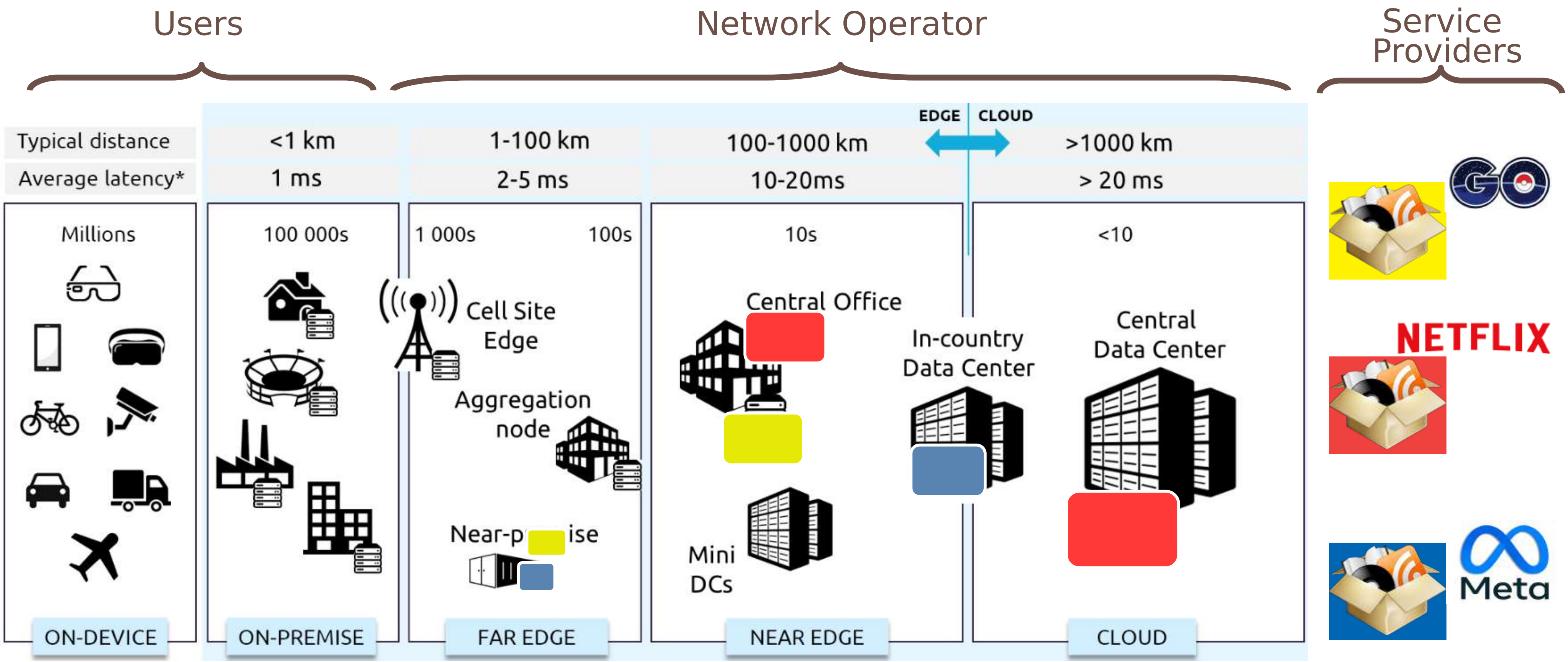
Planning and control of telecommunication and transport networks

Ass. Prof. *Andrea Araldo*
(Télécom SudParis)

Andrea Araldo

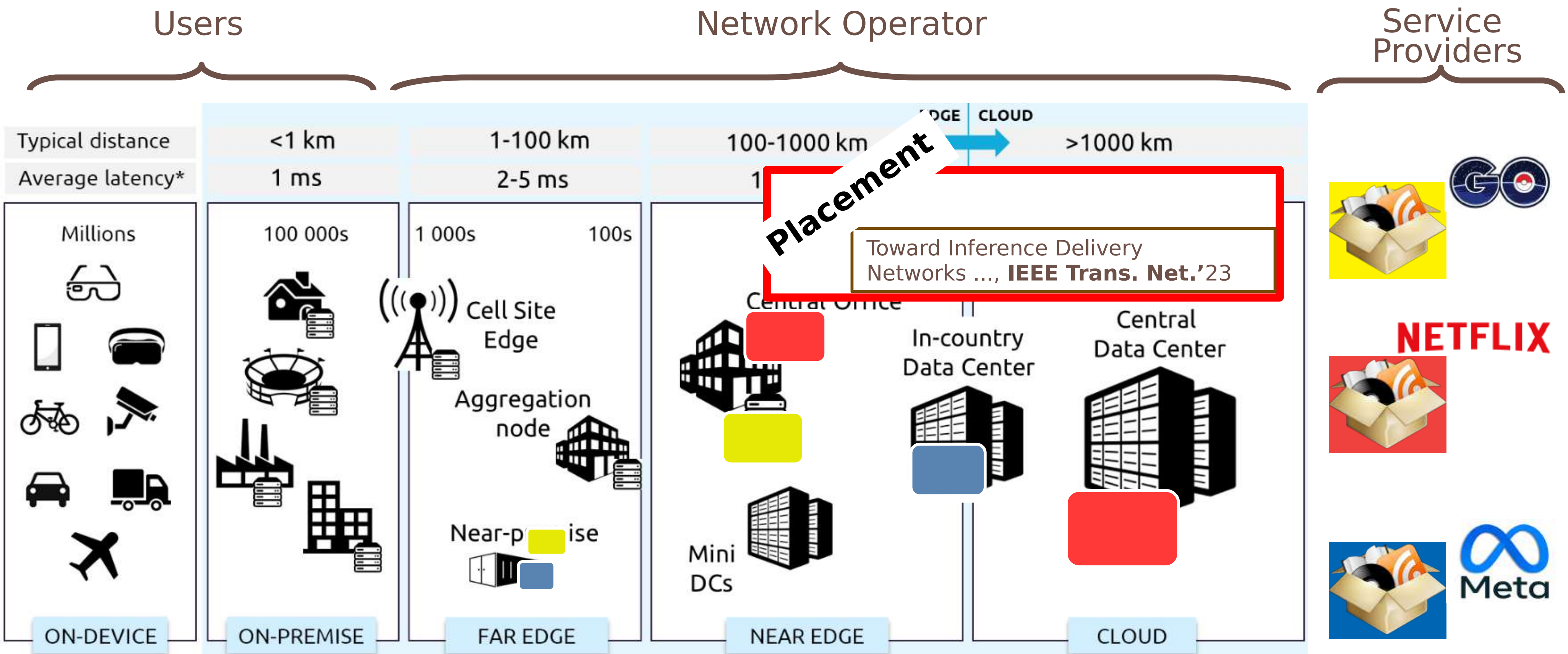
Cloud-to-edge continuum



Backgroung figure from European industrial technology roadmap for the next generation cloud-edge offering, p12, The European Commission, 2021, link

