Universitat Pompeu Fabra Barcelona

From Cloud to Edge: Vision and Challenges for Service Orchestration

Boris Sedlak

Co-funded by the European Union

Funded by the European Union NextGenerationEU

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA Y TURISMO

Plan de Recuperación, Transformación y Resiliencia

Abstract

Hidden in main presentation

Shifting computation from Cloud to Edge promises low latency, but suffers from heterogeneous infrastructure and dynamic disruptions. This talk explores structural challenges in operating and orchestrating such systems, and shows how concepts from neuroscience (Active Inference) can optimize services through continuous adaptation. We highlight critical research gaps, including standardized benchmarking and intent-based control, that will prove essential to realize truly resilient and self-organizing services.

Who am I ??

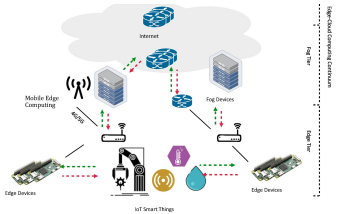
Boris Sedlak

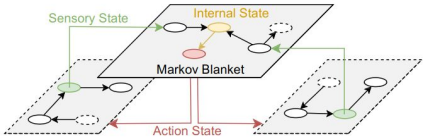
Postdoc @ UPF Barcelona
Distributed Intelligence & Systems-Engineering Lab

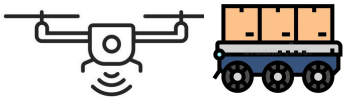
PhD @ TU Wien, Vienna
Distributed Systems Group

Software Engineer @ Lotterien
Agile Development and Testing

Structure of the Talk







Current Trends and Common Problems

Personal Methodologies and Latest Results

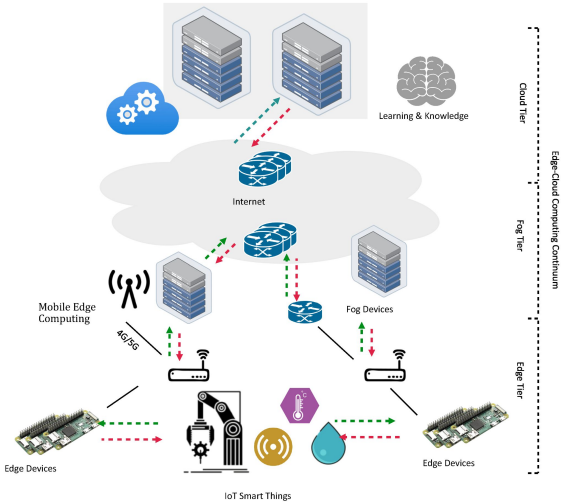
Research Directions and Challenges

I – Common Concepts

Edge Computing performs computation in close distance to the data source (e.g., an IoT device); thus, decreases latency for computations and improves privacy preservations

Computing Continuum (CC) as a composition of multiple processing tiers (e.g., Edge, Fog, and Cloud resources) that combines their individual benefits (i.e., latency / availability)

Smart Cities are a common instance of distributed systems, where interconnected services (e.g., traffic surveillance or road surveillance) collaborate based on collected sensor data



Example of a Computing Continuum architecture [1]

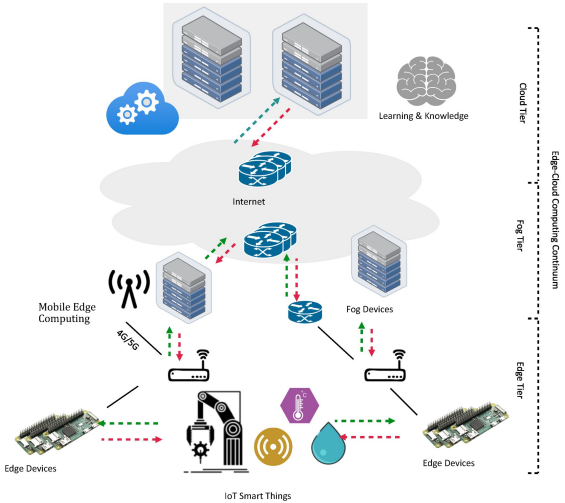
[1] P. Donta, I. Murturi, V. Casamayor, B. Sedlak, and S. Dustdar; **Exploring the Potential of Distributed Computing Continuum Systems** (2023)

I – Common Concepts

Edge Computing performs computation in close distance to the data source (e.g., an IoT device); thus, decreases latency for computations and improves privacy preservations

Computing Continuum (CC) as a composition of multiple processing tiers (e.g., Edge, Fog, and Cloud resources) that combines their individual benefits (i.e., latency / availability)

Smart Cities are a common instance of distributed systems, where interconnected services (e.g., traffic surveillance or road surveillance) collaborate based on collected sensor data



Example of a Computing Continuum architecture [1]

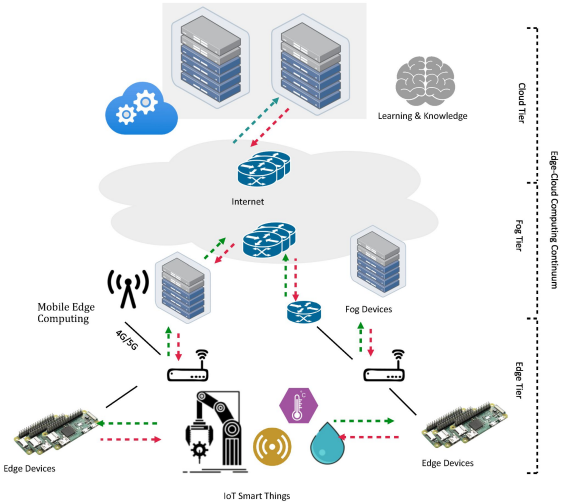
[1] P. Donta, I. Murturi, V. Casamayor, B. Sedlak, and S. Dustdar; **Exploring the Potential of Distributed Computing Continuum Systems** (2023)

I – Common Concepts

Edge Computing performs computation in close distance to the data source (e.g., an IoT device); thus, decreases latency for computations and improves privacy preservations

Computing Continuum (CC) as a composition of multiple processing tiers (e.g., Edge, Fog, and Cloud resources) that combines their individual benefits (i.e., latency / availability)

Smart Cities are a common instance of distributed systems, where interconnected services (e.g., traffic surveillance or road surveillance) collaborate based on collected sensor data



Example of a Computing Continuum architecture [1]

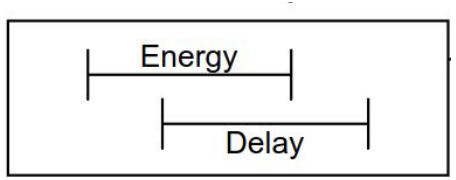
[1] P. Donta, I. Murturi, V. Casamayor, B. Sedlak, and S. Dustdar; **Exploring the Potential of Distributed Computing Continuum Systems** (2023)

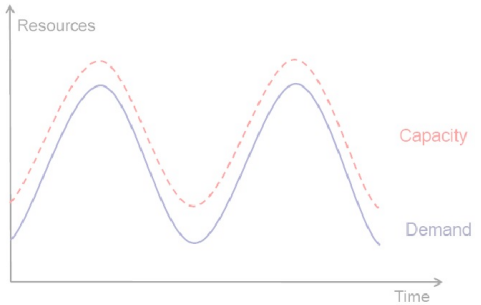
I – Common Concepts (cont.)

