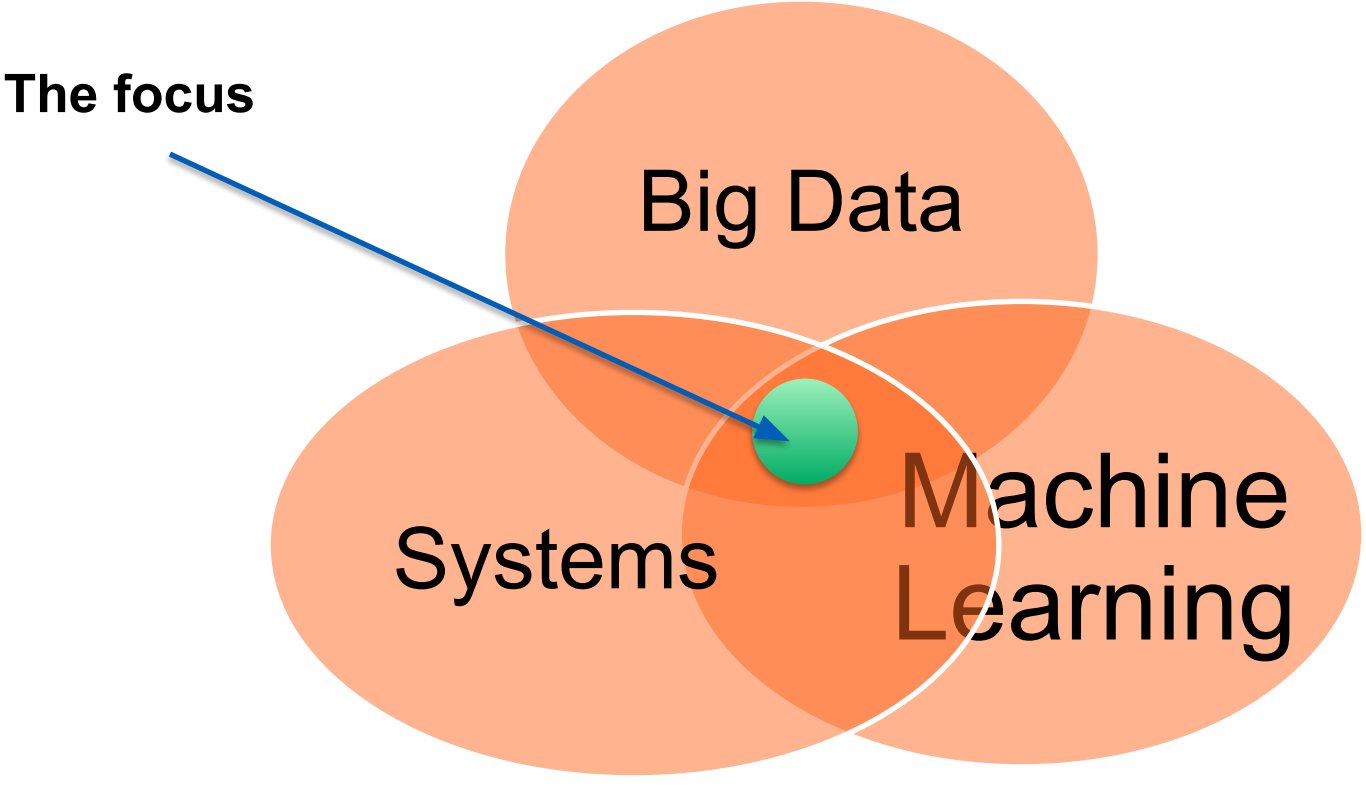
Aalto University
School of Science

Robustness, Reliability, Resilience, and Elasticity (R3E) for Machine Learning Systems in Edge-Cloud Continuum

Hong-Linh Truong
Department of Computer Science
linh.truong@aalto.fi, *https://rdsea.github.io*

CS-E4660 Advanced Topics in Software Systems, Fall 2024
11/09/2024

Our focus in this course



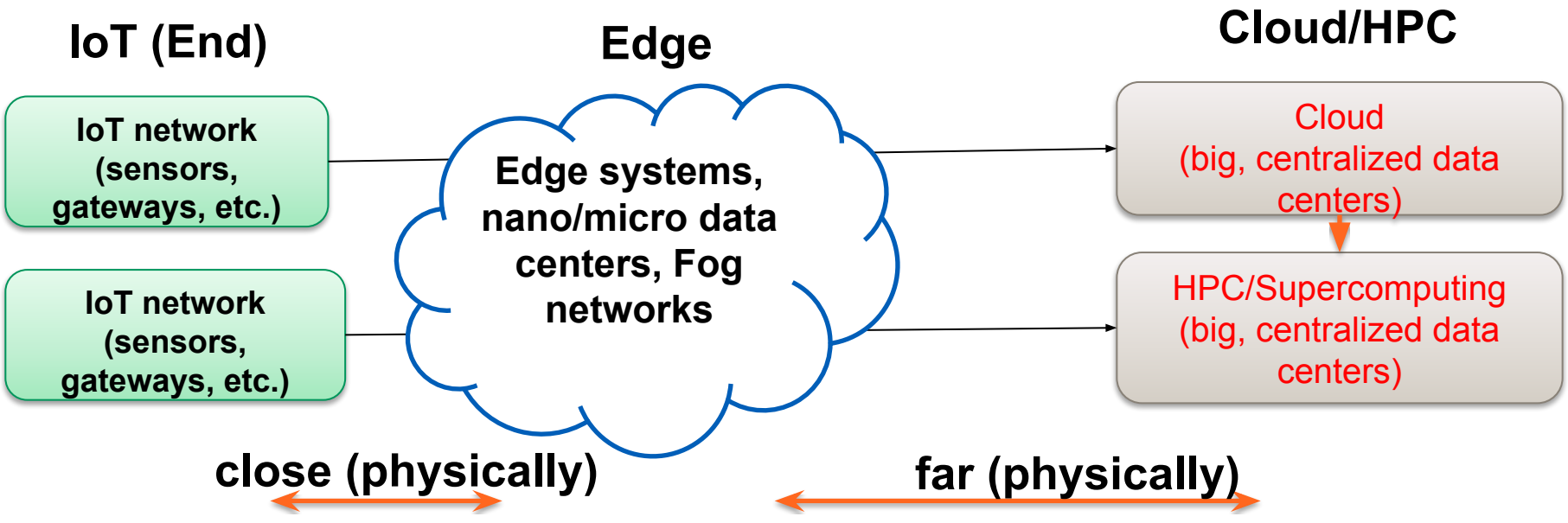
What we will not focus on?

- **The basic assumption in big data and ML systems**
  - how to train ML models
  - how to run CI/CD pipelines for big data/ML
  - how build a big data/ML system in general
- **What we want to focus on**
  - see and view systems as a whole, end-to-end
  - understand key performance, dynamicity, elasticity, explainability issues for end-to-end big data/ML systems
  - monitoring, observability, and optimization techniques
  - open problems for research and theses

Learning objectives

- Understand Edge-cloud continuum and complexity in end-to-end ML systems
- Explore and study basic concepts and issues when engineering ML systems in edge-cloud continuum
- Understand design goals and concerns for robustness, reliability, resilience and elasticity (R3E) of ML systems

IoT-Edge-Cloud



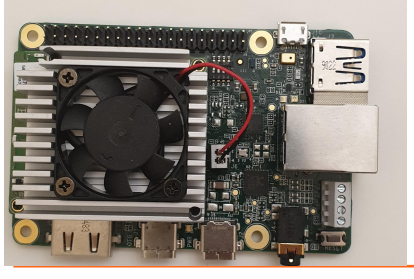
"Edge" is just an abstraction view ("near edge" vs "far edge")

Heterogeneous computing infrastructures

- **Various tasks**
  - including data processing, inference, storage, etc
  - different computing requirements
    - *GPU, CPUs, etc.*
- **Different types of deployment**
  - single vs distributed locations
  - distributed machines
  - edge and cloud