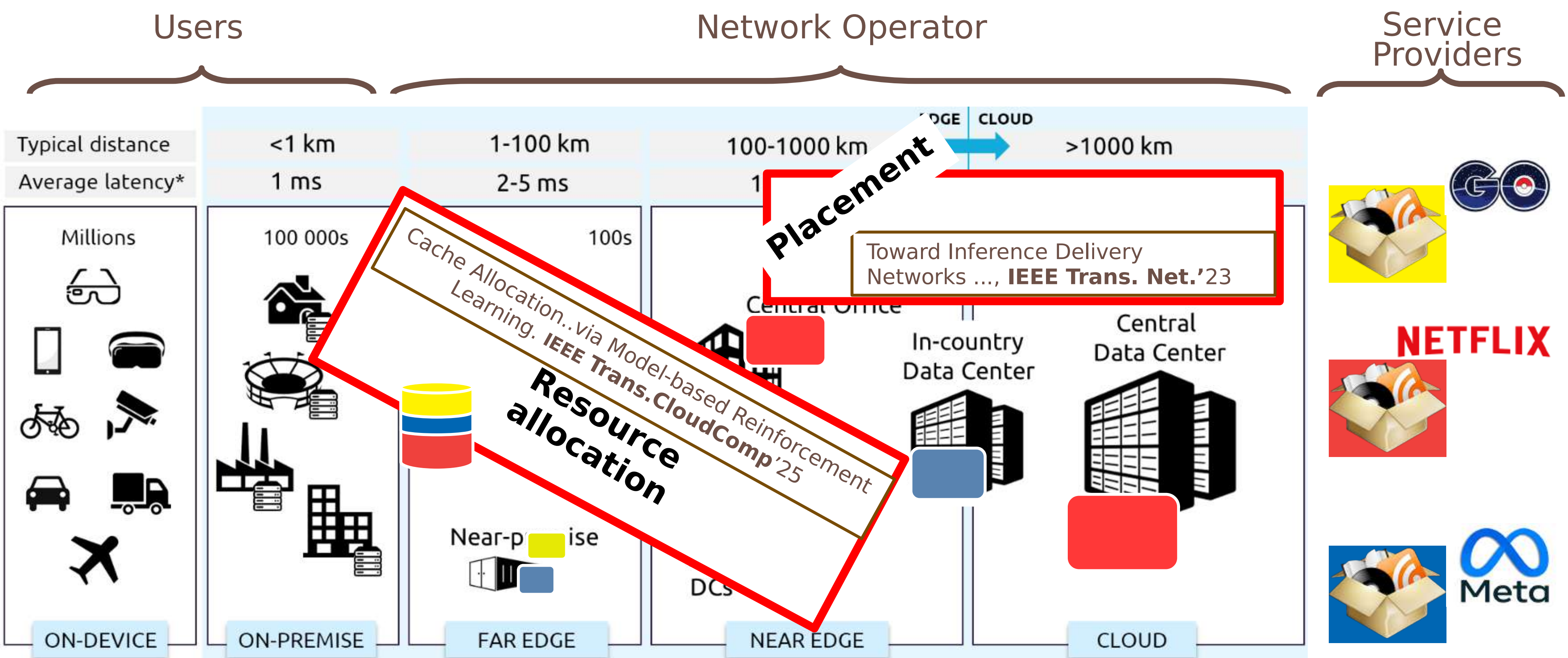
Andrea Araldo

Cloud-to-edge continuum



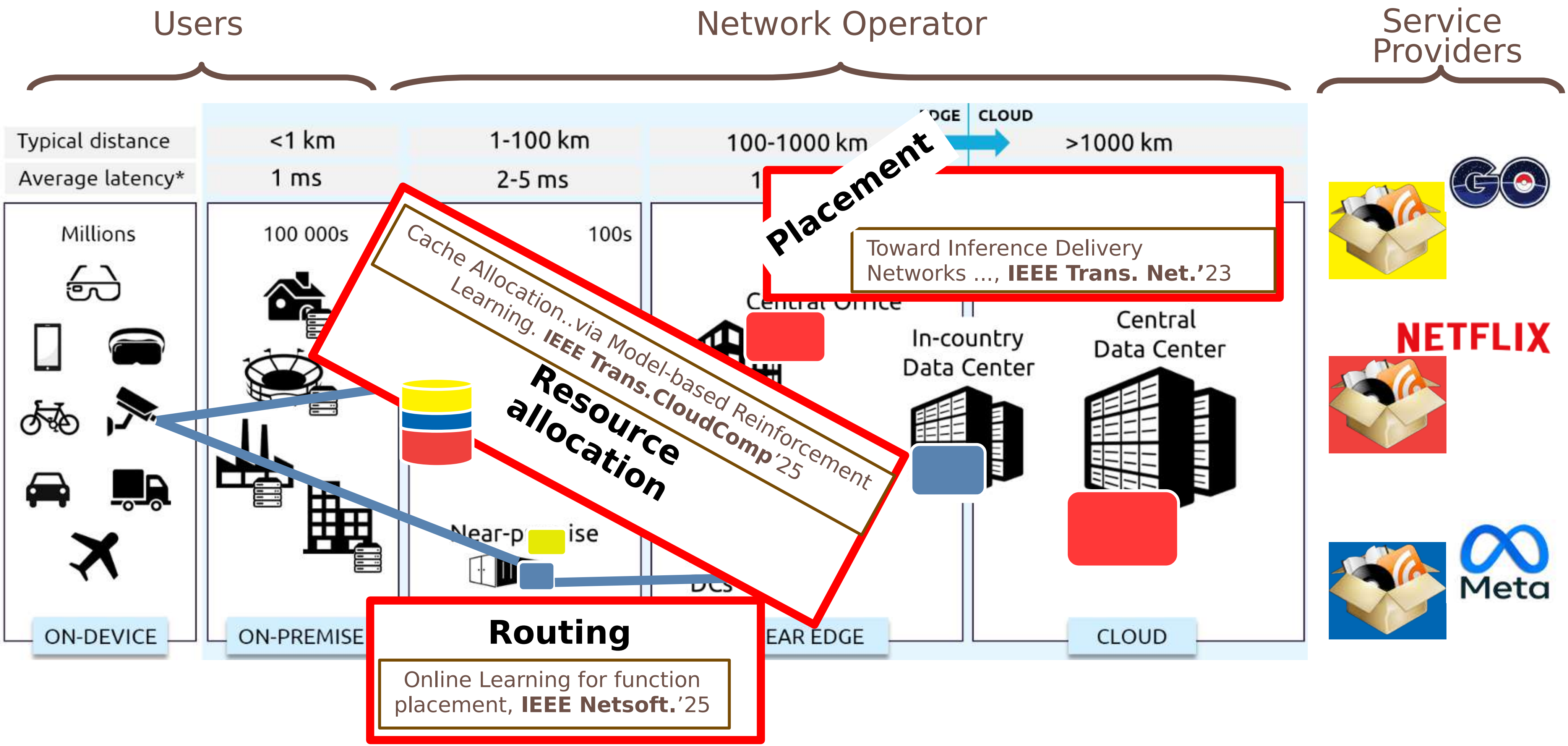
Backgroung figure from European industrial technology roadmap for the next generation cloud-edge offering, p12, The European Commission, 2021, link

Cloud-to-edge continuum



Backgroung figure from European industrial technology roadmap for the next generation cloud-edge offering, p12, The European Commission, 2021, link

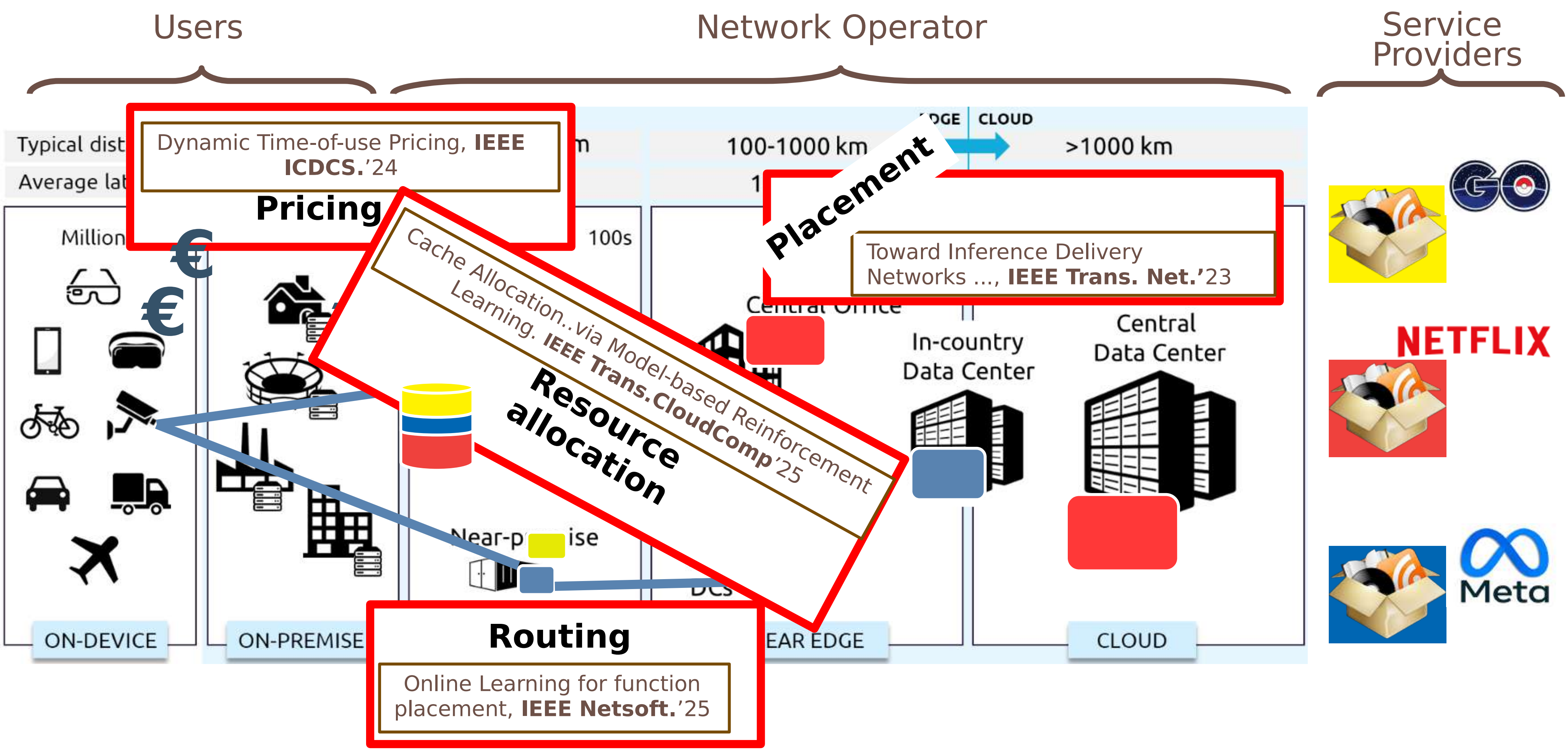
Cloud-to-edge continuum



Backgroung figure from European industrial technology roadmap for the next generation cloud-edge offering, p12, The European Commission, 2021, link

Andrea Araldo

Cloud-to-edge continuum



Backgroung figure from European industrial technology roadmap for the next generation cloud-edge offering, p12, The European Commission, 2021, link

Learning with no long training

No long training is possible in a real system

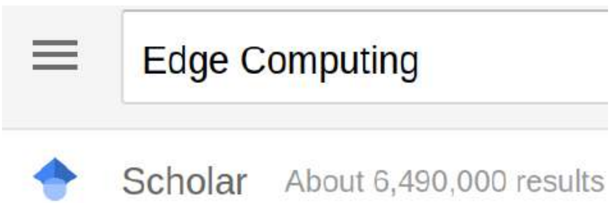
- Model-based q-learning[1]
  - Calibrate a ***model*** to match observations
  - ***Simulate*** x times more transitions than the real one
  - Guarantee: convergence in probability boundedly close to the optimum
- Generalized Hidden Parameter Markov Decision Processes[2]
  - 2 Bayesian Neural Networks
    - 1 to learn state transitions
    - 1 to learn reward
  - Train on different synthetic scenarios
    - Keep some parameters fixed when changing scenarios (general dynamics)
    - Let other parameters change at each scenario (scenario-specific dynamics)
  - When applied in a real case, only few parameters need to be adapted
  - Guarantee: none

- Online learning[3]
  - Take a placement action
  - Assume worst case for the requests
  - Estimate a subgradient
  - Apply gradient ascent
  - Guarantee: sublinear regret
- Multi-armed bandits[4]

  ~KL-UCB algorithm
  - Estimate bounds for the unknown parameters
  - Compute policy using bounds
  - Guarantee: sublinear regret

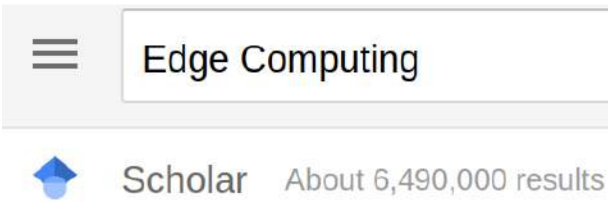
[1] Cache Allocation..via Model-based Reinforcement Learning. **IEEE Trans.CloudComp**'25
[2] Dynamic Time-of-use Pricing, **IEEE ICDCS.**'24
[3] Toward Inference Delivery Networks ..., **IEEE Trans. Net.'**23
[4] Online Learning for function placement, **IEEE Netsoft.**'25

Why is not Edge Computing implemented?



- Unprecedented **business opportunity** for network operators
  - Network operators own the Edge
  - Service Providers must pass through network operators to run at the edge

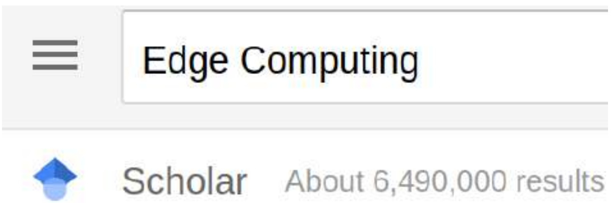
Why is not Edge Computing implemented?



- Unprecedented **business opportunity** for network operators
  - Network operators own the Edge
  - Service Providers must pass through network operators to run at the edge

But today, Edge Computing is not deployed

Why is not Edge Computing implemented?