Service Level Objectives (SLOs) specify requirements that must be ensured throughout operation (e.g., latency < *t*).

Elasticity Strategies scale a system according to current demand; e.g., if performance is insufficient, allocate more resources. However, what if there are no more resources?

Service Level Agreements (SLAs) as a flexible contract (€) between service provider and consumer.





Elasticity allocates the right amount of resources [2]

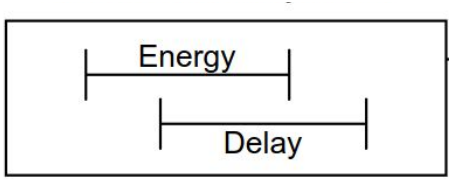
[2] Ricciardi et al., **Saving Energy in Data Center Infrastructures** (2011)

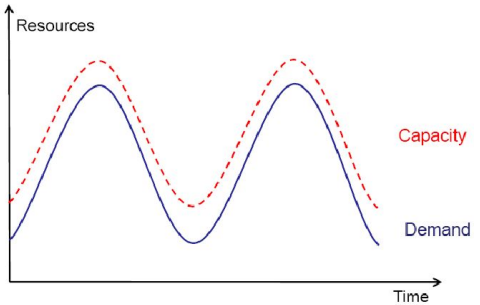
I – Common Concepts (cont.)

Service Level Objectives (SLOs) specify requirements that must be ensured throughout operation (e.g., latency < *t*).

Elasticity Strategies scale a system according to current demand; e.g., if performance is insufficient, allocate more resources. However, what if there are no more resources?

Service Level Agreements (SLAs) as a flexible contract (€) between service provider and consumer.





Elasticity allocates the right amount of resources [2]

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

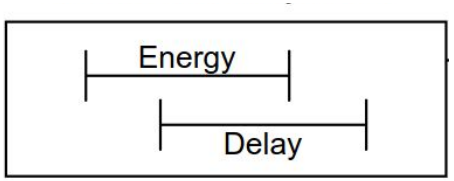
I – Common Concepts (cont.)

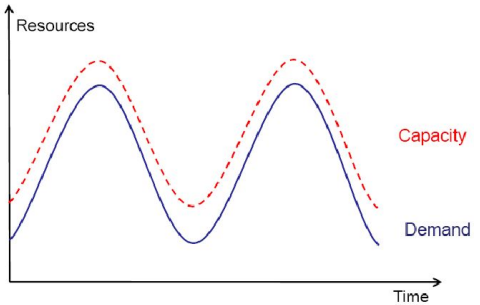
Hidden in main presentation

Service Level Objectives (SLOs) specify requirements that must be ensured throughout operation (e.g., latency < *t*).

Elasticity Strategies scale a system according to current demand; e.g., if performance is insufficient, allocate more resources. However, what if there are no more resources?

Service Level Agreements (SLAs) as a flexible contract (€) between service provider and consumer.

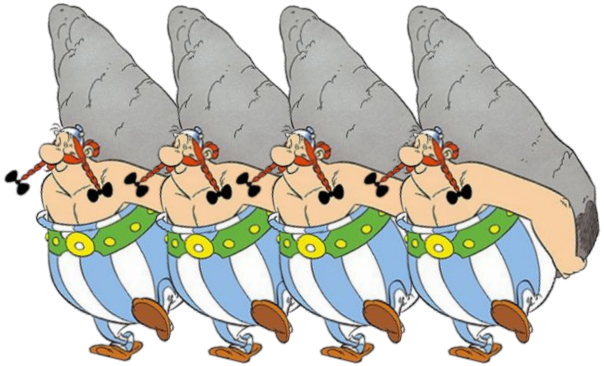




Elasticity allocates the right amount of resources [2]

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

I – Fundamental Problems



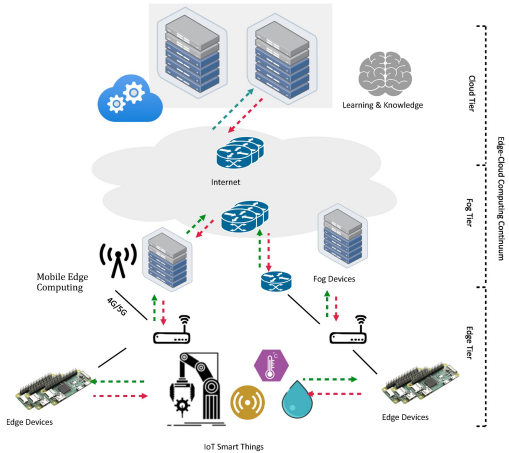
Homogeneous Resources (Cloud)

Heterogeneous Resources (CC)

Free images from https://www.nicepng.com/ourpic/u2e6r5y3e6y3t4i1_dogmatix-obelix-asterix-obelix-asterix-a-obelix-png/

I – Fundamental Problems

Homogeneous Resources (Cloud)

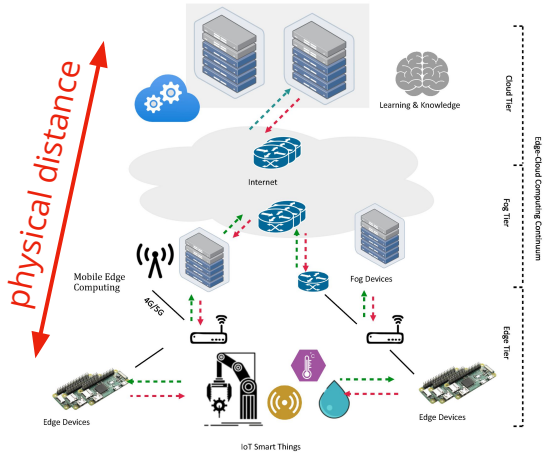


Heterogeneous Resources (CC)

Free images from https://www.freepik.com/free-photos-vectors/server-rack

I – Fundamental Problems

Homogeneous Resources (Cloud)