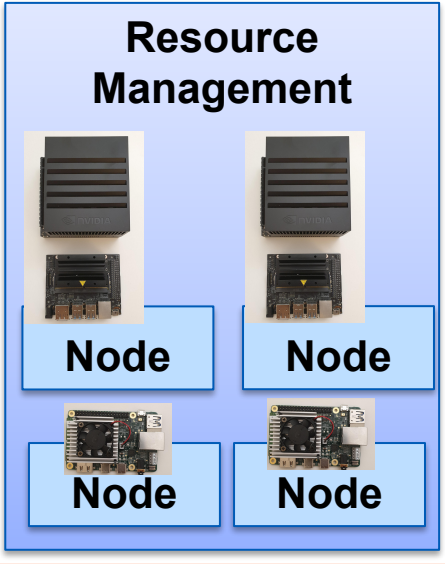
Edge systems: new types of edge and edge-cloud

Coral with Edge TPU System-on-Module, Google Edge TPU ML accelerator coprocessor

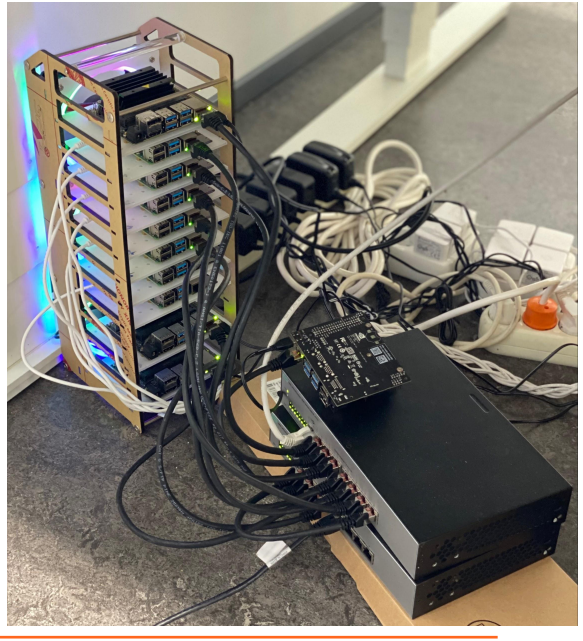


Jetson NVIDIA (GPU+CPU)

Distributed Edge systems



Our testbed for this course



→ **challenges with software libraries, updates, etc.**

Cloud/HPC

- **Clusters of VMs/containers**
  - e.g., in Aalto we use CSC (https://www.csc.fi/)
- **High performance systems**
- **Known accelerators**
  - GPU and FPGA
- **(New) AI Accelerators/Processing Units**
  - TPU (Tensor Processing Unit)
  - Neutral Network Processor (NNP)
  - Vision Processor Unit (VPU)
  - IPU(Intelligent Processing Unit)

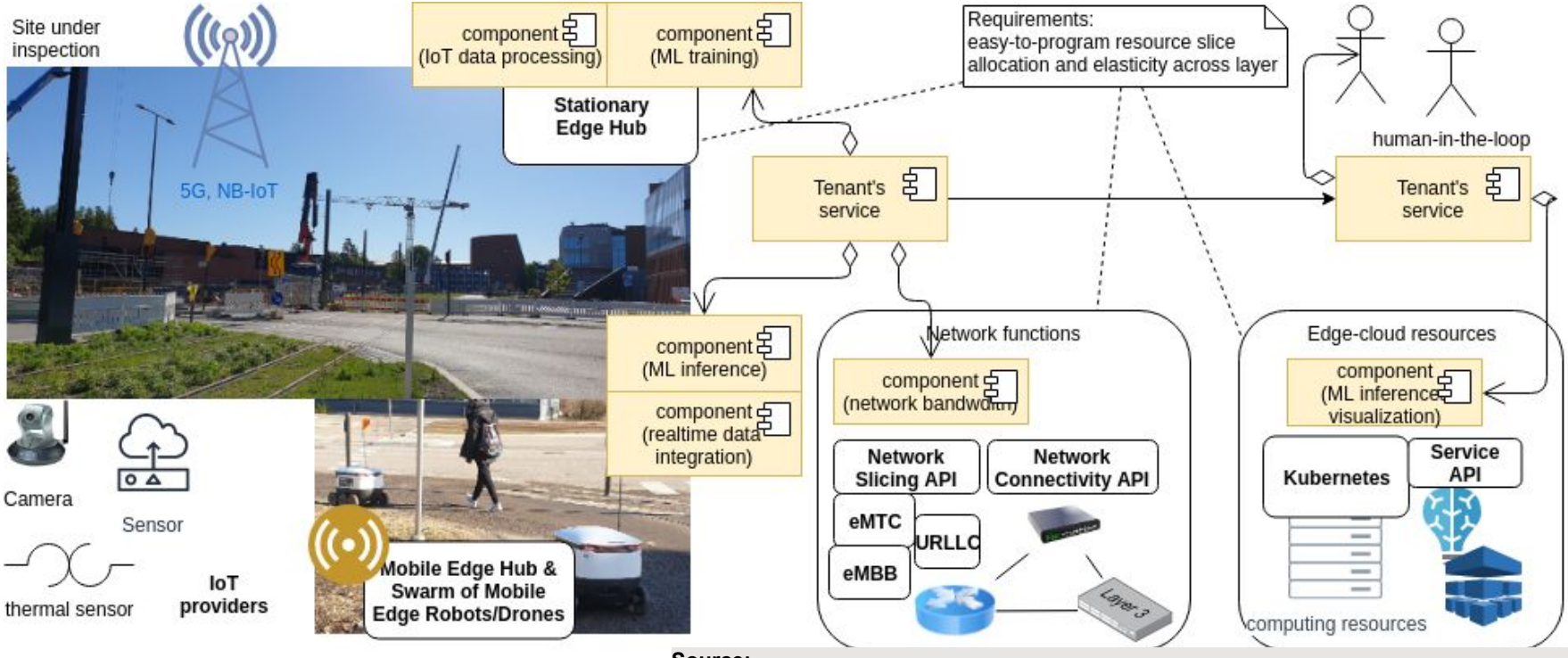
Illustration of the HPE Cray EX cabinets. Copyright: Hewlett Packard Enterprise

Figure source: https://www.lumi-supercomputer.eu/deep-dive-into-the-building-of-the-lumi-data-center/

Edge-cloud continuum

- **Enable widely decentralized complex workloads (AI/ML, data analytics, real-time operations, etc)**
  - in distributed and heterogeneous environments
- **Interconnecting edge and cloud resources and using them in similar manners**
  - like in the same system to, moving tasks seamlessly between edge and clouds
  - reduce issues and costs due to diverse development, deployment and operations
- **From the technical view point: for both edge and cloud**
  - use similar enabling technologies, like containers and orchestrators
  - employ similar management techniques and methods (deployment, monitoring, policies, etc.)

Example of an edge-cloud continuum

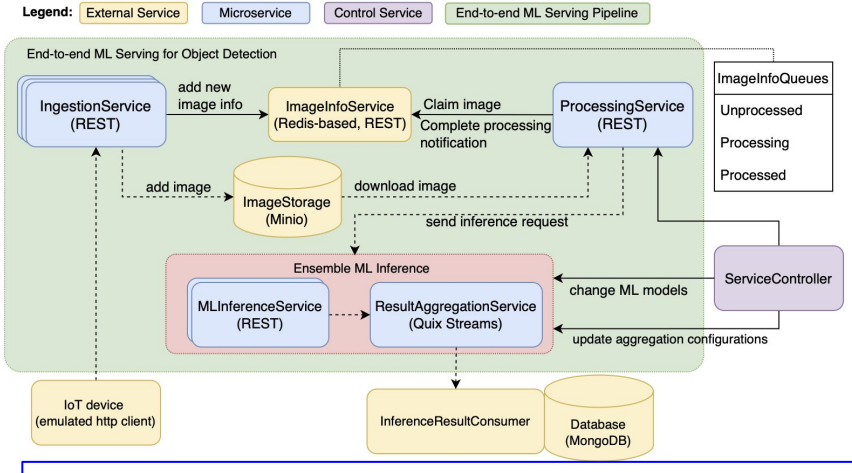


In city blocks, villages, etc.