- Unprecedented **business opportunity** for network operators
  - Network operators own the Edge
  - Service Providers must pass through network operators to run at the edge

But today, Edge Computing is not deployed

Network operators are reluctant to bear the high cost and risk all alone

Cooperative strategies for large technological infrastructures

- Actors
  - Network operators
  - Service Providers
- Decisions
  - Investment
    - Dimensioning resources
    - Timing of investments of each player
  - Sharing
    - of revenues, cost, risk
    - Dynamic resource allocation
- Questions
  - Is the coalition *stable*?
  - Is the coinvestment *profitable*?
- Uncertainty
  - User engagement, energy and resource availability are random processes
  - Bounds on probability of stability and profitability
- Aim
  - Propose multi-agent decision strategies ensuring stability or profitability with at least 99% probability
- Coalitional (stochastic, robust) game theory

Sakr, Araldo, Chahed, Patanè & Kofman . Coalitional game-theoretical approach to coinvestment with application to edge computing. **IEEE ICC** 2025

Sakr, Chahed, Patanè & Kofman. Co-Investment under Revenue Uncertainty Based on Stochastic Coalitional Game Theory, major revision in The Annals of Operations Research, 2026

Sustainability

- How to reduce the environmental footprint of the computation continuum cloud→ edge→ devices?
- Moving computation close to users requires hardware

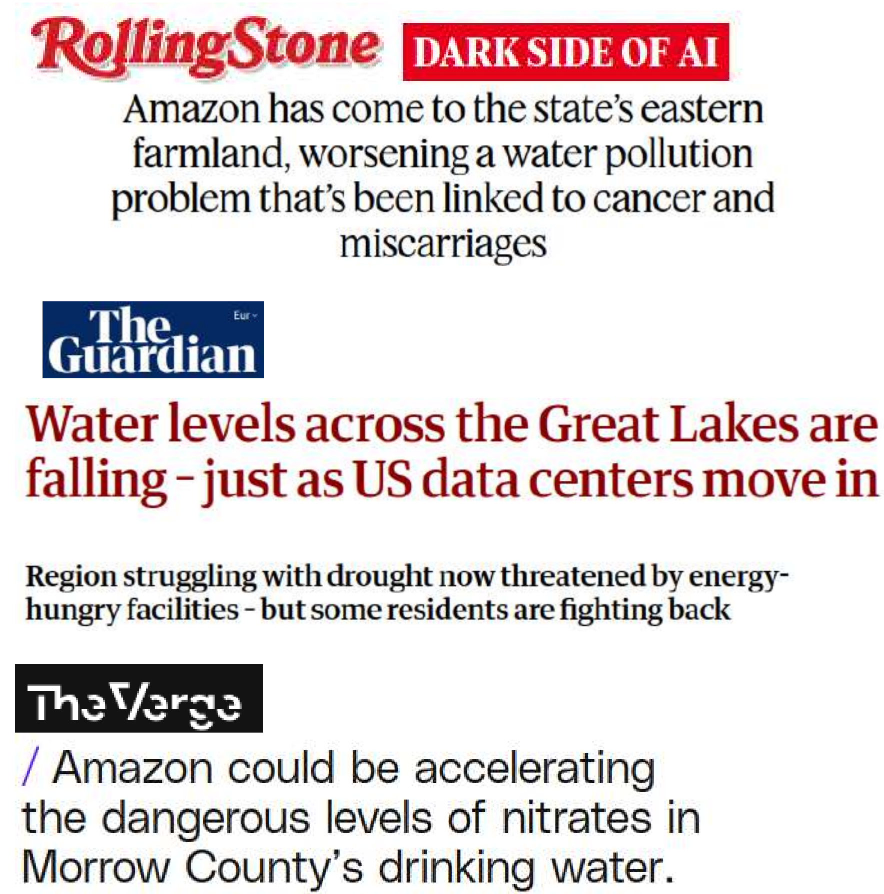
  -- Externality for its production, depletion of rare resources

  +Potential proximity to renewable energy
- Moving computation to the cloud

  -- Concentration of externalities

  -- Huge heating requirements

  +Consolidation
- Offload on available computation resources whenever possible
  - Vehicles,[1] network edge nodes, …


RollingStone DARK SIDE OF AI

Amazon has come to the state's eastern farmland, worsening a water pollution problem that's been linked to cancer and miscarriages

The Guardian

Water levels across the Great Lakes are falling – just as US data centers move in

Region struggling with drought now threatened by energy-hungry facilities – but some residents are fighting back

The Verge

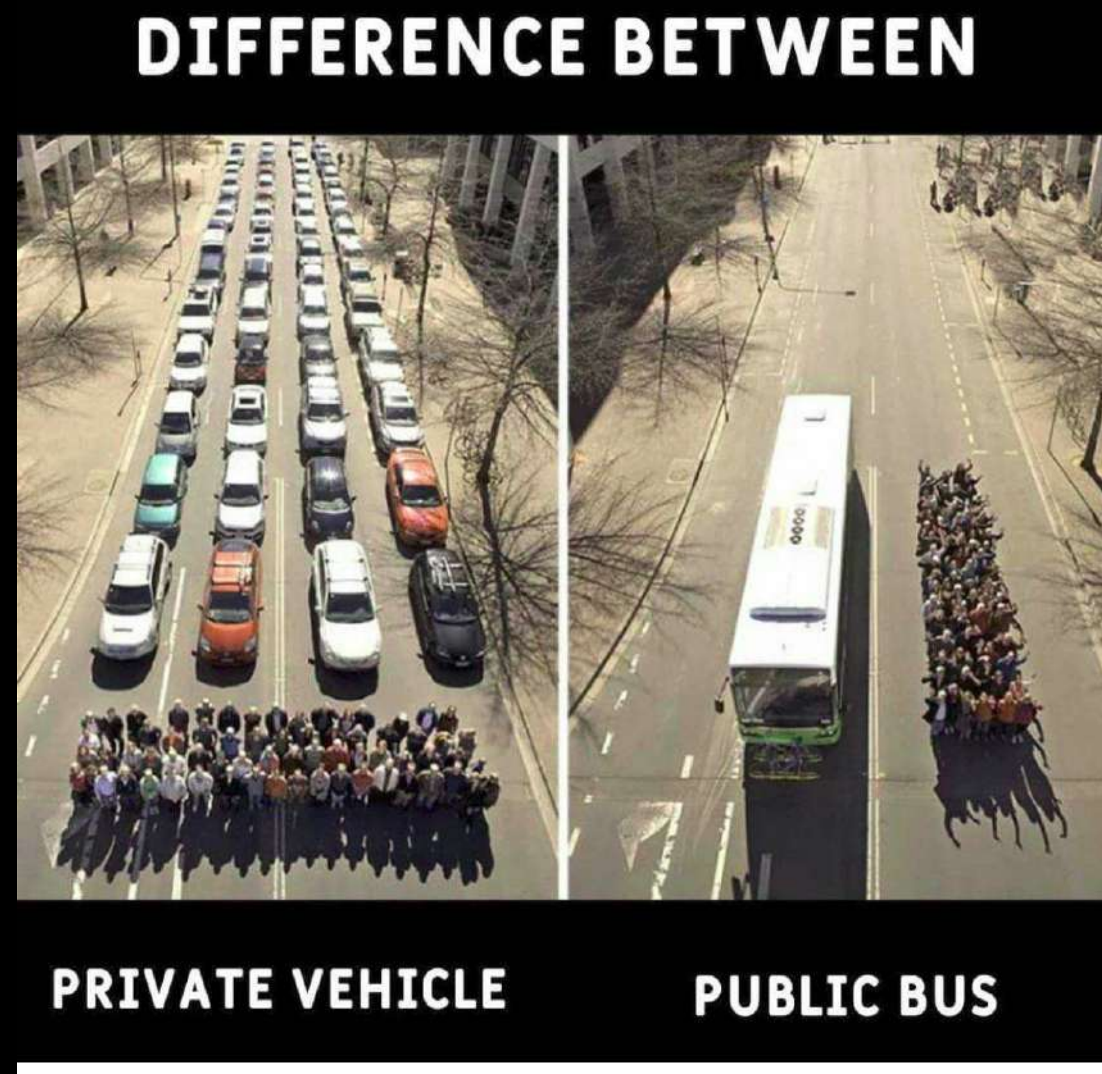
/ Amazon could be accelerating the dangerous levels of nitrates in Morrow County's drinking water.

Patané, Araldo, Achir, Boukhatem, Vehicular Cloud Computing: A cost-effective alternative to Edge Computing in 5G networks, **Computer Networks**, 2025

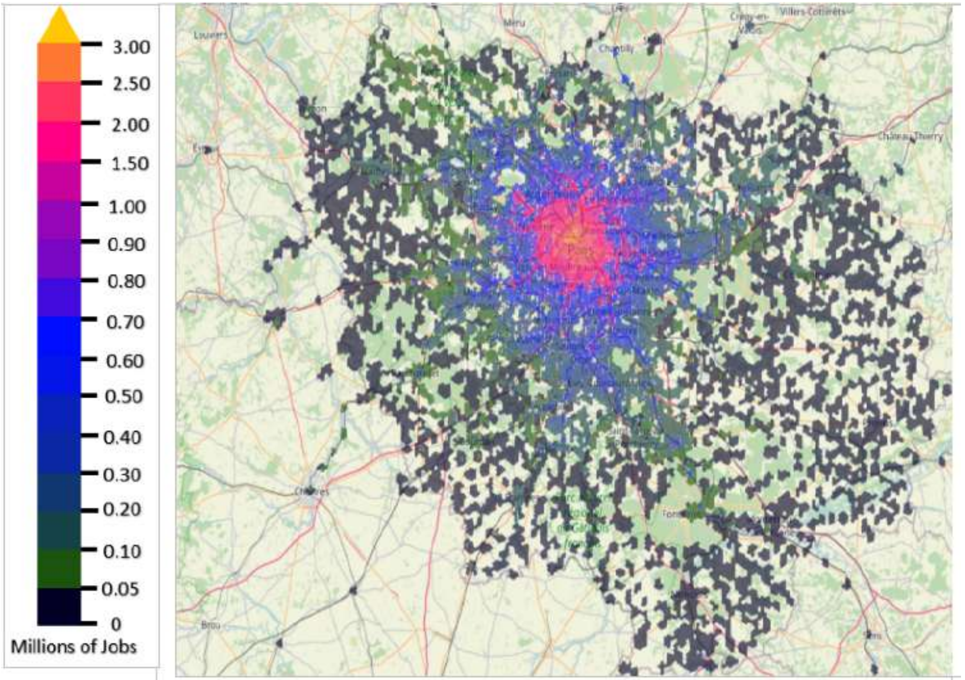




link



Inequity of accessibility distribution



[1] A. Badeanlou, **A. Araldo**, M. Diana, *Assessing transportation accessibility equity via open data*, subm. to hEART'22
[2] Biazzo et al. (2019). General scores for accessibility and inequality measures in urban areas. Royal Society Open Science