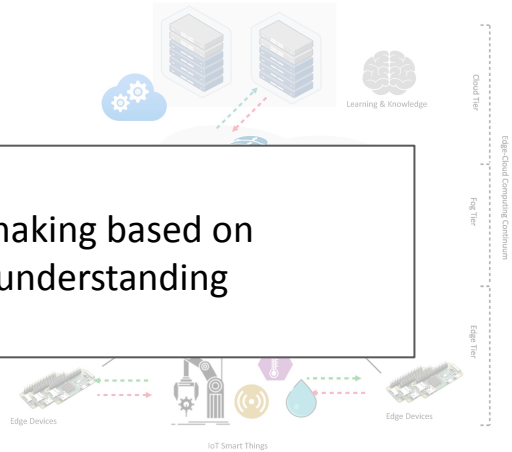


Heterogeneous Resources (CC)

Free images from https://www.freepik.com/free-photos-vectors/server-rack

I – Fundamental Problems

Decentralized decision-making based on local observations and understanding



Homogeneous Resources (Cloud)

Heterogeneous Resources (CC)

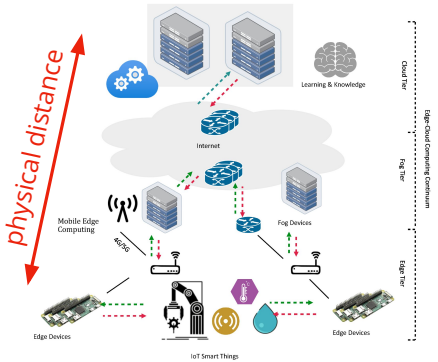
Free images from https://www.freepik.com/free-photos-vectors/server-rack

I – Fundamental Problems

Hidden in main presentation

Heterogeneity prevents simple task distribution between available devices. Don't know a priori what will be the *behavior* of service X on device Y

Physical distribution creates latency between different distributed devices; transferring state information (e.g., CPU load) creates lots of traffic on the network.



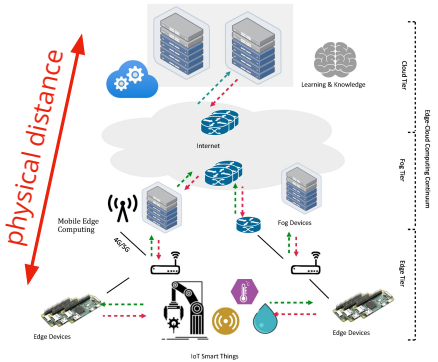
I – Fundamental Problems

Hidden in main presentation

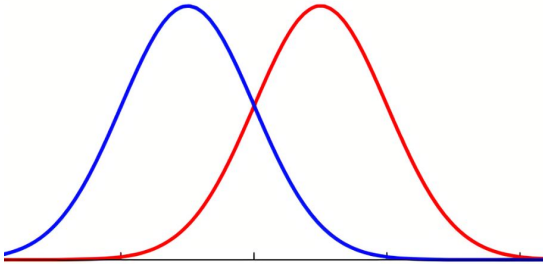
Heterogeneity prevents simple task distribution
between available devices. Don't know a priori what
will be

Physi
distrib
CPU load) creates lots of traffic on the network.

Decentralized decision-making based on local observations and understanding



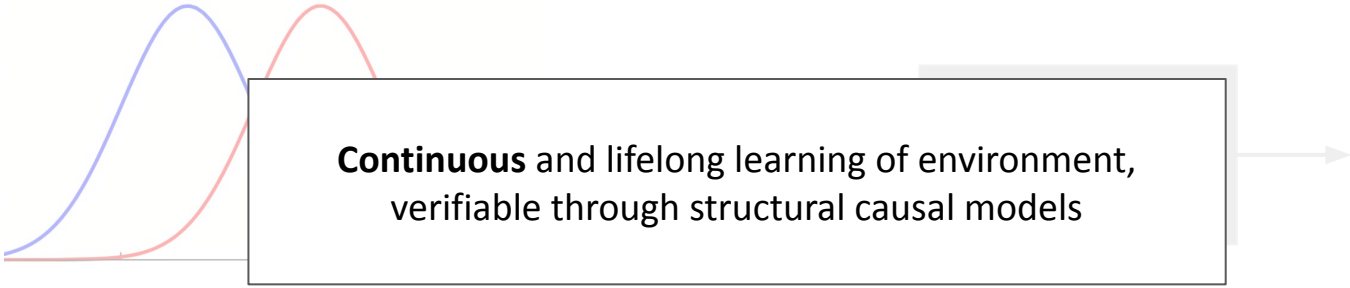
I – Fundamental Problems (cont.)



System dynamics change over time

Low trust into black-box models

I – Fundamental Problems (cont.)



Continuous and lifelong learning of environment, verifiable through structural causal models

System dynamics change over time

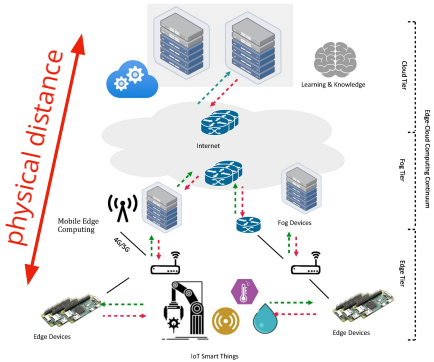
Low trust into black-box models

I – Fundamental Problems (cont.) **Hidden in main presentation**

Heterogeneity prevents simple task distribution between available devices. Don't know a priori what will be

Physi distrib CPU load) creates lots of traffic on the network.

Decentralized decision-making based on local observations and understanding



Variable drifts cause any ML model to lose accuracy over time. Cannot train models in one-shot operations but must incorporate changes continuously.

Low-trust into black-box ML models that cannot give human-verifiable chains of thought. E.g., debugging

I – Fundamental Problems (cont.) **Hidden in main presentation**

Heterogeneity prevents simple task distribution
between available devices. Don't know a priori what
will be

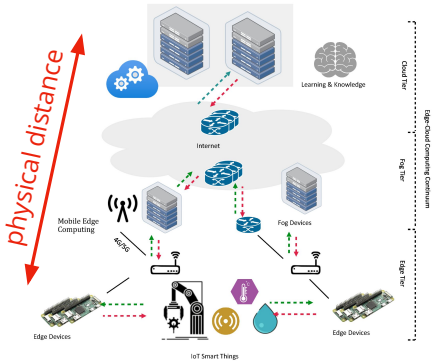
Decentralized decision-making based on local observations and understanding

Physi
distrib
CPU load) creates lots of traffic on the network.