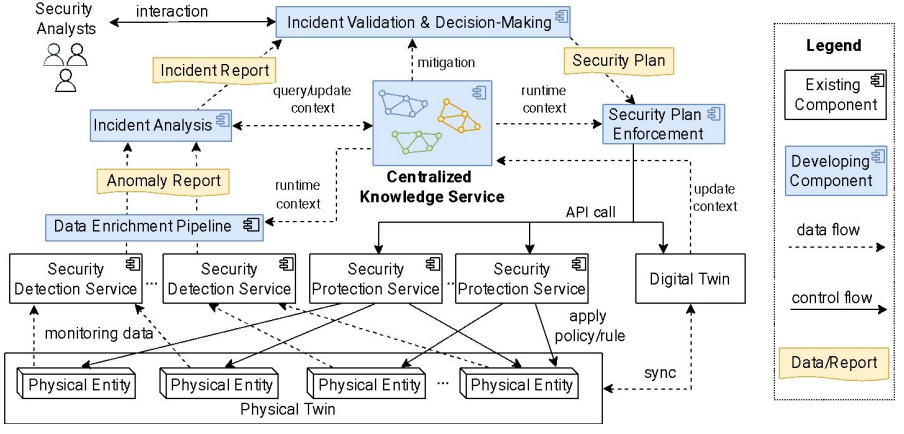
Source: https://www.researchgate.net/publication/361261810_Robustness_via_Elasticity_Accelerators_for_the_IoT-Edge-Cloud_Continuum

Machine learning in edge-cloud continuum

AI/ML applications: e.g., malware detection, object detection & classification, digital twins



Orchestration, observability and analytics tools/services for edge-cloud ML applications/systems



Including also Large-Language Models (LLMs) utilities and AI Agents

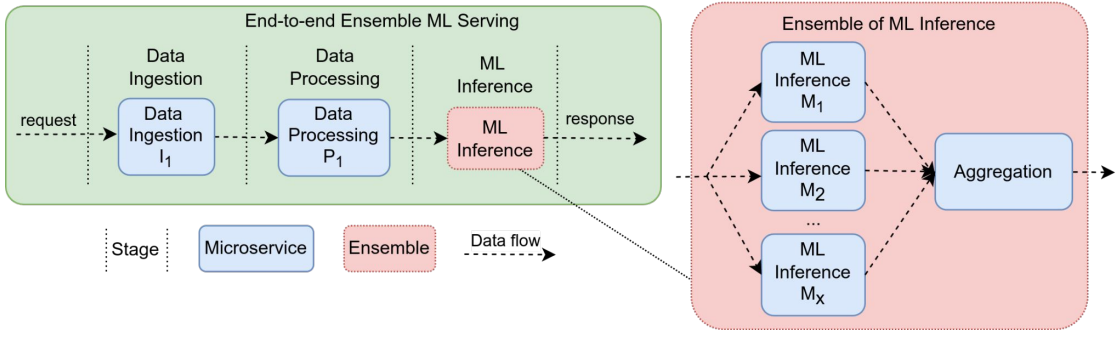
kubernetes

Aalto University
School of Science

Basic interactions and components in ML in edge-cloud continuum

CS-E4660 Advanced Topics in Software Systems, Fall 2024
11/09/2024

Software structure: complex and diverse architectural styles and interactions



While:

- Microservices, serverless, batch workflows, stream processing, & reactive system
- Heterogeneous infrastructures, compute resources and connectivities

AI/ML-based inference/prediction is just one component in a complex, end-to-end AI/ML system

Software runtime and libraries: not a single software framework/stack

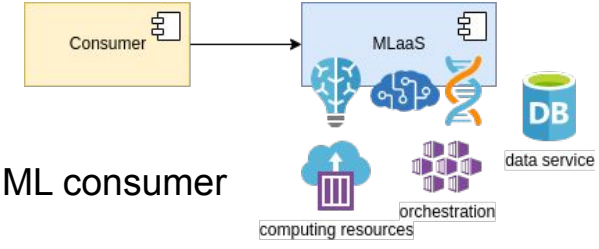
- Many **platform services** according to workloads and architectures, e.g., *messaging systems* with MQTT, NATS, CoAP, AMQP, Kafka, or Pulsar

- Multiple powerful **AI/ML programming frameworks** (PyTorch, TensorFlow, scikit-learn, Keras, OpenLLM, etc.)

- Diverse **AI/ML serving platforms** (Seldon, BentoML, Triton, Ray, ZenML, Lightning LitServe, etc.)

Service models

Two stakeholders engagement (Google, Microsoft, OpenAI, etc.)