Inequity of accessibility distribution

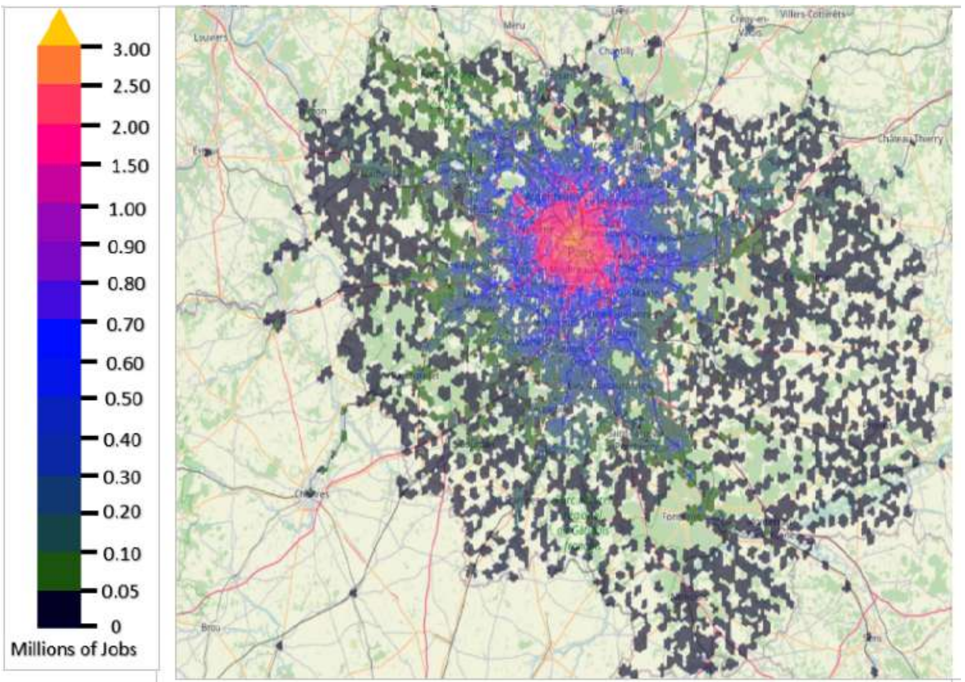
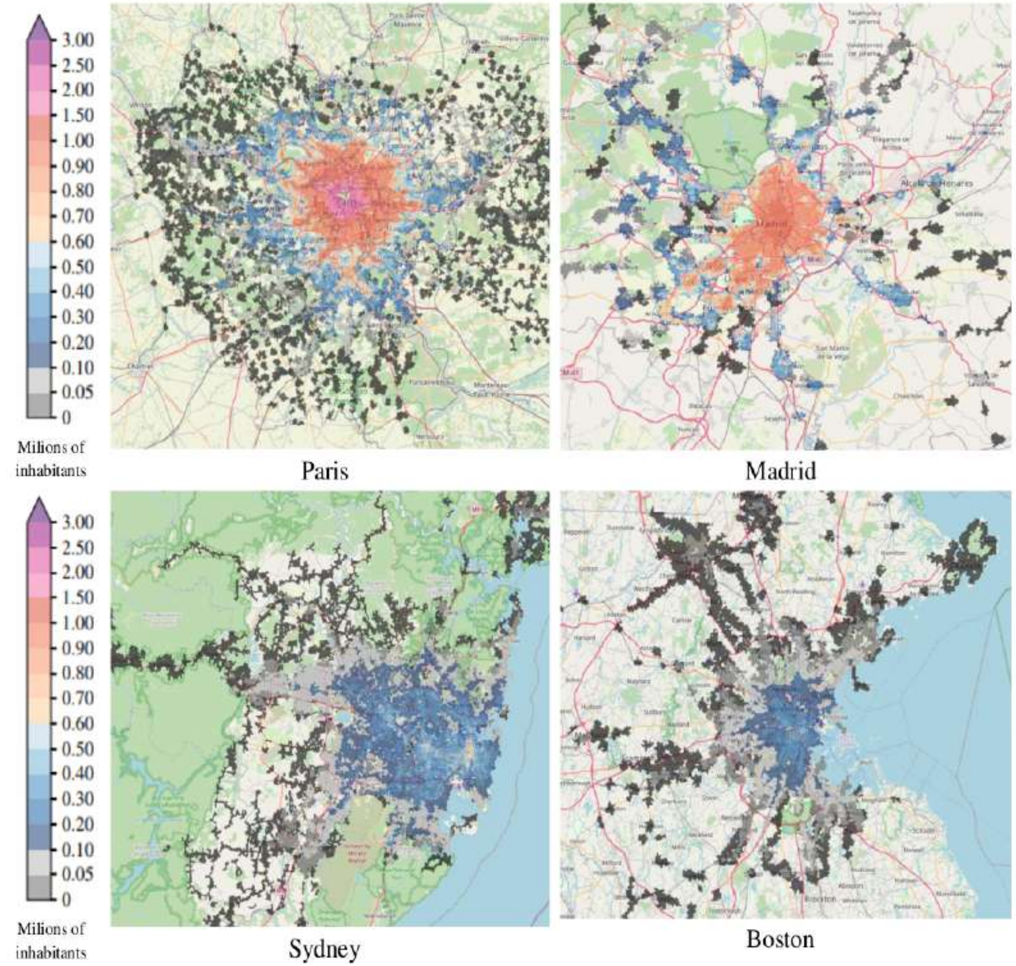


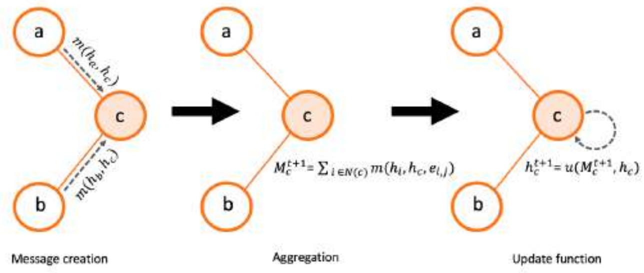
Figure 4-8 Sociality Score of four considered cities (Millions of Inhabitants)



[1] A. Badeanlou, **A. Araldo**, M. Diana, *Assessing transportation accessibility equity via open data*, subm. to hEART'22
[2] Biazzo et al. (2019). General scores for accessibility and inequality measures in urban areas. Royal Society Open Science

More equitable Public Transport

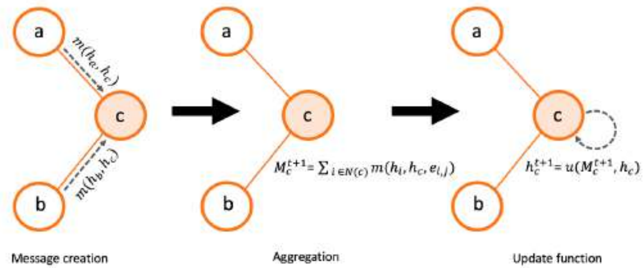
- Redesign bus lines for equality



Message Passing Neural Network (MPNN)

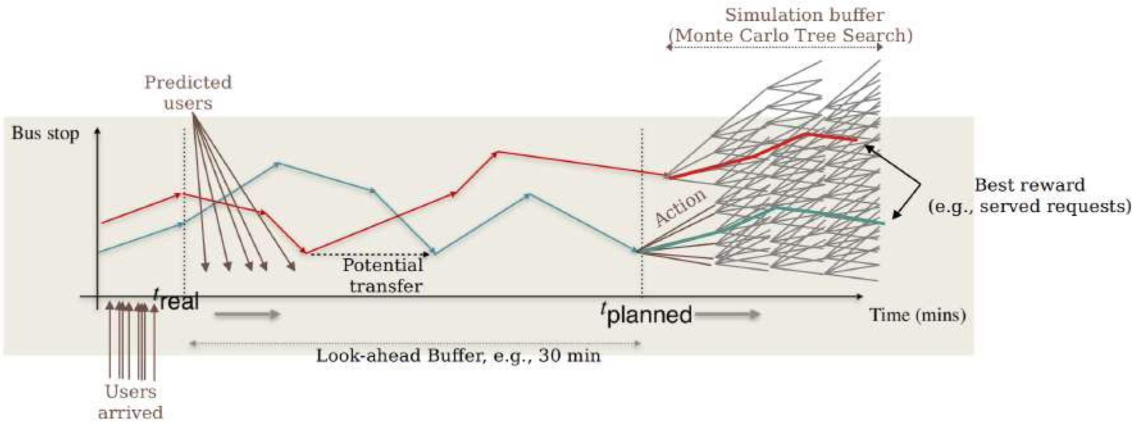
More equitable Public Transport

- Redesign bus lines for equality



Message Passing Neural Network (MPNN)

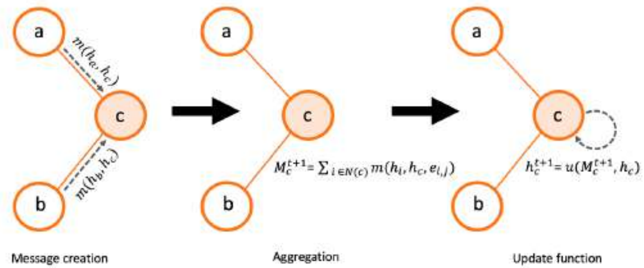
- Design a bus network in real time



Andrea Araldo

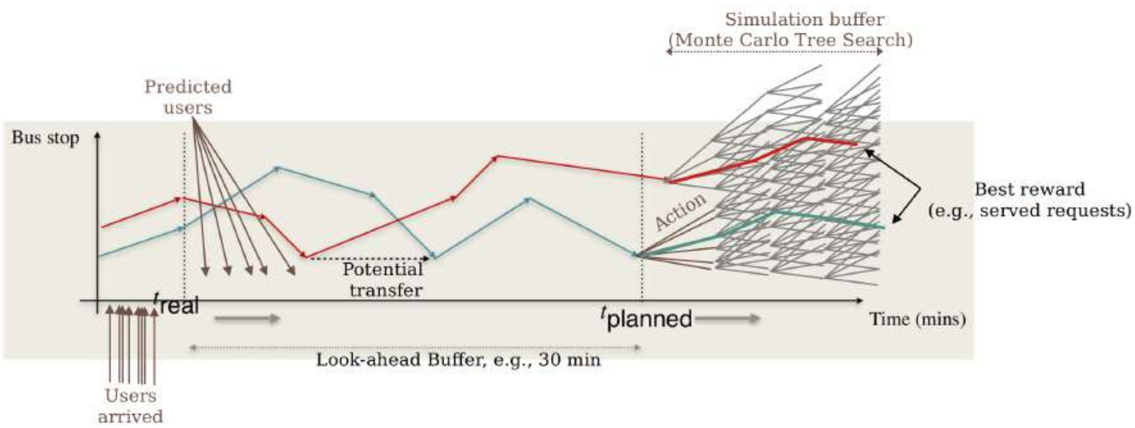
More equitable Public Transport

- Redesign bus lines for equality



Message Passing Neural Network (MPNN)

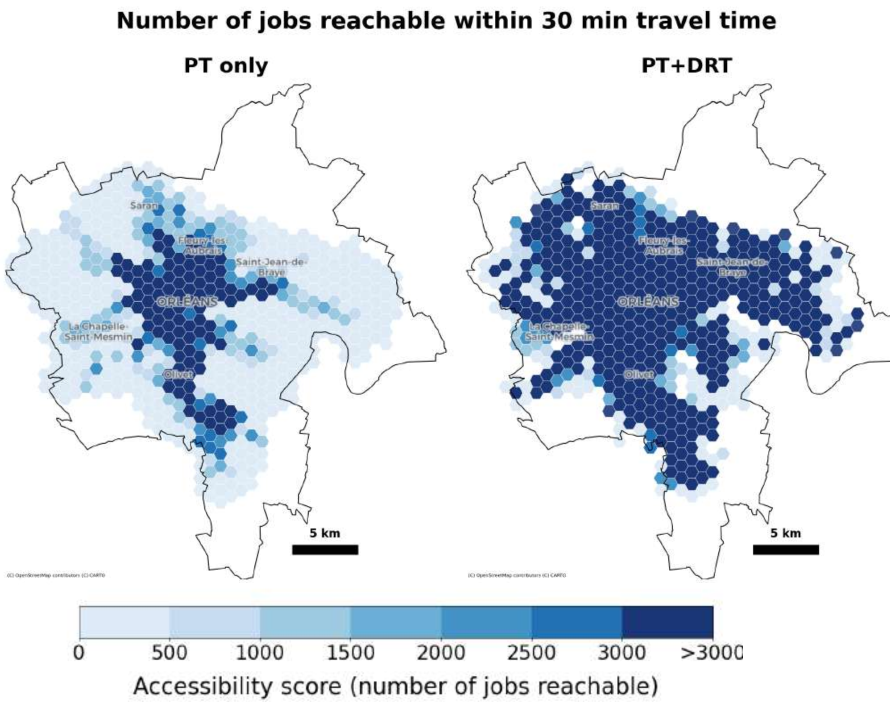
- Design a bus network in real time



- Mobility on Demand
  - Estimate accessibility
  - Plan for equality
  - ?Stochastic geometry for strategic-level modeling?