Varia
over t
but m

Continuous and lifelong learning of environment, verifiable through structural causal models

Low-t
human-verifiable chains of thought. E.g., debugging



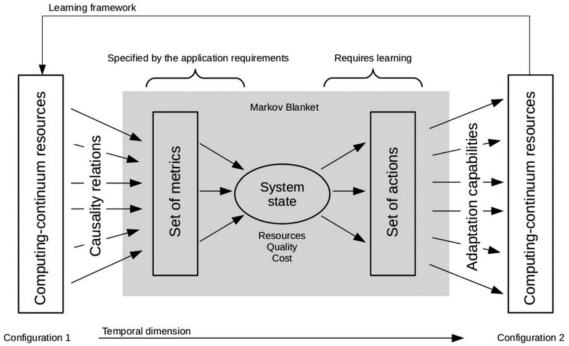
II – Markov Blanket (MB)

Individual systems (e.g., humans or devices) and their interactions can be expressed through MBs; a MB creates a **formal boundary** between a system and external states:

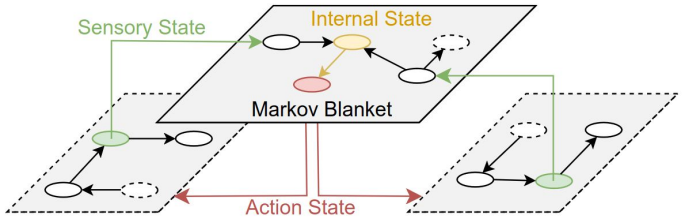
“How can a system perceive its surrounding environment and how do its actions affect the environment?”

“What information is relevant to make decisions locally [=*those that impact SLOs*], and which can I simply discard?“

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



Behavioral Markov blanket of a system [3]



Action-perception cycle between multiple entities [4]

[3] Dustdar et al., **On Distributed Computing Continuum Systems** (2023)
[4] Sedlak et al., **Markov Blanket Composition of SLOs**, at IEEE Services Edge 2024

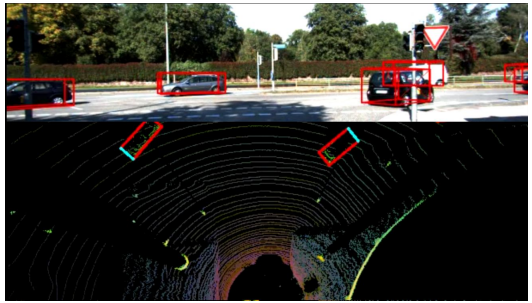
II – Stream Processing Scenarios

Commonly addressed use cases revolve around continuous **stream processing**; usually pose complex **time-critical** requirements, and have many sensory states and adaptations available.

Video Processing (Yolo V8)

Object detection in a video stream using Yolo [5]

Mobile Mapping (Lidar)



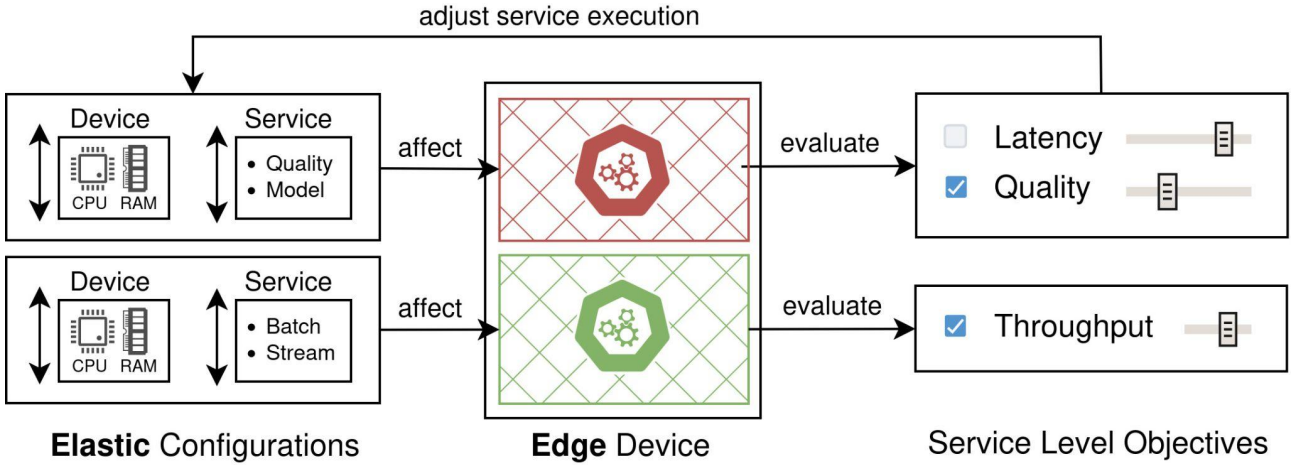
Creating a mobile map from binaries using Lidar [5]

QR Scanner (OpenCV)

QR code scanning in a video using OpenCV [5]

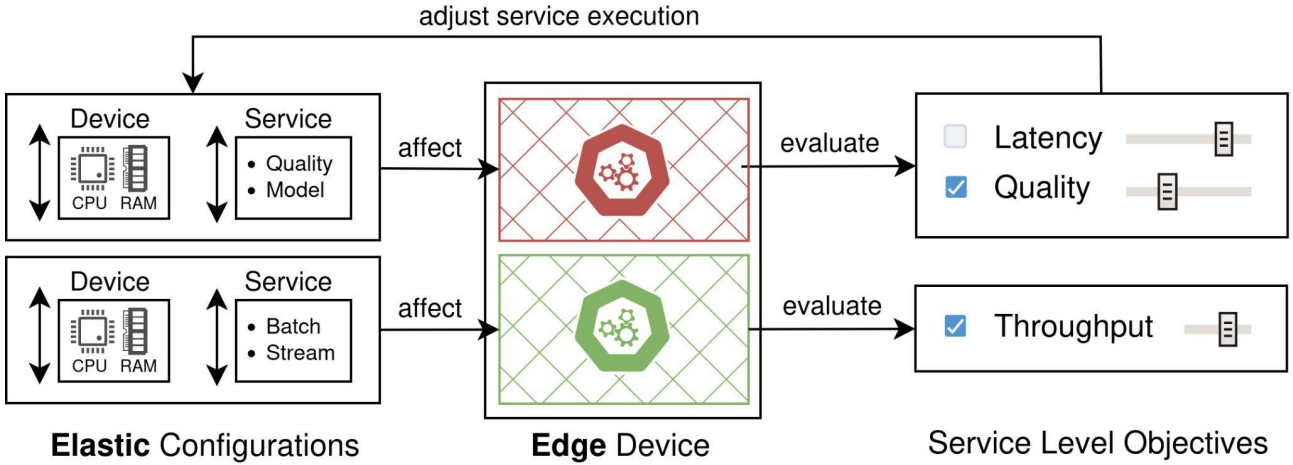
[5] Sedlak et al., **Adaptive Stream Processing on Edge Devices through Active Inference** (2025)