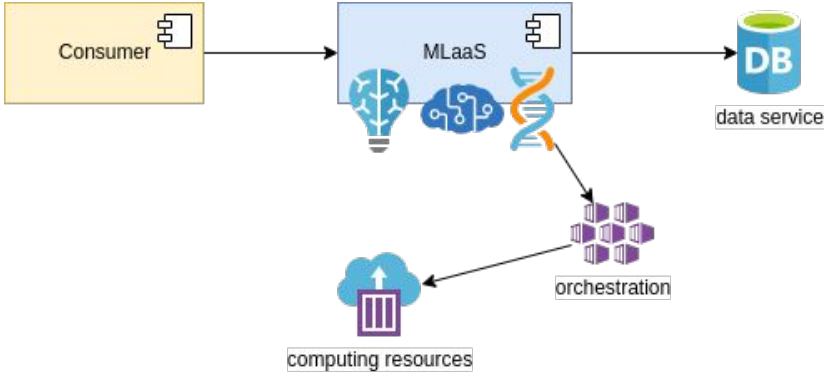


ML consumer

All belongs to the same provider

A special form: for internal uses in a large company

Three stakeholders engagement

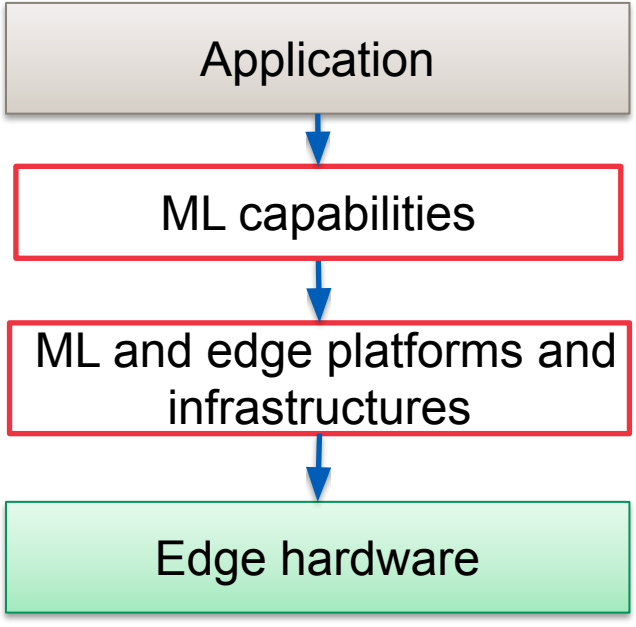


other providers

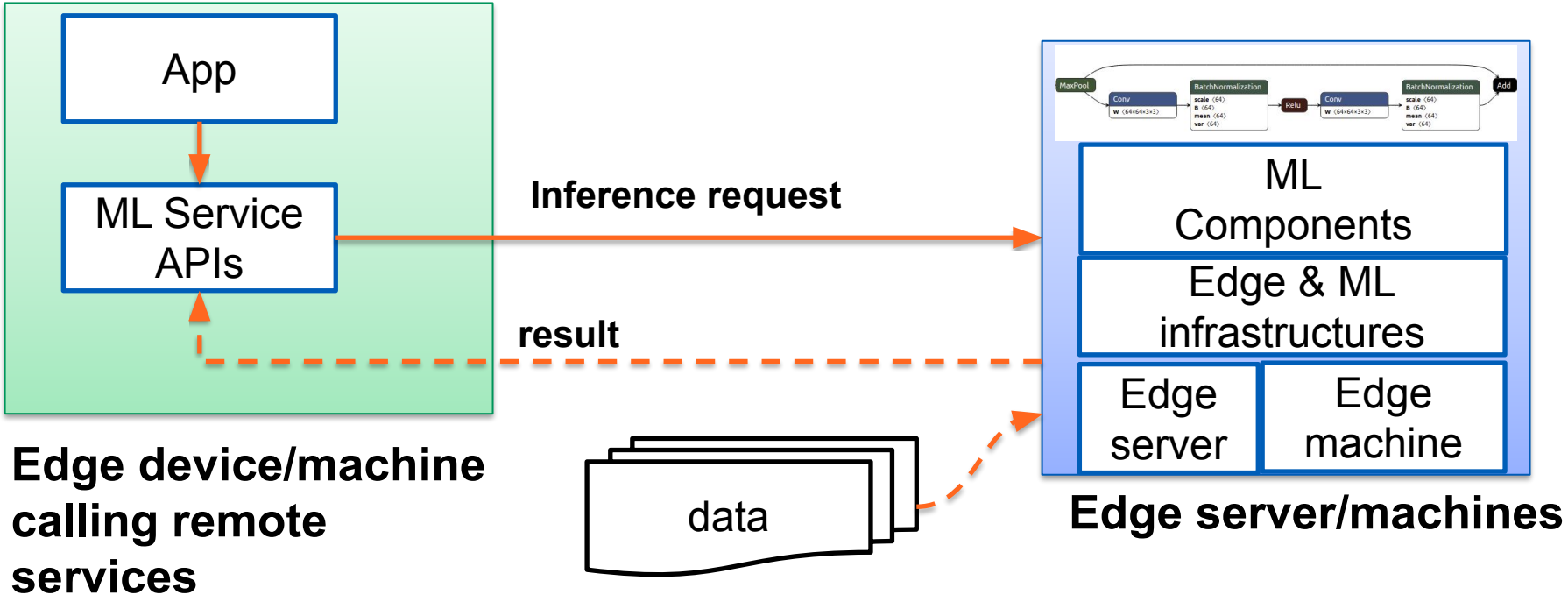
Relies on external platforms for orchestration, computing resources and other data (e.g., using Sedon, SagerML, etc.)

Interaction models in edge-cloud ML systems

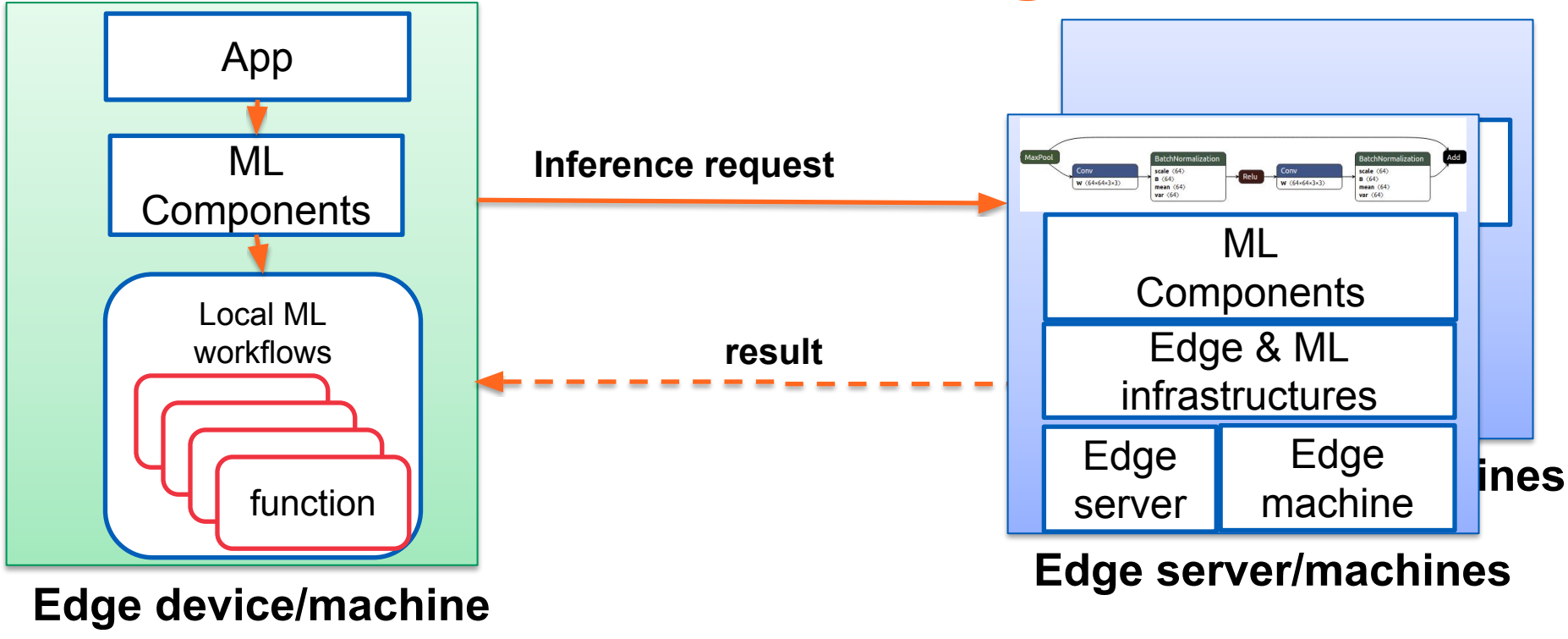
Which components do what, and where are they?



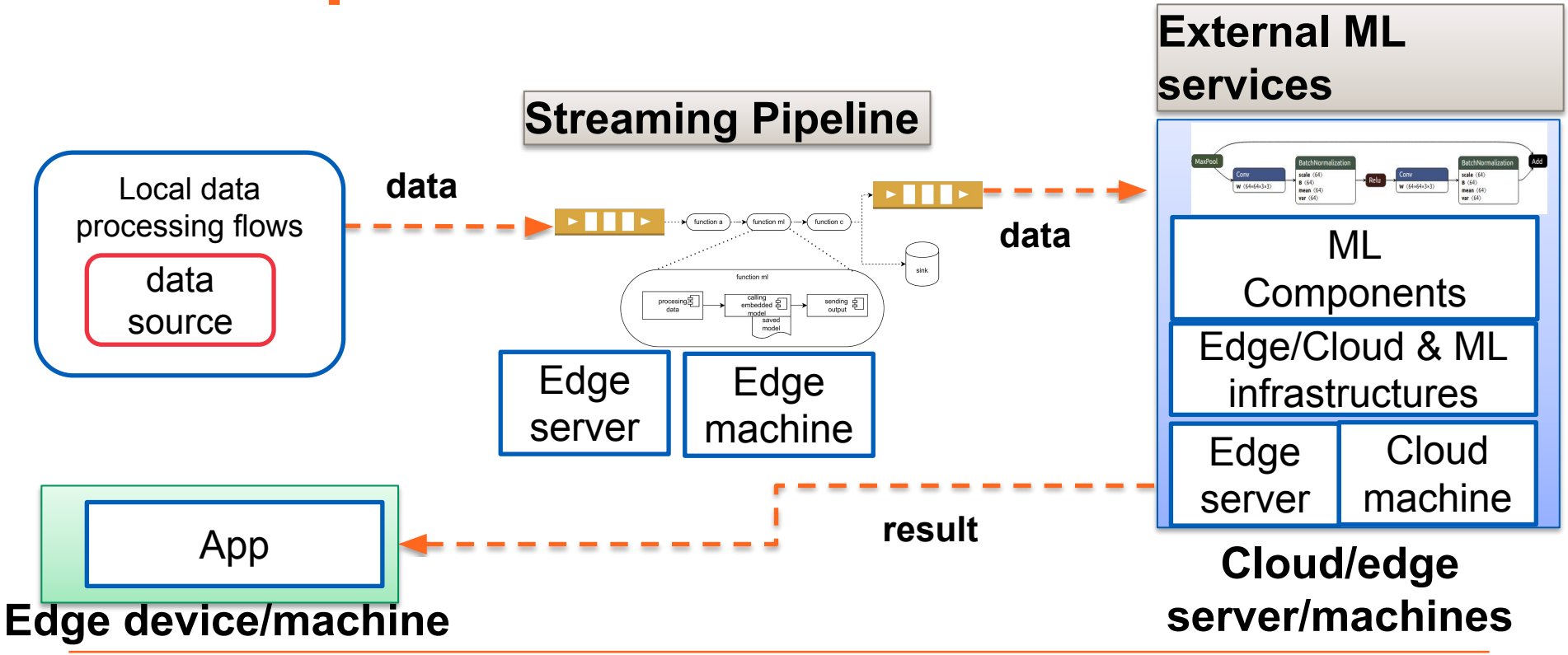
Interaction models: common client-server