PADAM MOBILITY

SIEMENS



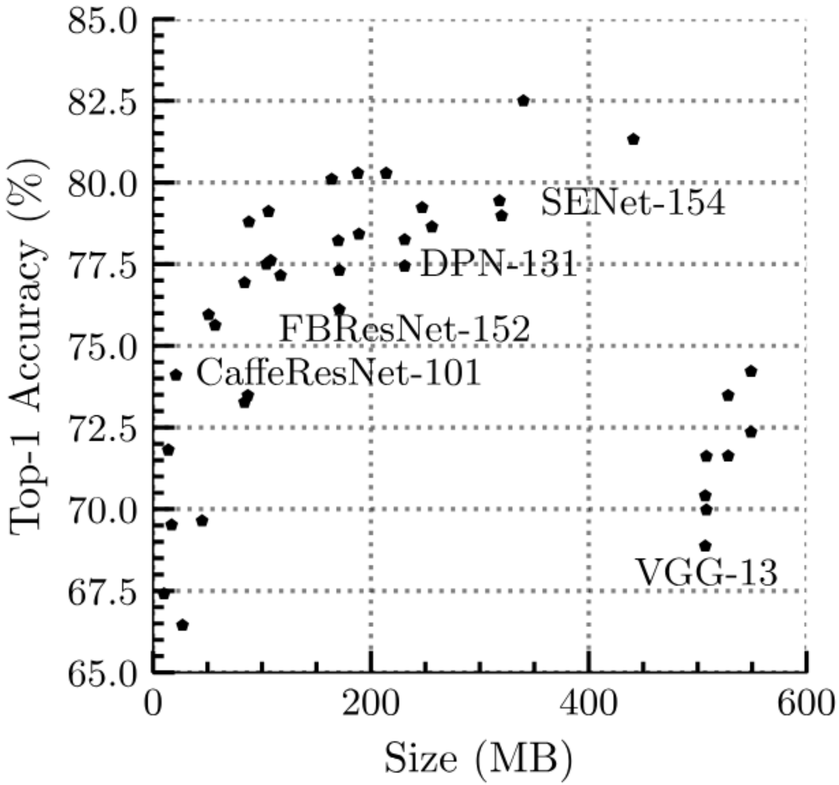
Backup

Toward **Inference Delivery Networks**: Distributing Machine Learning with Optimality Guarantees

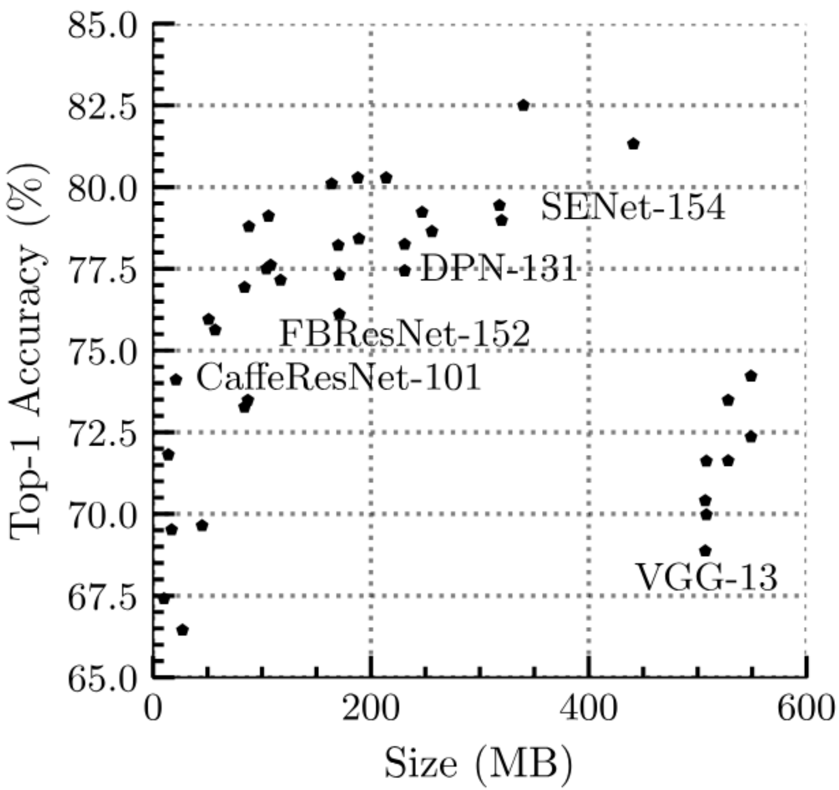
IEEE Transactions on Networking 2023

Tareq Salem, Gabriele Castellano, Giovanni Neglia, Fabio Pianese, Andrea Araldo

Motivation


85.0
82.5
80.0
77.5
75.0
72.5
70.0
67.5
65.0
Top-1 Accuracy (%)
SENet-154
DPN-131
FBResNet-152
CaffeResNet-101
VGG-13
0
200
400
600
Size (MB)

Motivation



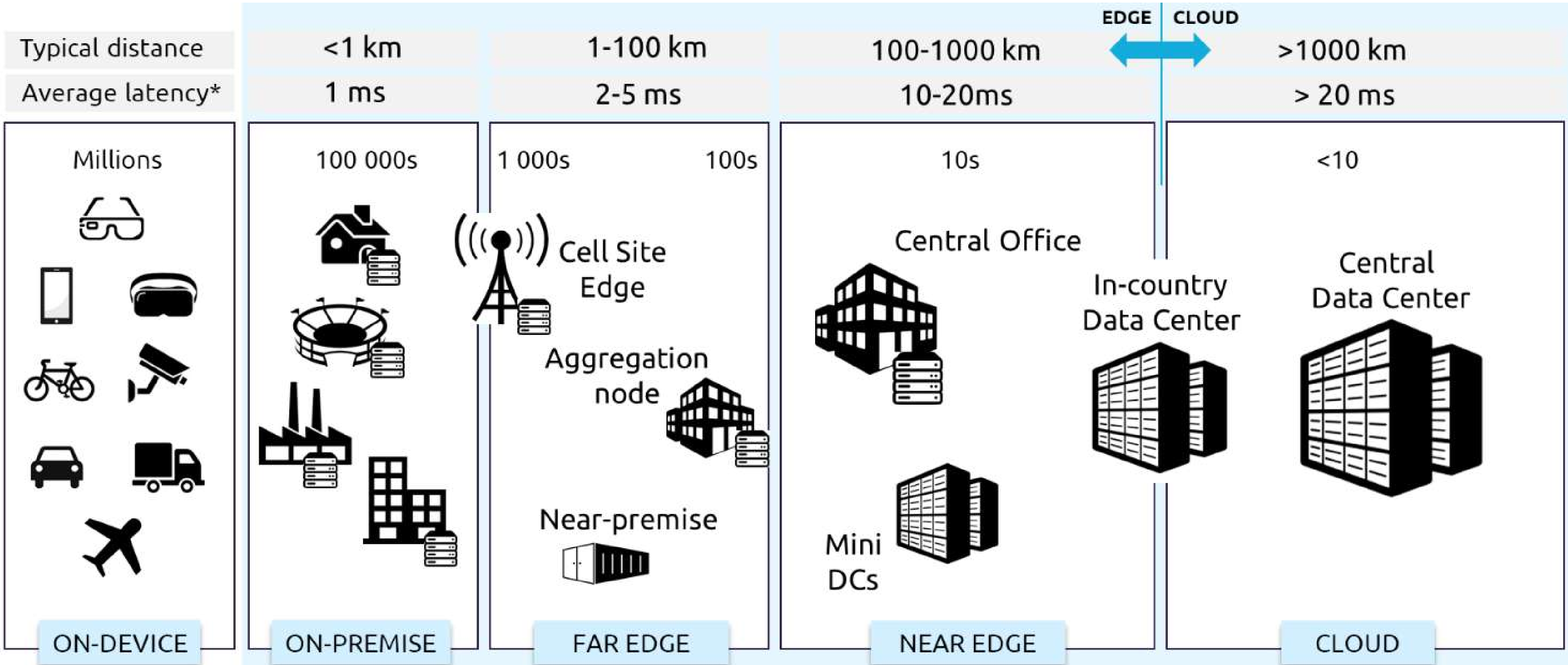
Research question:

In the cloud-to-edge continuum, **where** to place models and **which version**?

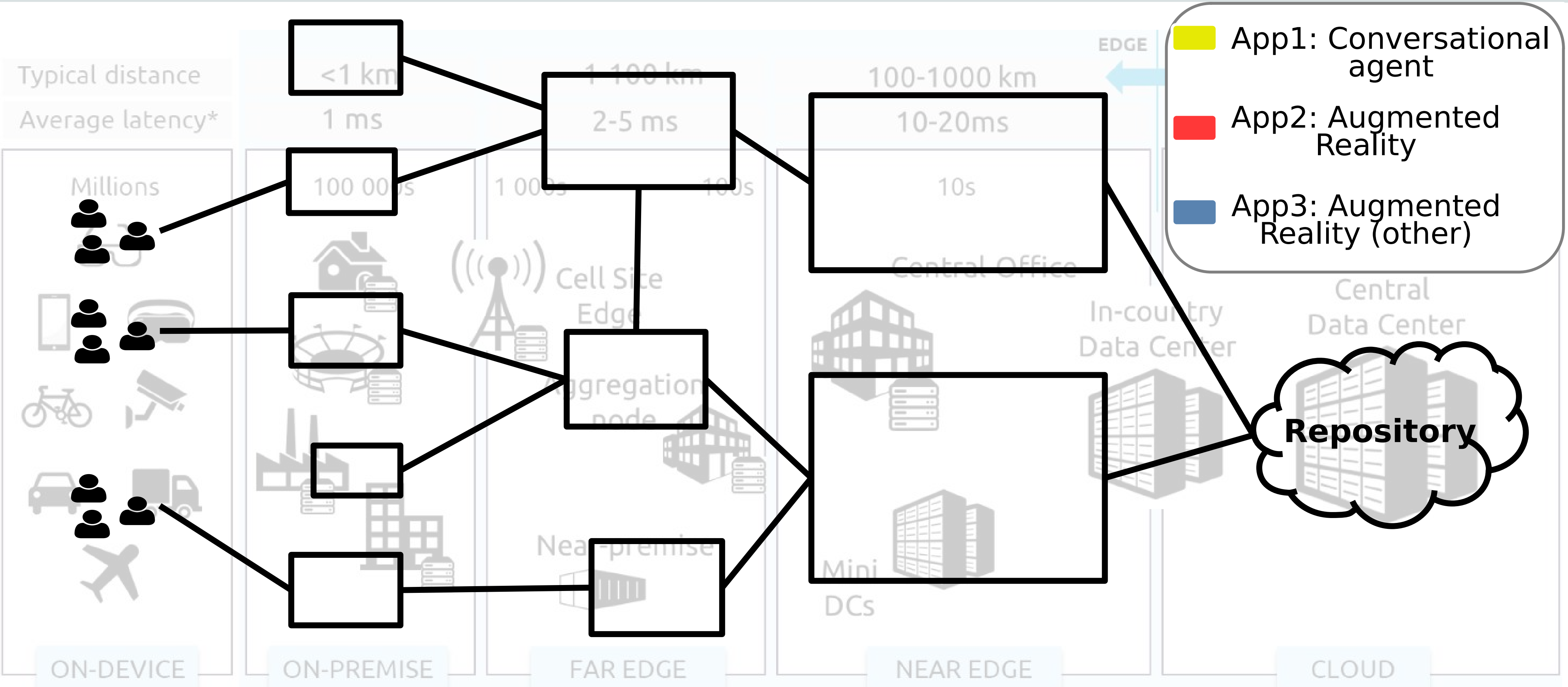
Novel idea:

~~Content~~ Inference Delivery Networks

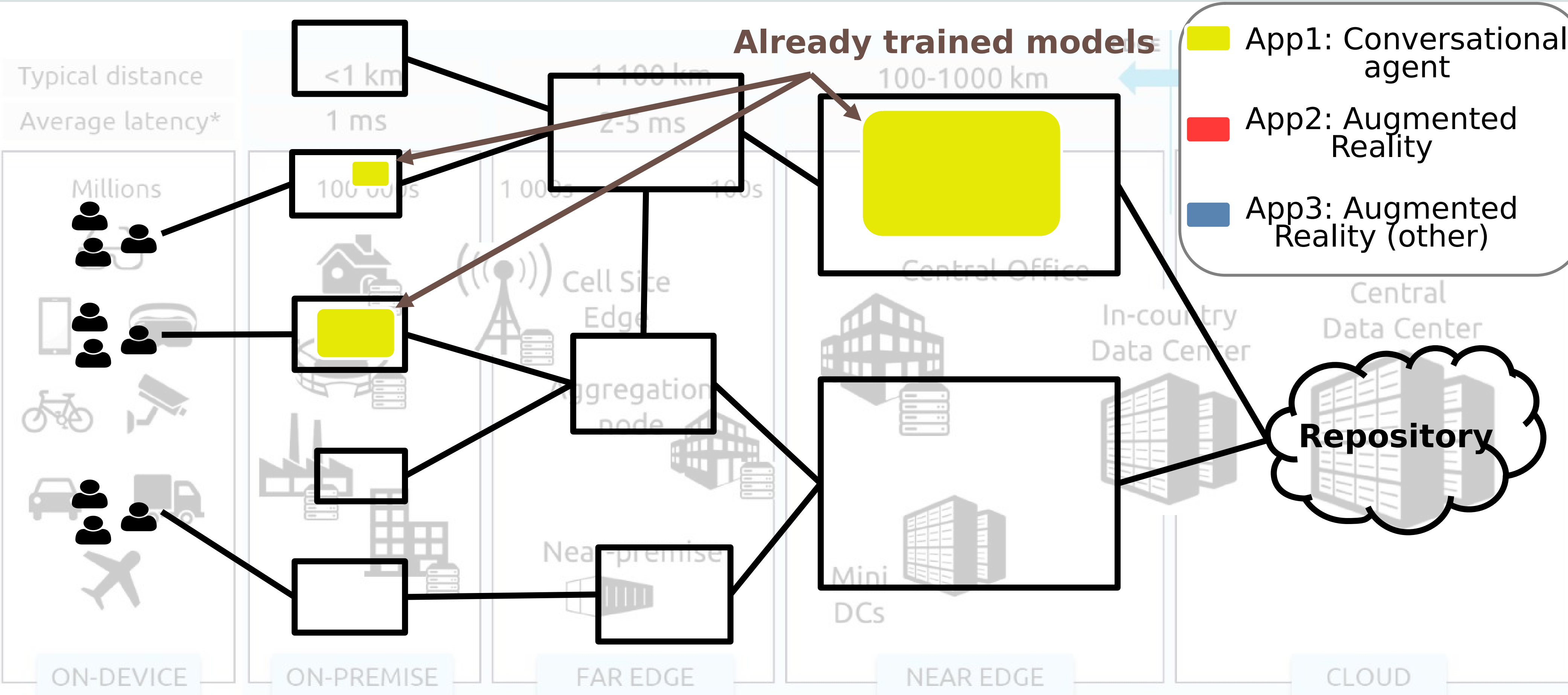
Problem: Inference Delivery Networks



Problem: Inference Delivery Networks



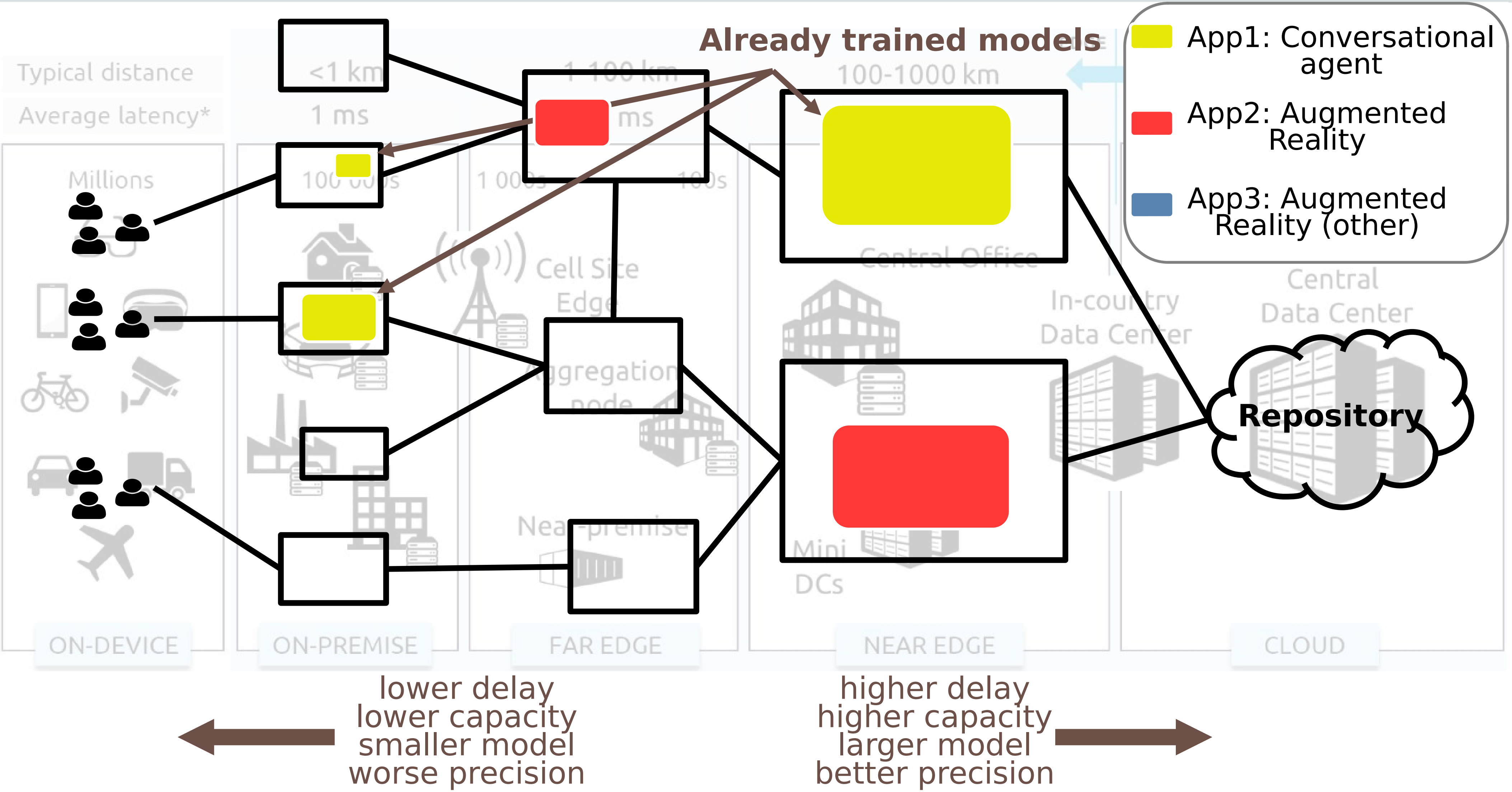
Problem: Inference Delivery Networks



lower delay
lower capacity
smaller model
worse precision

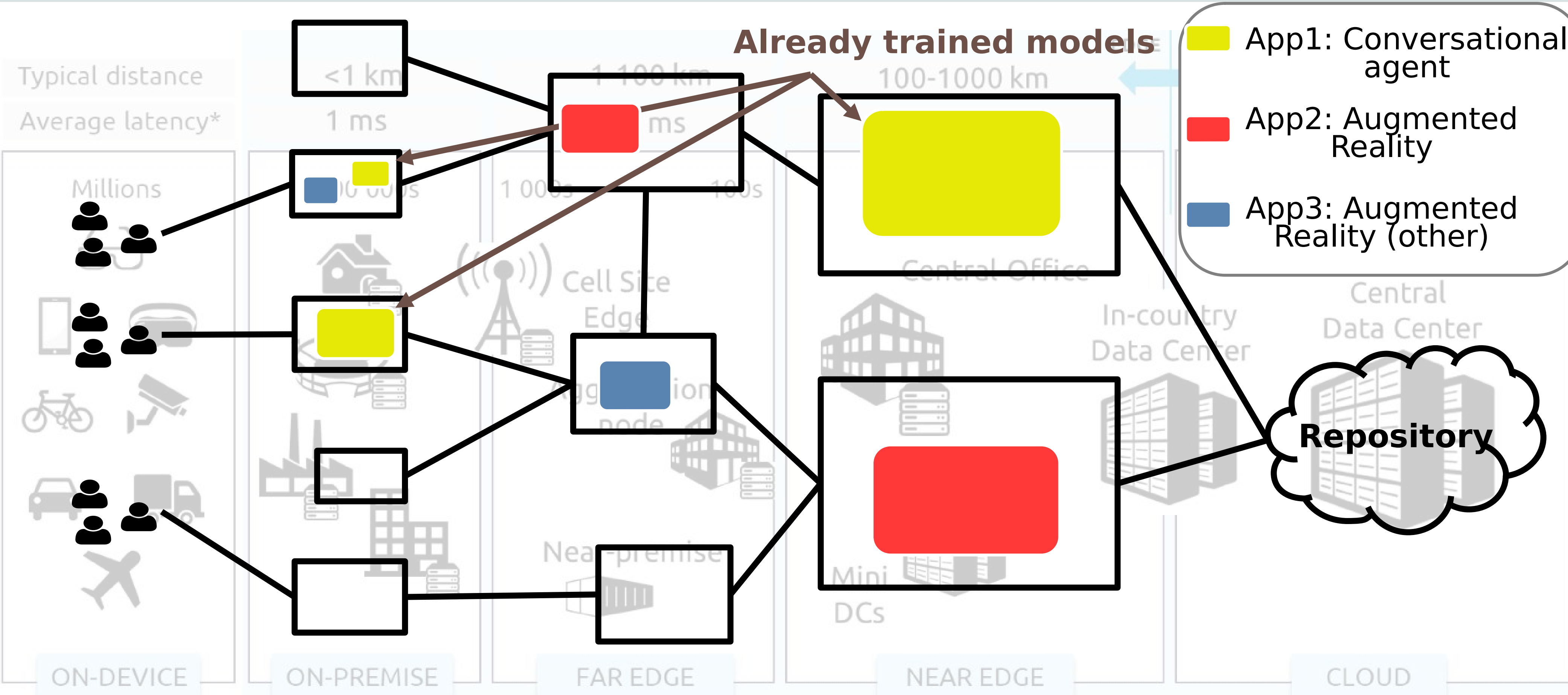
higher delay
higher capacity
larger model
better precision