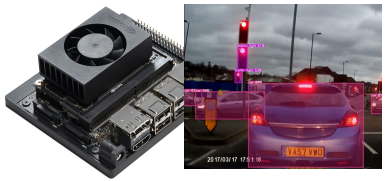
II - Case Study: Optimizing Stream Processing



[6] Sedlak et al., **Multi-dimensional Autoscaling of Processing Services: A Comparison of Agent-based Methods** (2025)

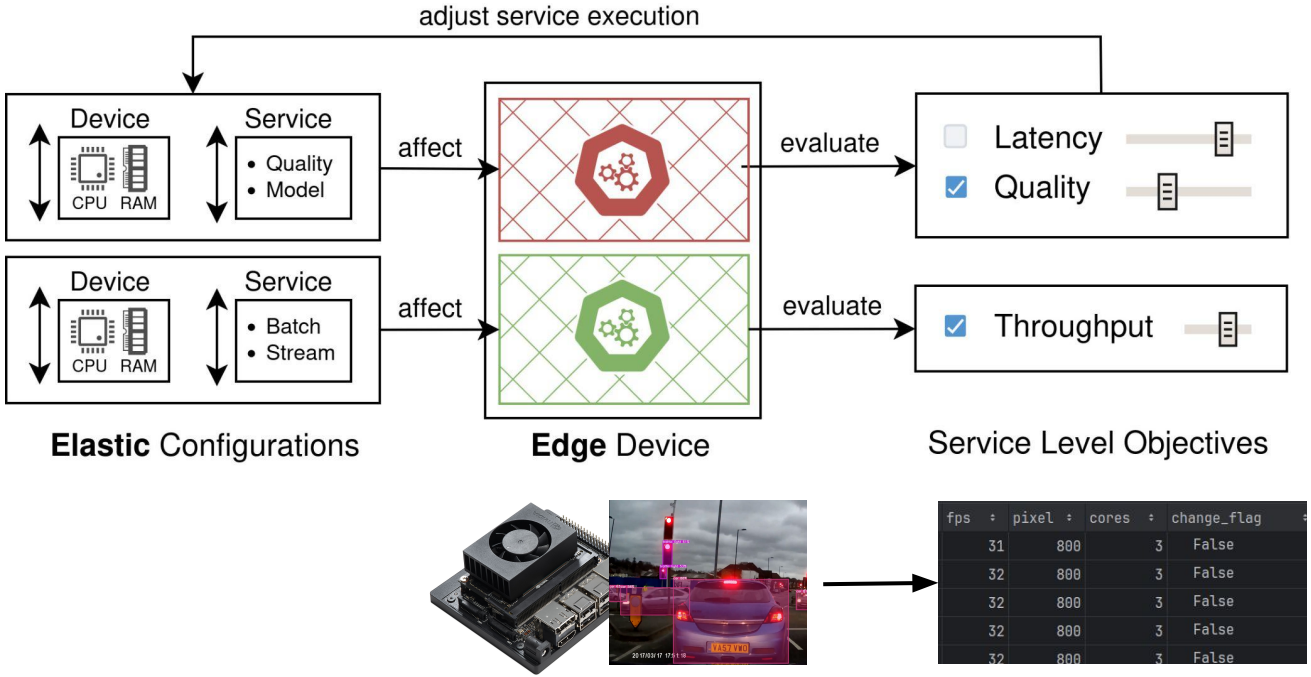
II - Case Study: Optimizing Stream Processing





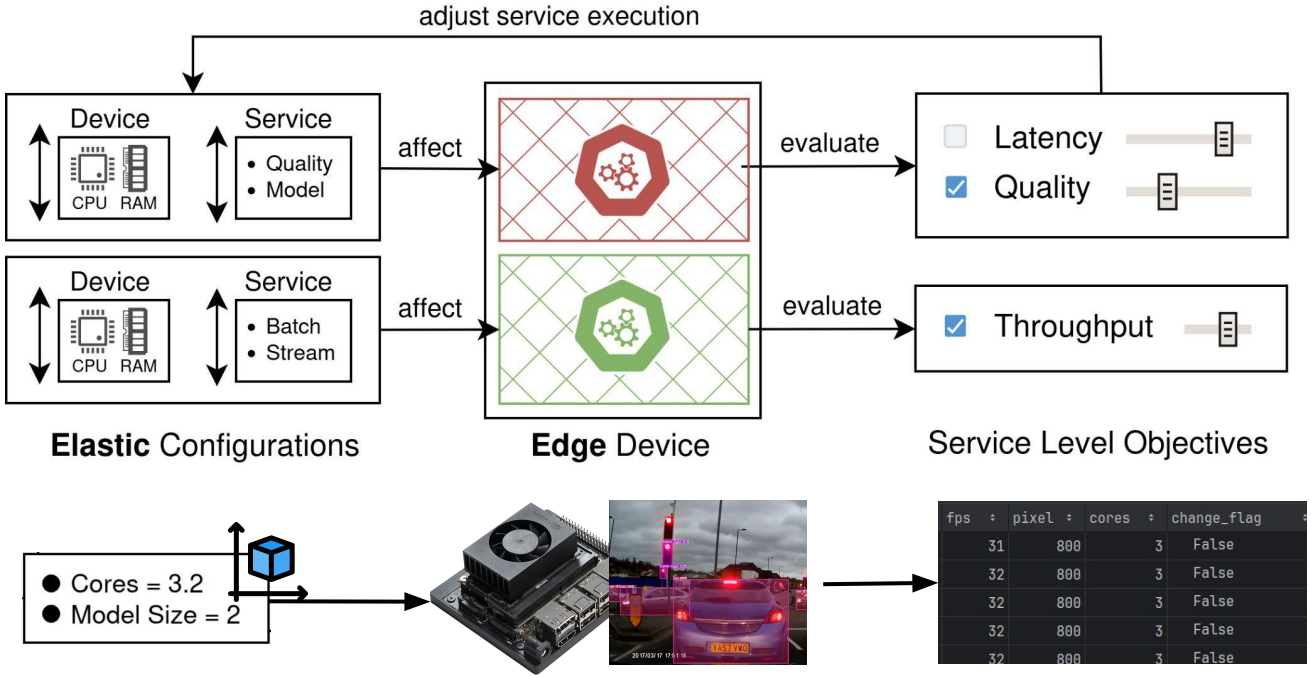
[6] Sedlak et al., **Multi-dimensional Autoscaling of Processing Services: A Comparison of Agent-based Methods** (2025)

