CC structures optimize local SLO fulfillment?

- Setup
  Clients distributed equally between **comparable** devices, introducing network *congestion* for one of them; rebalance load.

- Result
  Cluster-wide SLO fulfillment (Σ) improved from 1.03 to 1.53.

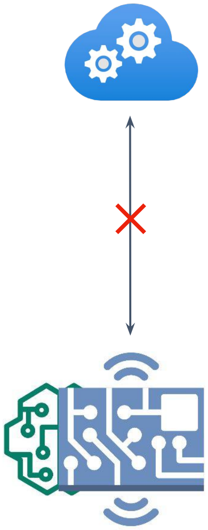
- Implication
  Was able to **raise the scope** of elasticity strategies, but requires sufficient data to model the relation of *congestion* → *slo_rate*.



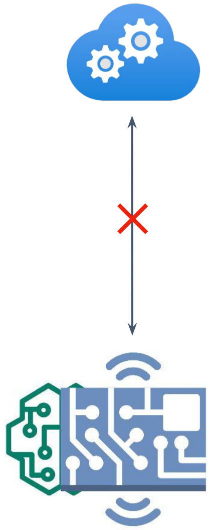


Summary

- Impossible to centrally evaluate requirements
  - Decentralize SLO fulfillment for CC components
  - Enforce requirements at the respective component



Summary



- Impossible to centrally evaluate requirements
  - Decentralize SLO fulfillment for CC components
  - Enforce requirements at the respective component

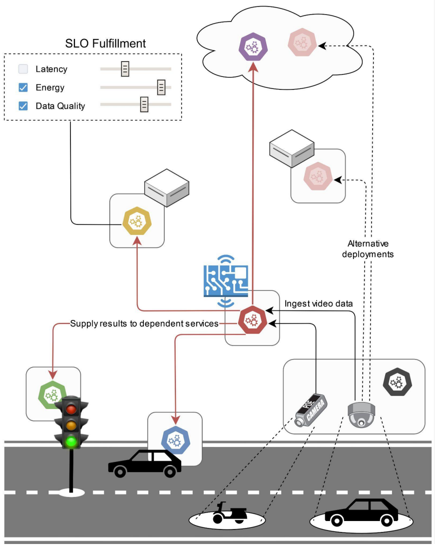
- Active Inference as key method for **self-adaptation**
  - **Autonomous** EOSC model training and updating
  - Fulfill SLOs through **continuous** reconfiguration

- Federation of models within higher-level components
  - Collaboration in the CC accelerate device onboarding
  - Assembled structures increased the **action scope**

Current Challenges and Outlook

- Pending comparison with other (ML) approaches
  - Evaluation of more complex use cases

- Composition of MBs for larger structures (**DeepSLOs**)
  - Constrain one MB depending on another's SLOs



Thankful for **feedback** and looking for potential **collaborations**