Problem: Inference Delivery Networks



Andrea Araldo

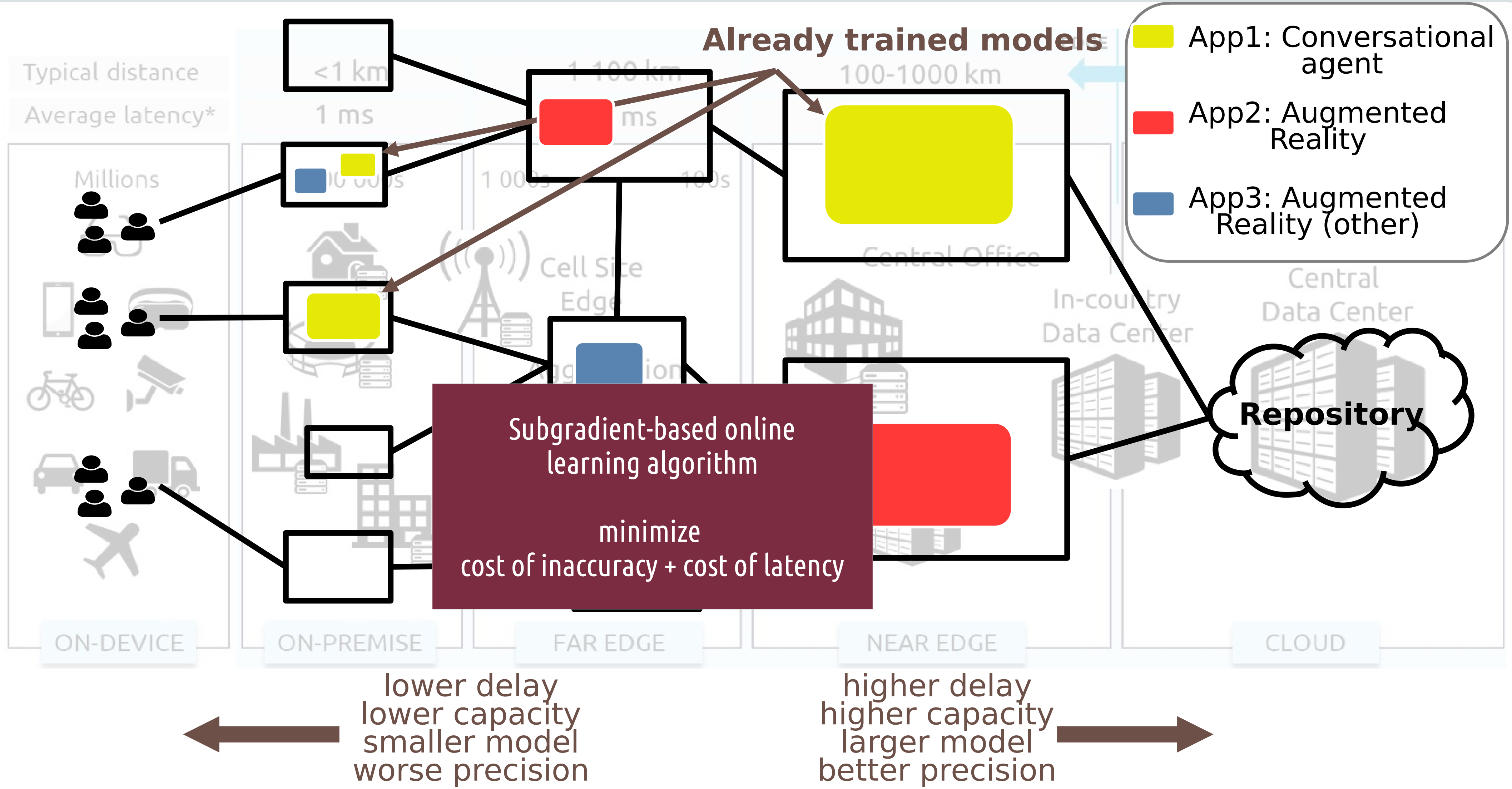
Problem: Inference Delivery Networks



lower delay
lower capacity
smaller model
worse precision

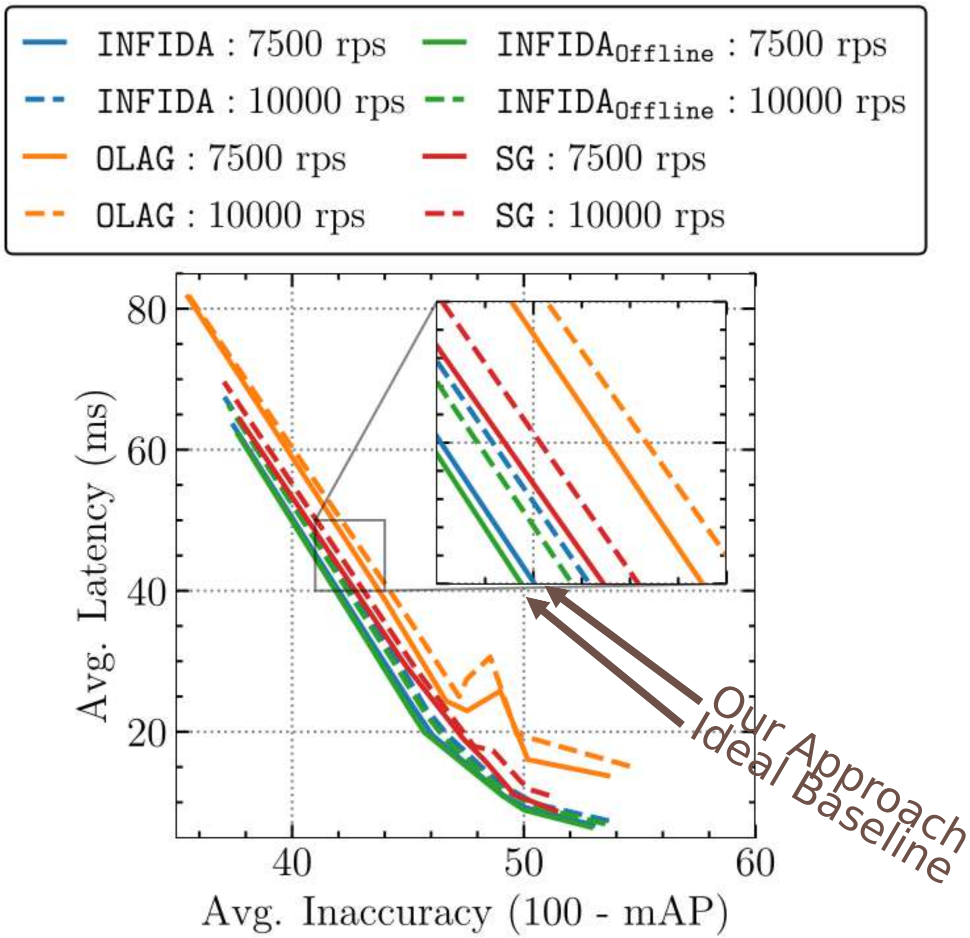
higher delay
higher capacity
larger model
better precision

Problem: Inference Delivery Networks



Results

Theorem
The regret
(worst-case deviation from optimum)
grows as √T



mAP: mean average precision (area under the precision-recall curve averaged across all classes)

Ideal baselines:

- INFIDA$_{offline}$: ideal basecase (in hindsight)
- SG: static greedy
- OLAG: online greedy

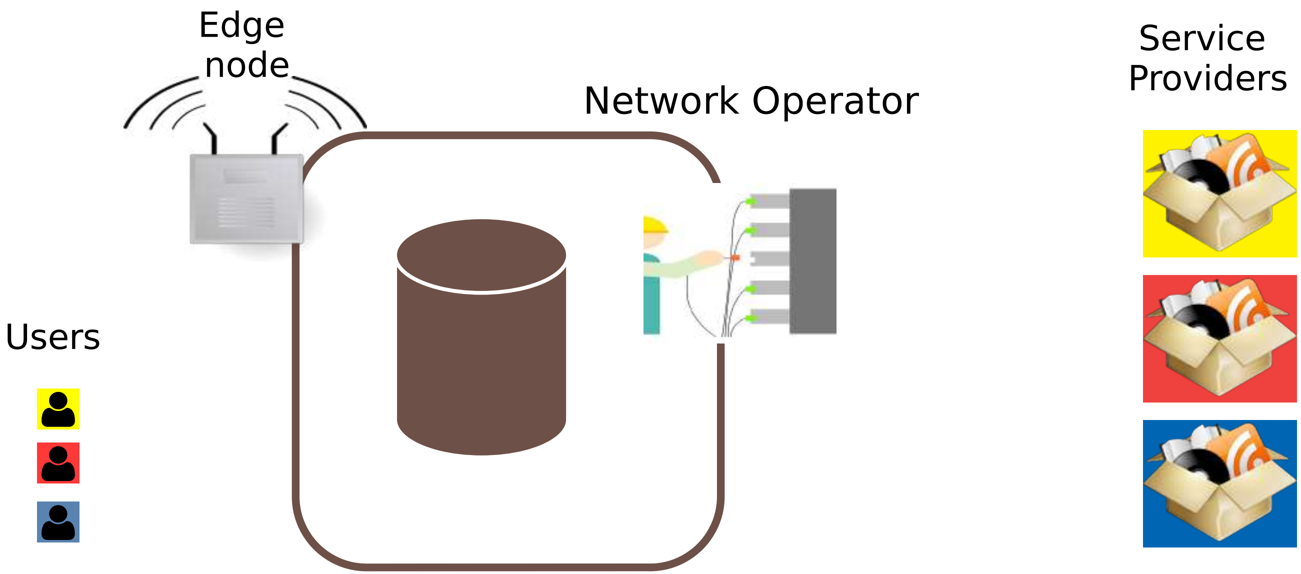
Although request arrival is not known our performance is close to that of an ideal omniscient decision-maker

Cache Allocation in Multi-tenant Edge Computing: an Online Model-based Reinforcement Learning Approach

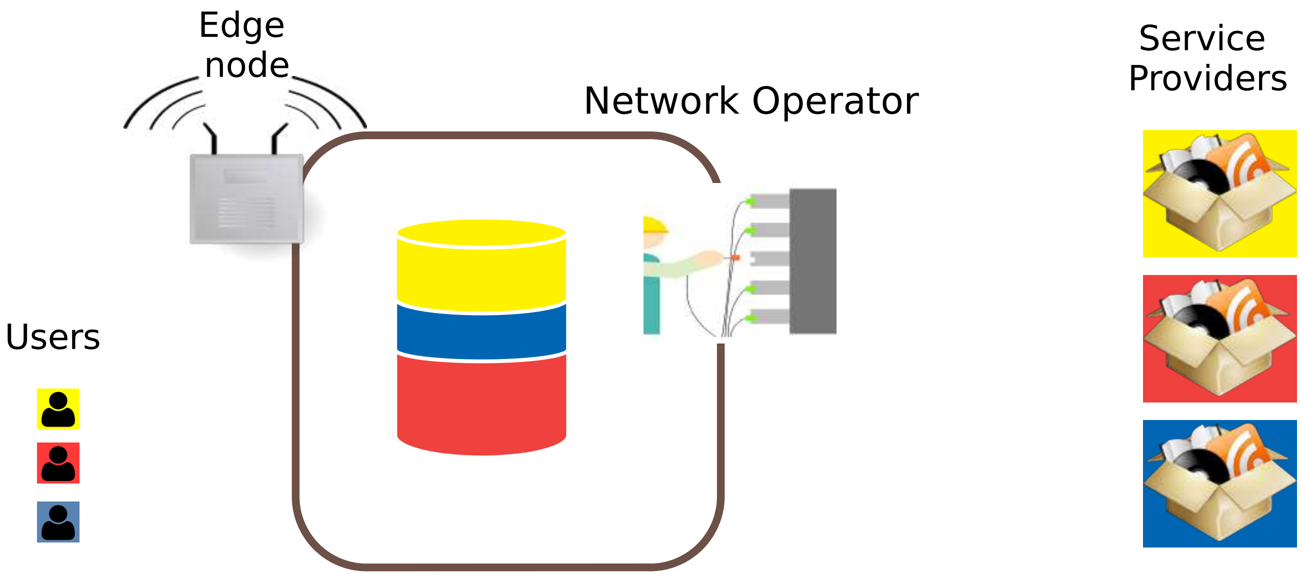
IEEE Transactions on Cloud Computing 2025