II - Case Study: Optimizing Stream Processing

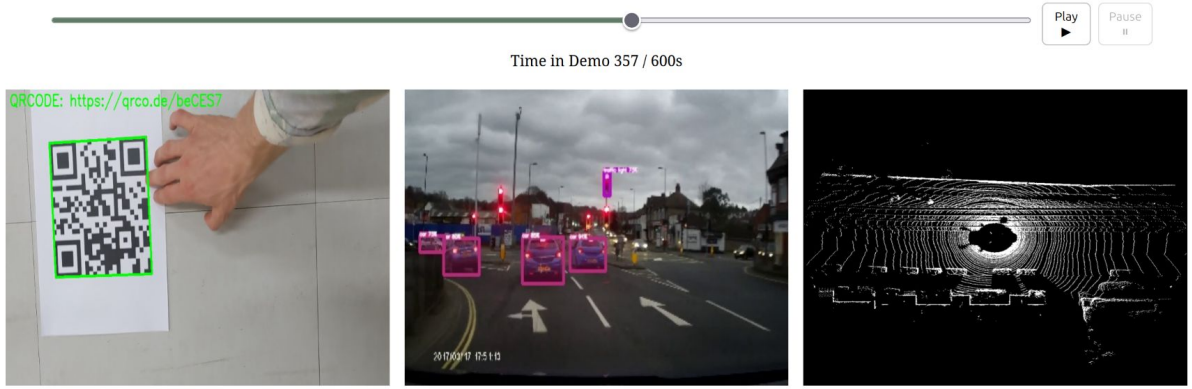


[6] Sedlak et al., **Multi-dimensional Autoscaling of Processing Services: A Comparison of Agent-based Methods** (2025)

II - Case Study: Optimizing Stream Processing



[6] Sedlak et al., **Multi-dimensional Autoscaling of Processing Services: A Comparison of Agent-based Methods** (2025)



Setup:

1 Edge Device
3 Processing Service

[7] Sedlak et al., **Visual Insights into Agentic Optimization of Pervasive Stream Processing Services** (2025)

Setup:

1 Edge Device
3 Processing Service

Many goals (SLOs), but no idea how to fulfill

[7] Sedlak et al., **Visual Insights into Agentic Optimization of Pervasive Stream Processing Services** (2025)

Setup:

1 Edge Device
3 Processing Service

Many goals (SLOs), but no idea how to fulfill

Explore through quick interventions how to ensure SLO fulfillment

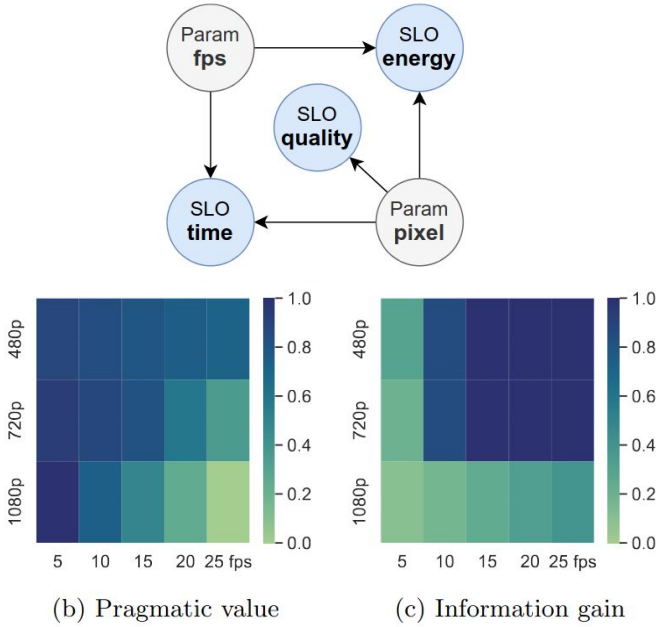
[7] Sedlak et al., **Visual Insights into Agentic Optimization of Pervasive Stream Processing Services** (2025)

II – Wider Implications

Interpretable behavior [6]

– Develop verifiable variable relations, e.g., increasing quality (pixel) also increases energy usage; adjust **parameters** (i.e., pixel & fps) according to SLOs

– Quantified preferences of an agent: (1) expected SLO fulfillment or (2) potential model improvement; determine the behavior of the scaling agent



(b) Pragmatic value (c) Information gain

[4] Sedlak et al., **Markov Blanket Composition of SLOs**, at IEEE Services EDGE 2024
[5] Sedlak et al., **Adaptive Stream Processing on Edge Devices through Active Inference** (2025)

II – Solution Properties

Hidden in main presentation

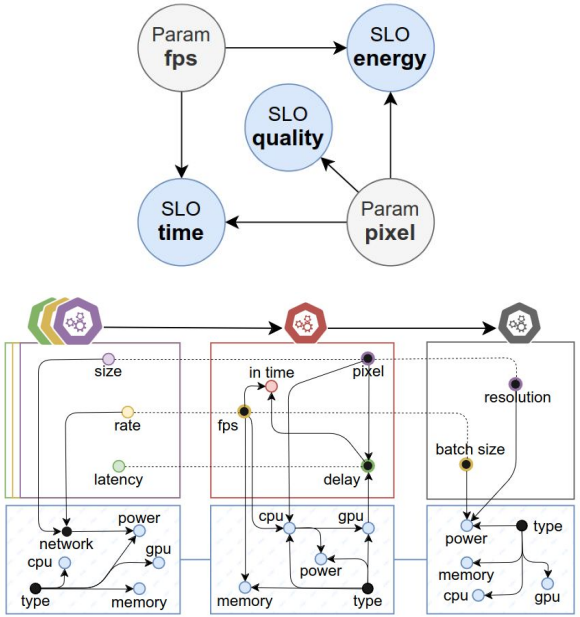
Interpretable behavior [6]

– Develop verifiable variable relations, e.g., increasing quality (pixel) also increases energy usage; adjust **parameters** (i.e., pixel & fps) according to SLOs

– Quantified preferences of an agent: (1) expected SLO fulfillment or (2) potential model improvement; determine the behavior of the scaling agent

Continuous composition [4]

– Gradually create increasingly accurate models for individual processing services; continuously compose to estimate the **impact** they have on each other



[4] Sedlak et al., **Markov Blanket Composition of SLOs**, at IEEE Services EDGE 2024
[5] Sedlak et al., **Adaptive Stream Processing on Edge Devices through Active Inference** (2025)

III – Research Directions

[8] Sedlak B. et al., **Service Orchestration in the Computing Continuum: Structural Challenges and Vision** (2025)

III – Research Directions
System & Action Simulation [25]