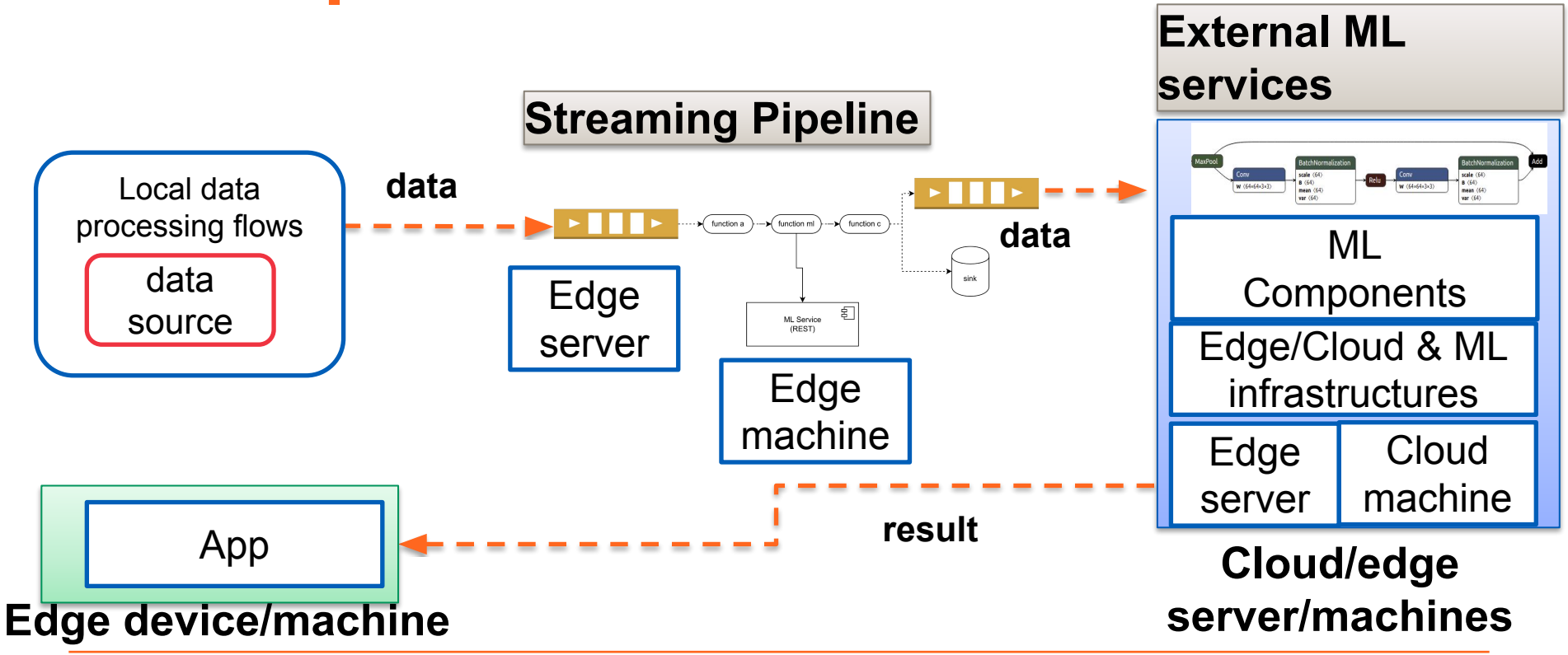
Interaction models: pre-processing and remote ML offloading



Interaction models: ML composition



Interaction models: ML composition



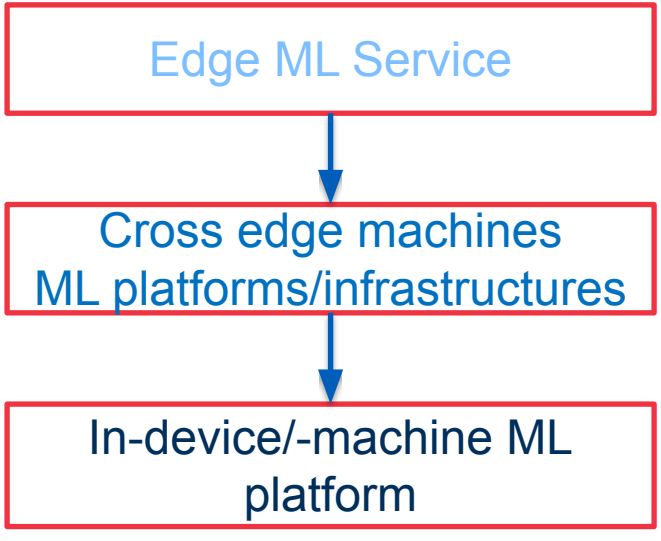
Aalto University
School of Science

R3E for end-to-end ML systems

CS-E4660 Advanced Topics in Software Systems, Fall 2024
11/09/2024

Multiple levels of optimization

Scope/level of abstraction



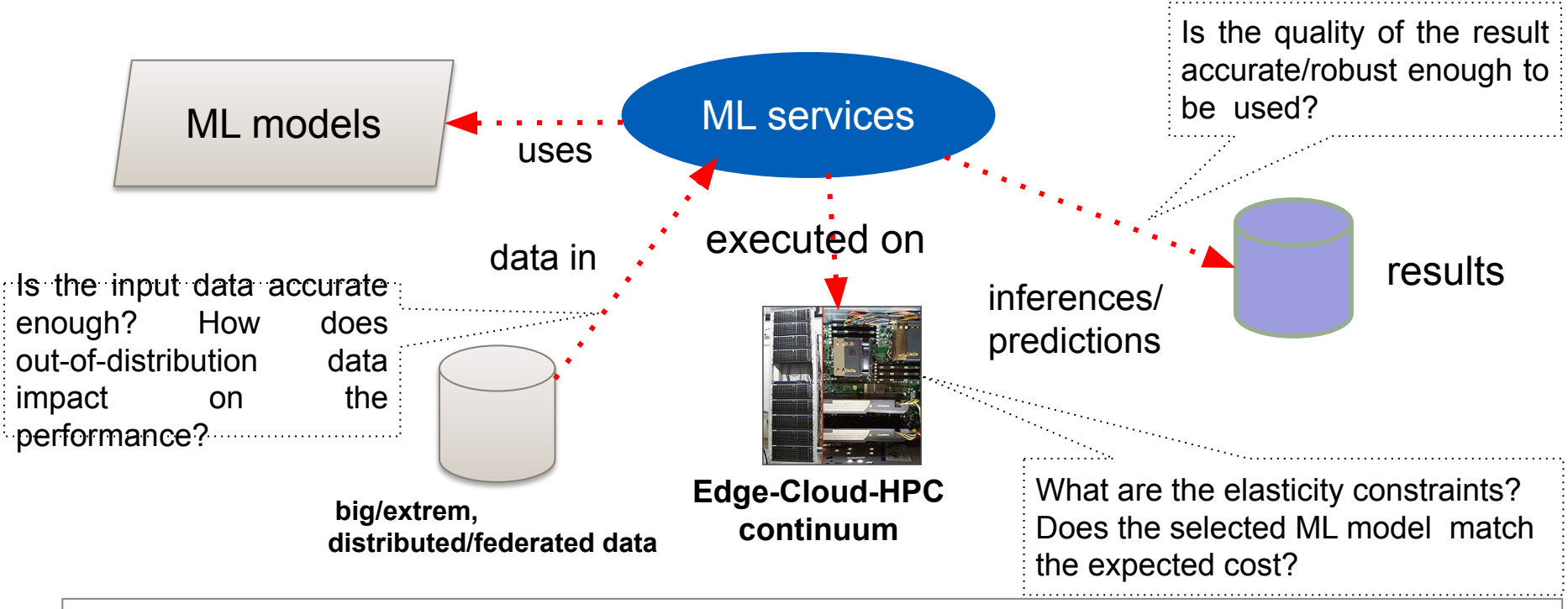
Research issues

ML serving, ML elasticity, ML observability and policies

ML function partitioning, distributed computation, orchestration, provisioning/(deployment, monitoring ..