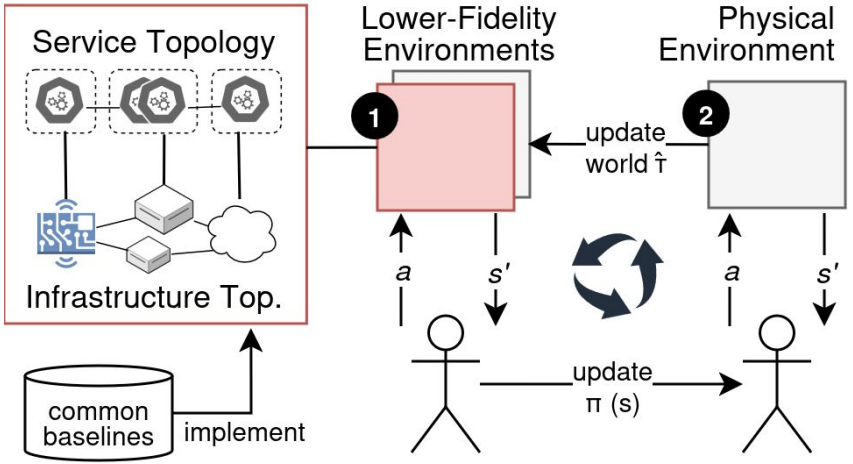
Running meaningful experiments is hard:

(1) Tedious process to implement realistic topologies and baselines; maintaining processing infrastructure is costly.

→ Provide simulation environments to train agents; **digital twins** for testing hypotheses

(2) Simulation environments are not accurate enough to reflect realistic conditions.

→ Continuously incorporate feedback from physical environments to improve accuracy



[8] Sedlak B. et al., **Service Orchestration in the Computing Continuum: Structural Challenges and Vision** (2025)

III – Research Directions
System & Action Simulation [25]

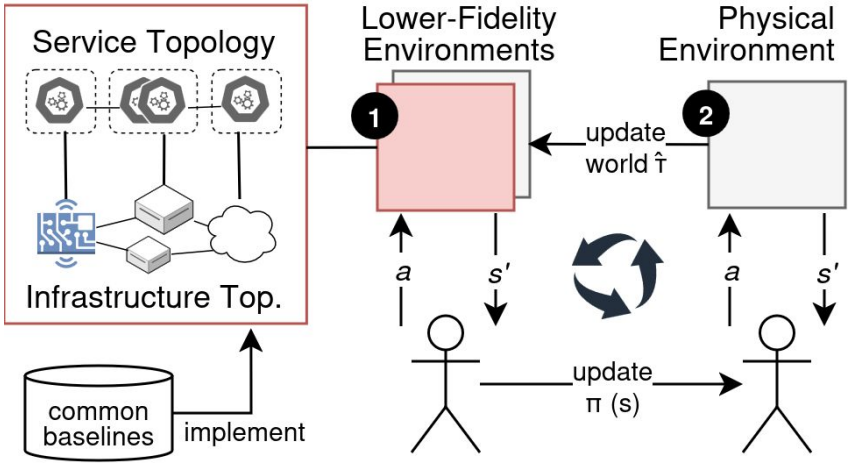
Running meaningful experiments is hard:

(1) Tedious process to implement realistic topologies and baselines; maintaining processing infrastructure is costly.

→ Provide simulation environments to train agents; **digital twins** for testing hypotheses

(2) Simulation environments are not accurate enough to reflect realistic conditions.

→ Continuously incorporate feedback from physical environments to improve accuracy



[8] Sedlak B. et al., **Service Orchestration in the Computing Continuum: Structural Challenges and Vision** (2025)

III – Research Directions
Model Update & Inference [25]

Hidden in main presentation

Continuous learning poses an **overhead**:

(1) Rolling out new models might not improve performance and lead to inconsistencies

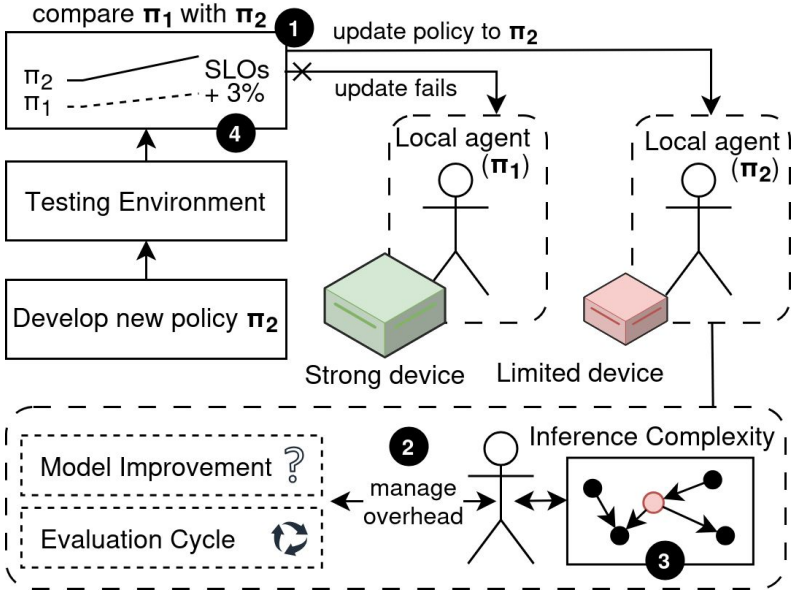
(2) Any adaptive agent poses itself poses additional load for a computing device

→ Make learning always context-aware

Ensure there's a **human-in-the-loop**:

(3) Raw metrics cannot be interpreted by human supervisors and require "massaging"

(4) Adaptive behavior must be aligned with high-level user intents (requirements)



III – Research Directions
Model Update & Inference [25]

Hidden in main presentation

Continuous learning poses an **overhead**:

(1) Rolling out new models might not improve performance and lead to inconsistencies

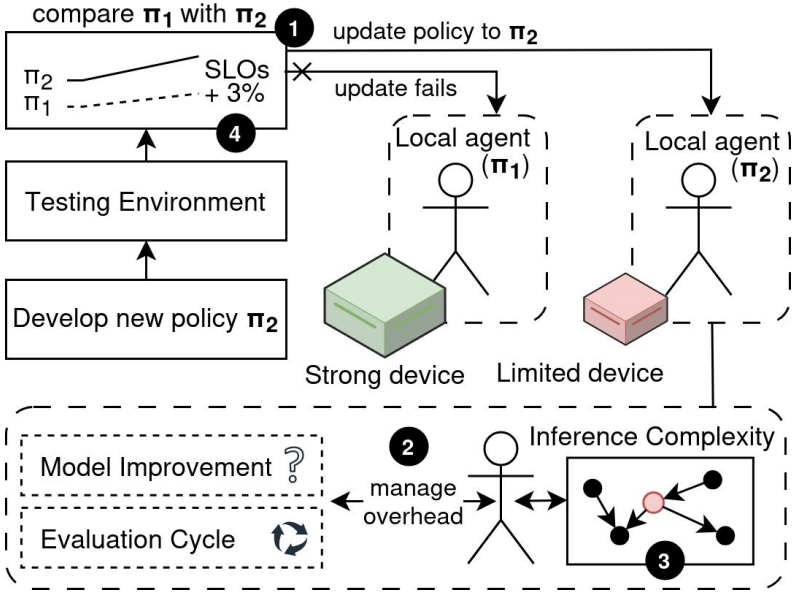
(2) Any adaptive agent poses itself poses additional load for a computing device

→ Make learning always context-aware

Ensure there's a **human-in-the-loop**:

(3) Raw metrics cannot be interpreted by human supervisors and require "massaging"

(4) Adaptive behavior must be aligned with high-level user intents (requirements)



III – Research Directions
Interplay of Infrastructure [25]

There is no unified infrastructure platform:

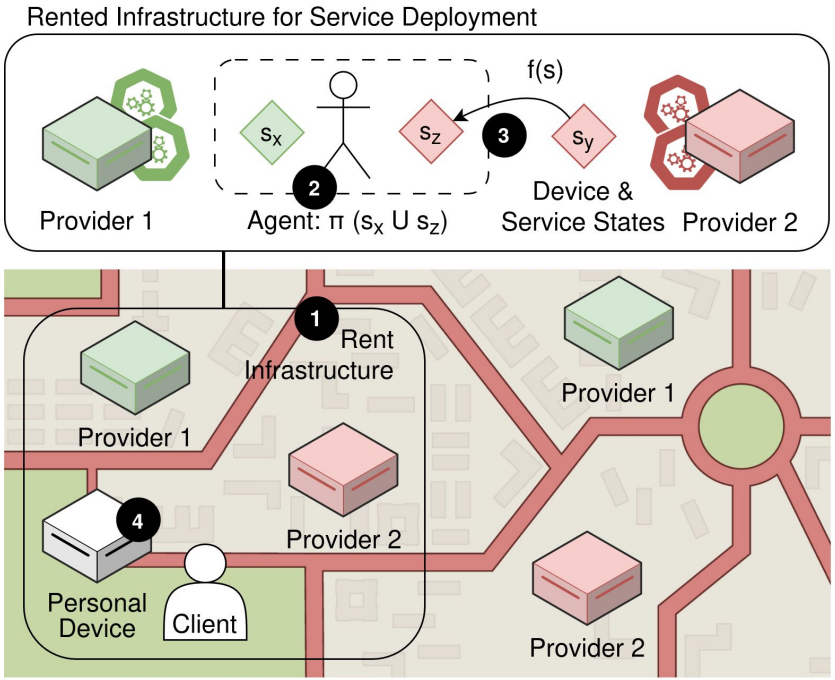
(1) Cannot combine arbitrarily nearby infrastructure due to restrictive policies

(4) Cannot locate arbitrary workloads on personal computing devices (**sovereignty**)

No entity has the full system view:

(2) Must cooperate with other entities but agents don't know their actions.

(3) Must target a global optimum without having information on many system parts



[8] Sedlak B. et al., **Service Orchestration in the Computing Continuum: Structural Challenges and Vision** (2025)

III – Research Directions
Interplay of Infrastructure [25]

There is no unified infrastructure platform:

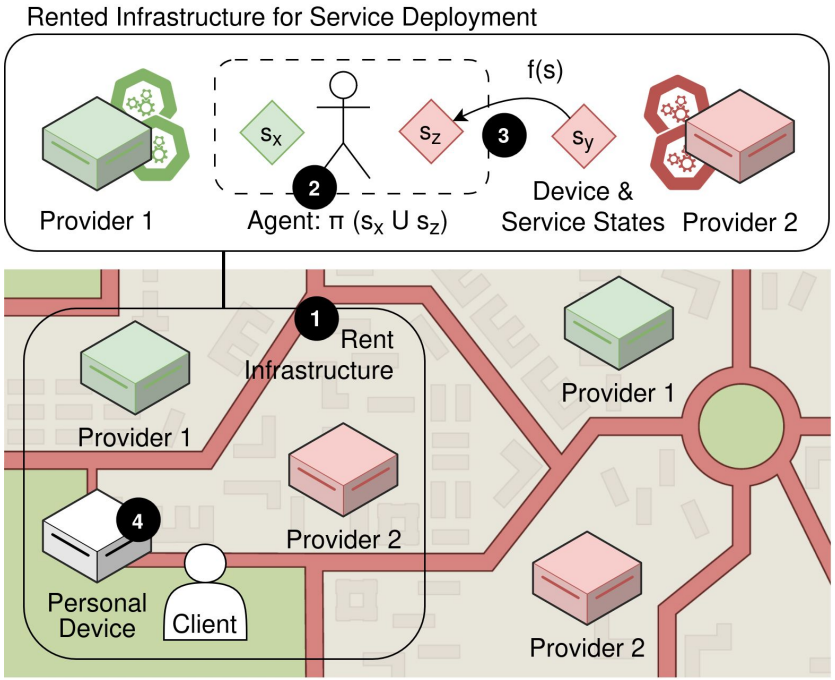
(1) Cannot combine arbitrarily nearby infrastructure due to restrictive policies

(4) Cannot locate arbitrary workloads on personal computing devices (**sovereignty**)

No entity has the full system view:

(2) Must cooperate with other entities but agents don't know their actions.

(3) Must target a global optimum without having information on many system parts

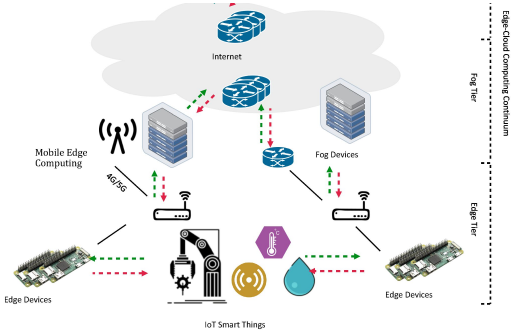


[8] Sedlak B. et al., **Service Orchestration in the Computing Continuum: Structural Challenges and Vision** (2025)

Summary

IoT & Edge enable **large-scale distributed services** that optimize our daily routines; CC as underlying infrastructure for supporting these services

Resource limitations and device **heterogeneity** complicate service orchestration; device & service behavior not guaranteed, leads to violated SLOs



Shift to **decentralized** decision-making, where individual devices and services only consider their local state and understanding for optimizing SLOs.

Create decision models with **few interventions** and allow human operators to **verify** results.