Device-machine specific optimization

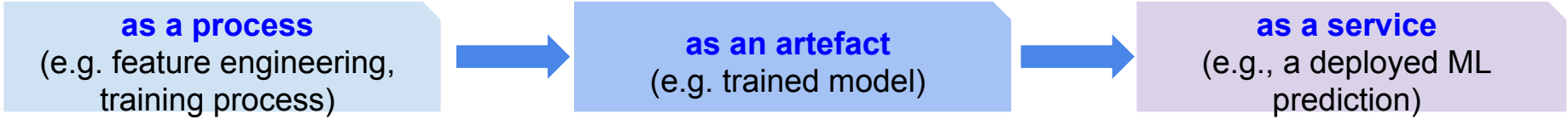
From distributed software system engineering: QoA dependencies across software layers and aspects



Truong, Coordination-aware assurance for end-to-end machine learning systems: the R3E approach, AI Assurance, 2023

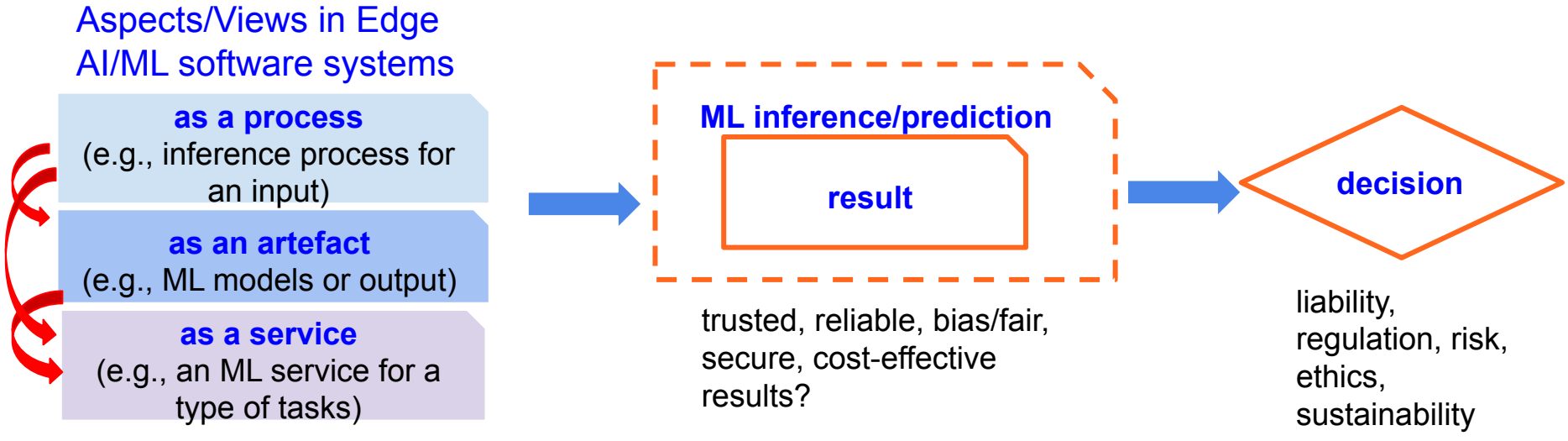
Strongly interdependencies

Any problem would lead to a huge waste (engineering effort, operation cost, societal impact due to wrong inference/prediction)



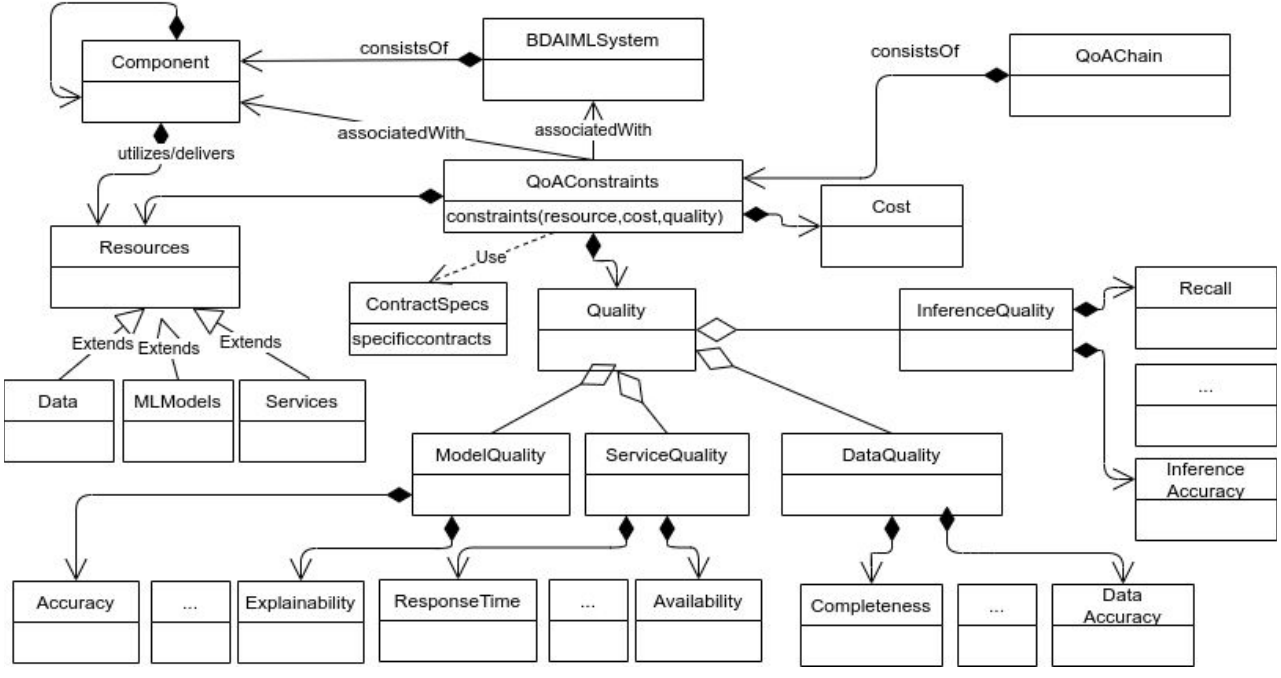
But they are too complex so currently we cannot examine them in an end-to-end manner

QoA4ML: addressing quality for edge AI/ML software from process, artefact and service views



Key attributes/indicators

Just example, can be more!



Source:
https://www.researchgate.net/publication/341762862_R3E_-An_Approach_to_Robustness_Reliability_Resilience_and_Elasticity_Engineering_for_End-to-End_Machine_Learning_Systems

Objectives for end-to-end ML systems engineering

- **Deal with end-to-end aspects that the real world requires**
  - e.g., not just ML models and their optimization
- **Reduce software and data engineering effort**
- **Scale our systems**
  - big data, large-scale infrastructures and high number of customers/tenants
- **Optimize the system under various constraints**
- **Offer a production-level "reliable service" for customers/tenants**

Our focus – R3E

- **Robustness**
  - ability to cope with errors
- **Reliability**
  - ability to function according to the indented specification (in a proper way)
- **Resilience**
  - “ability to provide the required capability in the face of adversity”(https://www.sebokwiki.org/wiki/System_Resilience)
- **Elasticity**
  - ability to stretch and return to normal forms (under external forces)

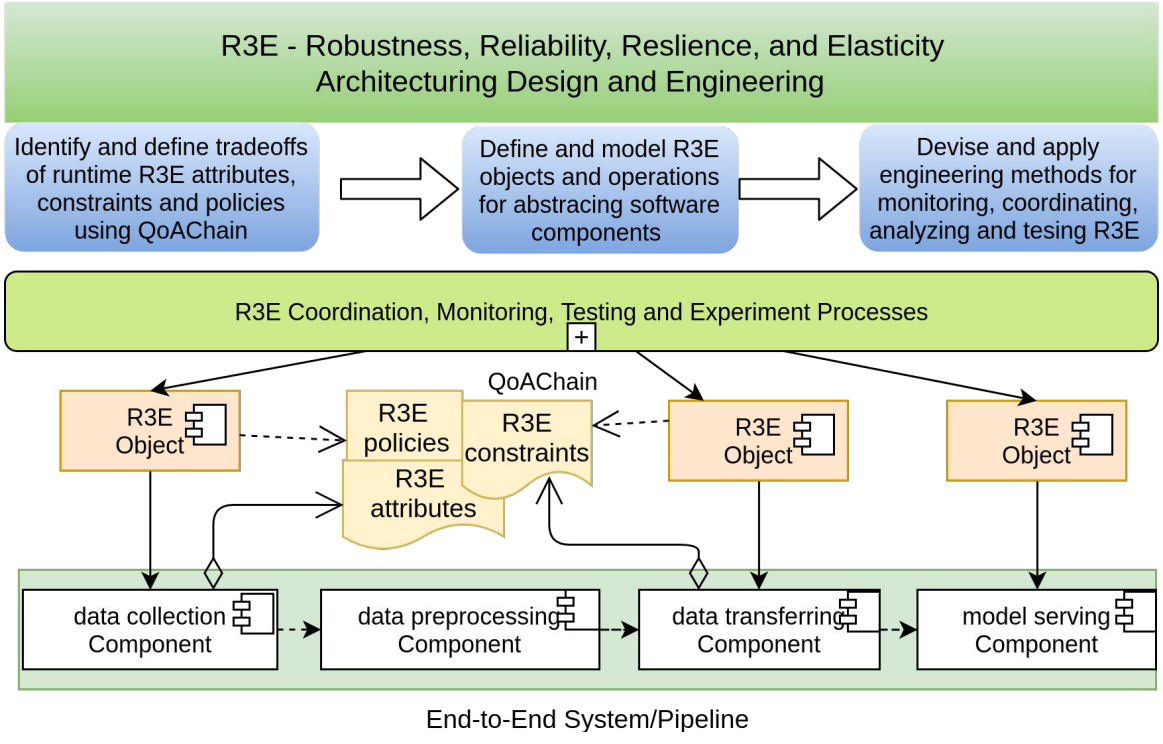
Short summary

R3E attributes	Cases from big data view	Cases from machine learning view
Robustness	deal with erroneous and bad data (Zhang et al., 2017), data processing job robustness	dealing with imbalanced data, learning in an open-world (out of distribution) situations (Kulkarni et al., 2020; Sehwag et al., 2019; Saria and Subbaswamy, 2019; Hendrycks and Dietterich, 2019)
Reliability	reliable data sources, support of quality of data (Zhang et al., 2020; Lee, 2019), reliable data services (Kleppmann, 2016), reliable data processing workflows/tasks (Zheng et al., 2017)	reliable learning and reliable inference in terms of accuracy and reproducibility of ML models (Saria and Subbaswamy, 2019; Henderson et al., 2017); uncertainties/confidence in inferences; reliable ML service serving
Resilience	software bugs, infrastructural resource failures, fault-tolerance and replication for data services and processing (Yang et al., 2017)	bias in data, adversary attacks in ML (Katzir and Elovici, 2018), resilience learning (Fischer et al., 2018), computational Byzantine failures (Blanchard et al., 2017)
Elasticity	utilizing different data resources; increasing and decreasing data usage with respect to data volume, velocity and quality; elasticity of underling resources for data processing (Wang and Balazinska, 2017)	elasticity of resources for computing (Huang et al., 2015; Harlap et al., 2017; Gujarati et al., 2017), elasticity of model parameters; performance loss versus model accuracy; elastic model services for performance

Source:
https://www.researchgate.net/publication/341762862_R3E_-An_Approach_to_Robustness_Reliability_Resilience_and_Elasticity_Engineering_for_End-to-End_Machine_Learning_Systems

Robustness and elasticity engineering based on QoA

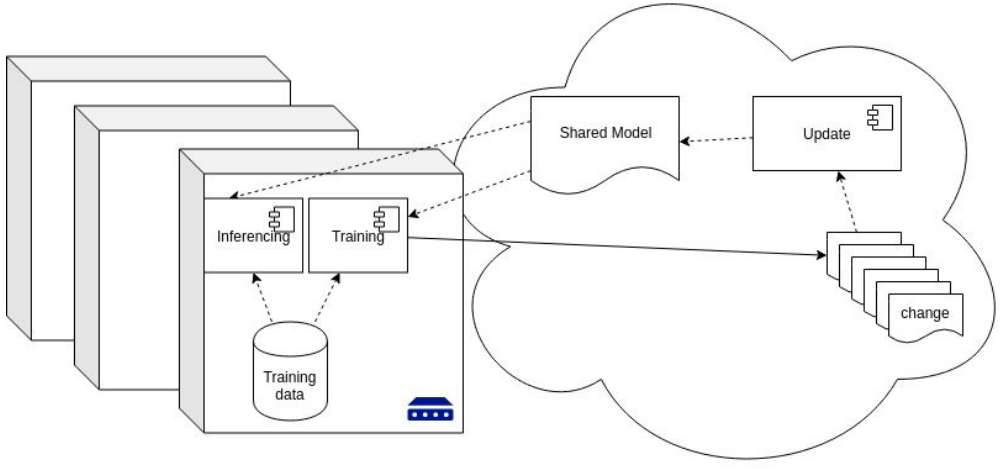
- **QoA-driven elasticity for robustness/ reliability**
- **Monitoring and coordination**



Hong-Linh Truong, Coordination-aware assurance for end-to-end machine learning systems: the R3E approach"

Federated/distributed training with edges

Decentralized with a distributed set of edge machines holding data and carrying out (sub) training/inferencing:

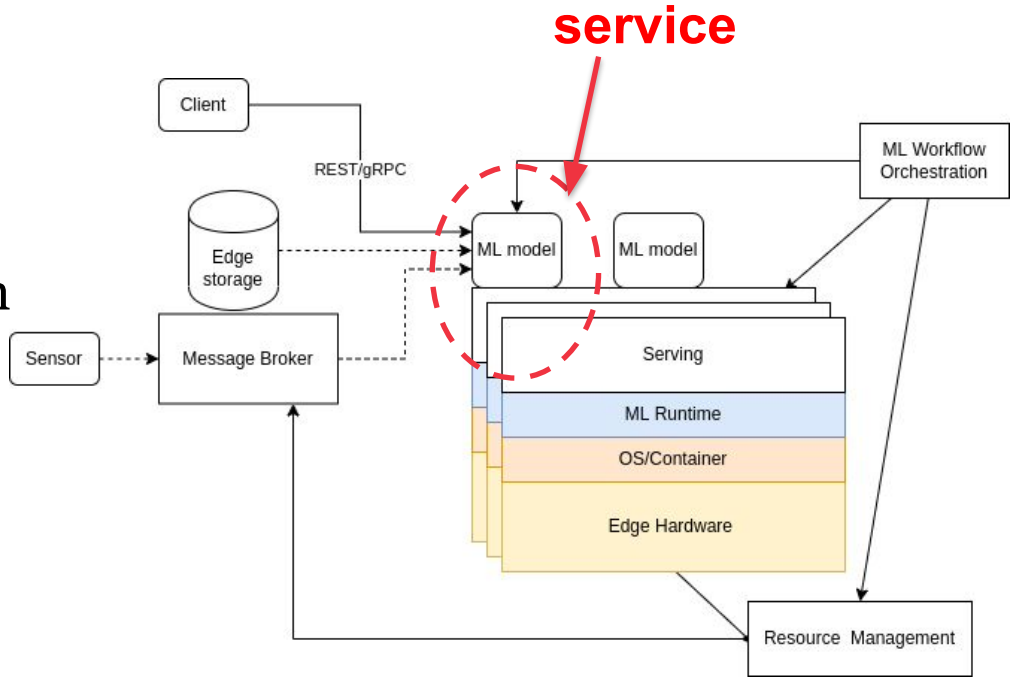


Benefit factors: data availability, privacy preservation, performance

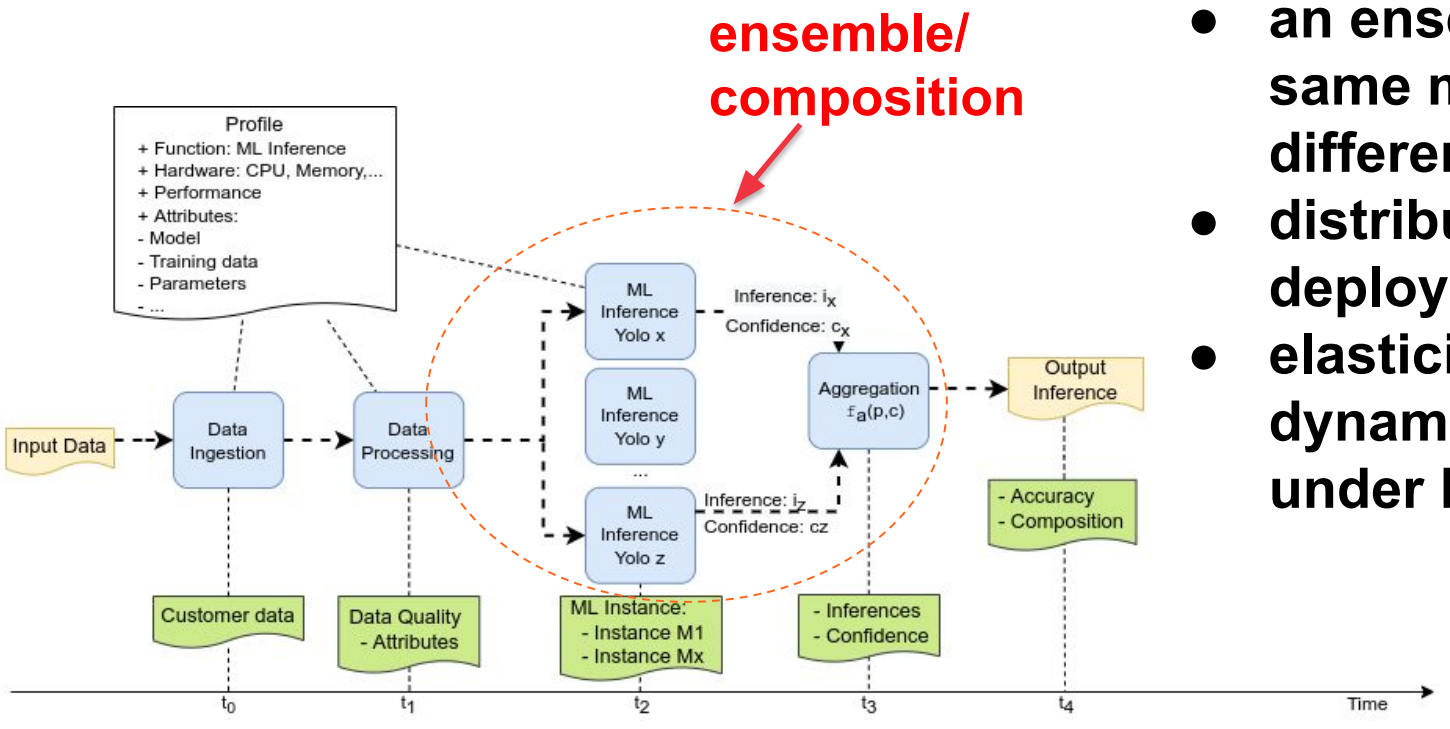
- **Challenges: performance, network, availability (due to sleeping, energy-saving) and trust**
- **What about R3E?**
  - consensus in updates, secured aggregation protocols, reliability and elasticity, and cost

Distributed ML serving

- **ML Serving (and R3E)**
  - how to distribute tasks in model serving?
  - how to partition ML tasks in edge and cloud?
  - how to deal with dynamic reconfigurations of models



Distributed ML serving: ensembles



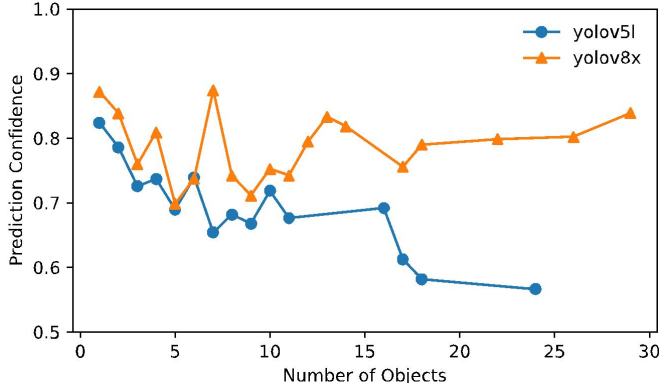
- an ensemble of the same models or different models
- distributed node deployment or not
- elasticity and dynamic control under R3E

Examples

QoA in Object Detection

- **Observing**: the influences of different factors on predictive performance of different ML models
- **Action:** update ML contracts with appropriate constraints to ensure service quality for ML consumers.

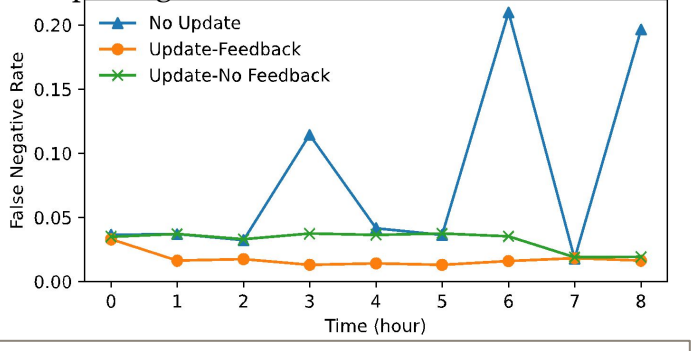
The relation between *confidence* of different ML models and *number of objects* in object detections



Continuous Optimization in Malware Detection

- **Observing:** ML models making false negative inference due to specific data quality (based on feedback from consumers)
- **Action:** adapt the ensemble to maintain service quality while dealing with data quality

False Negative Rate affected by continuously updating ensemble of ML models



Nguyen, Truong, & Truong-Huu. 2024. **Novel Contract-based Runtime Explainability Framework for End-to-End Ensemble Machine Learning Serving**. The IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI (CAIN '24).

Do we need to treat

Robustness, Reliability, Resilience, and Elasticity

equally in your design? from which views?

Study log for this week

Think about

- **What does it mean R3E for *YOUR data and machine learning systems*?**
- **Read one of the papers in the reading list for today lecture**

Then

- **in your experience/work, which ones of R3E concern you most? Why? What would you do? What do you look for?**
- **~1-2 page – submit it to the MyCourses for comments/feedback (keep it in your git)**

Thanks!

Hong-Linh Truong
Department of Computer Science

rdsea.github.io