Ayoub Ben-Ameur, Andrea Araldo, Tijani Chahed

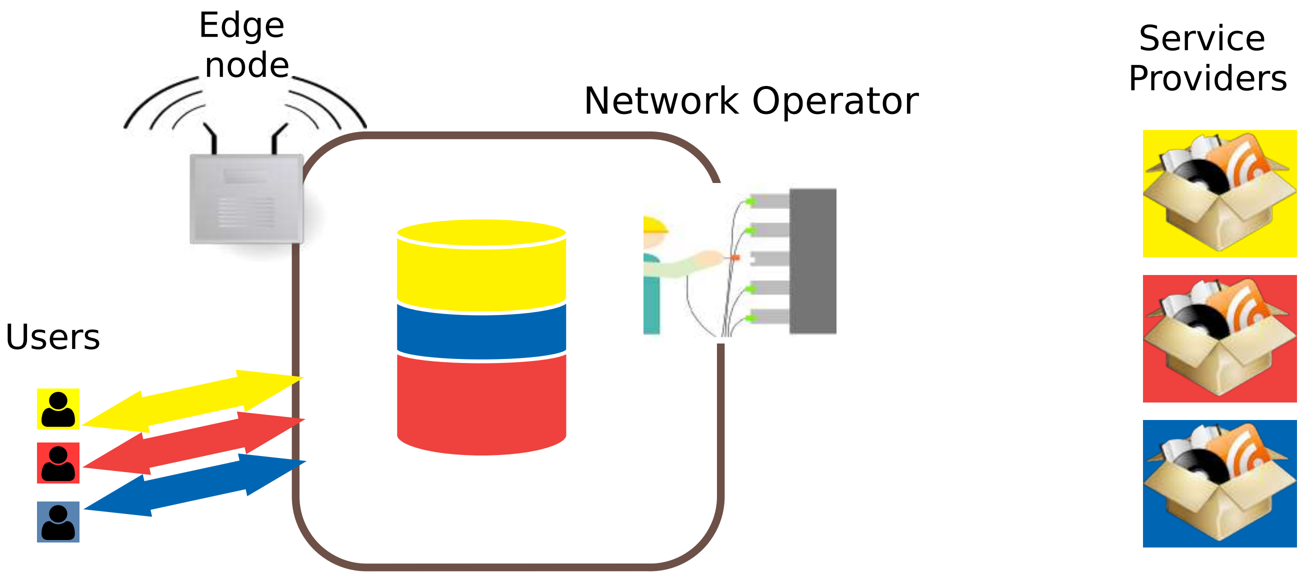
Resource allocation at the Edge



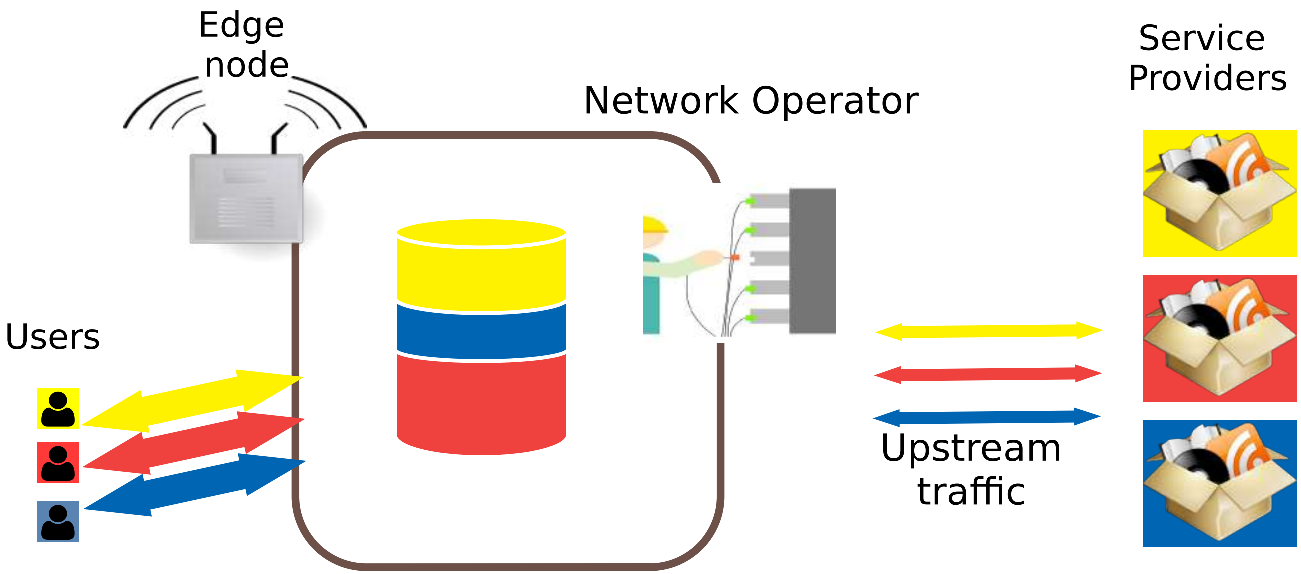
Resource allocation at the Edge



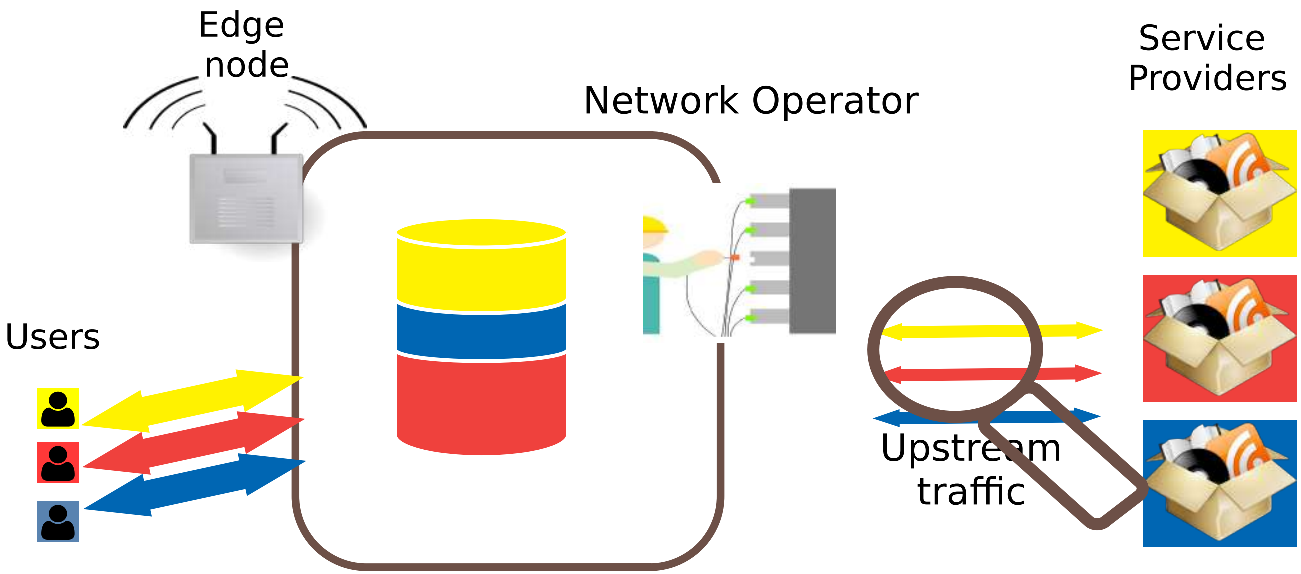
Resource allocation at the Edge



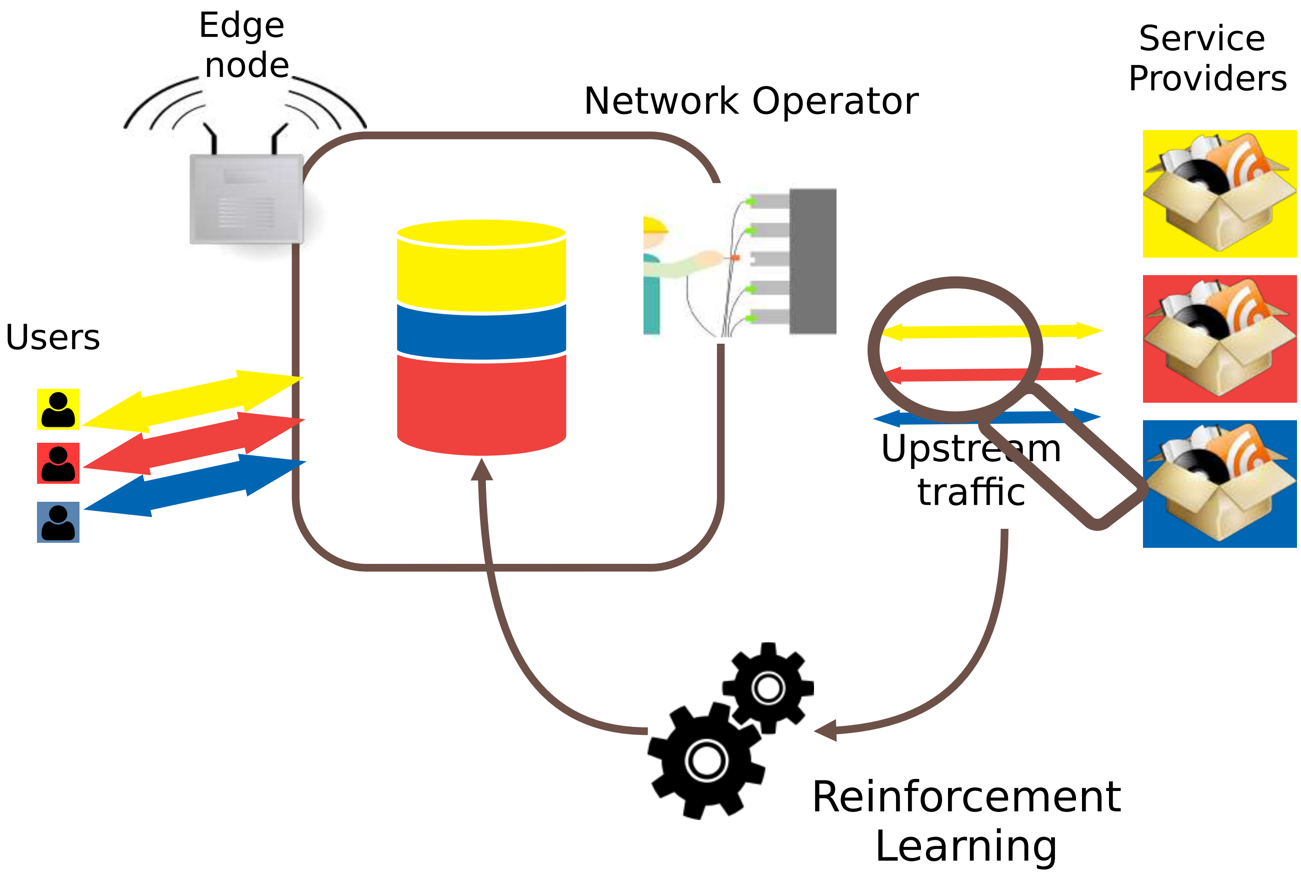
Resource allocation at the Edge



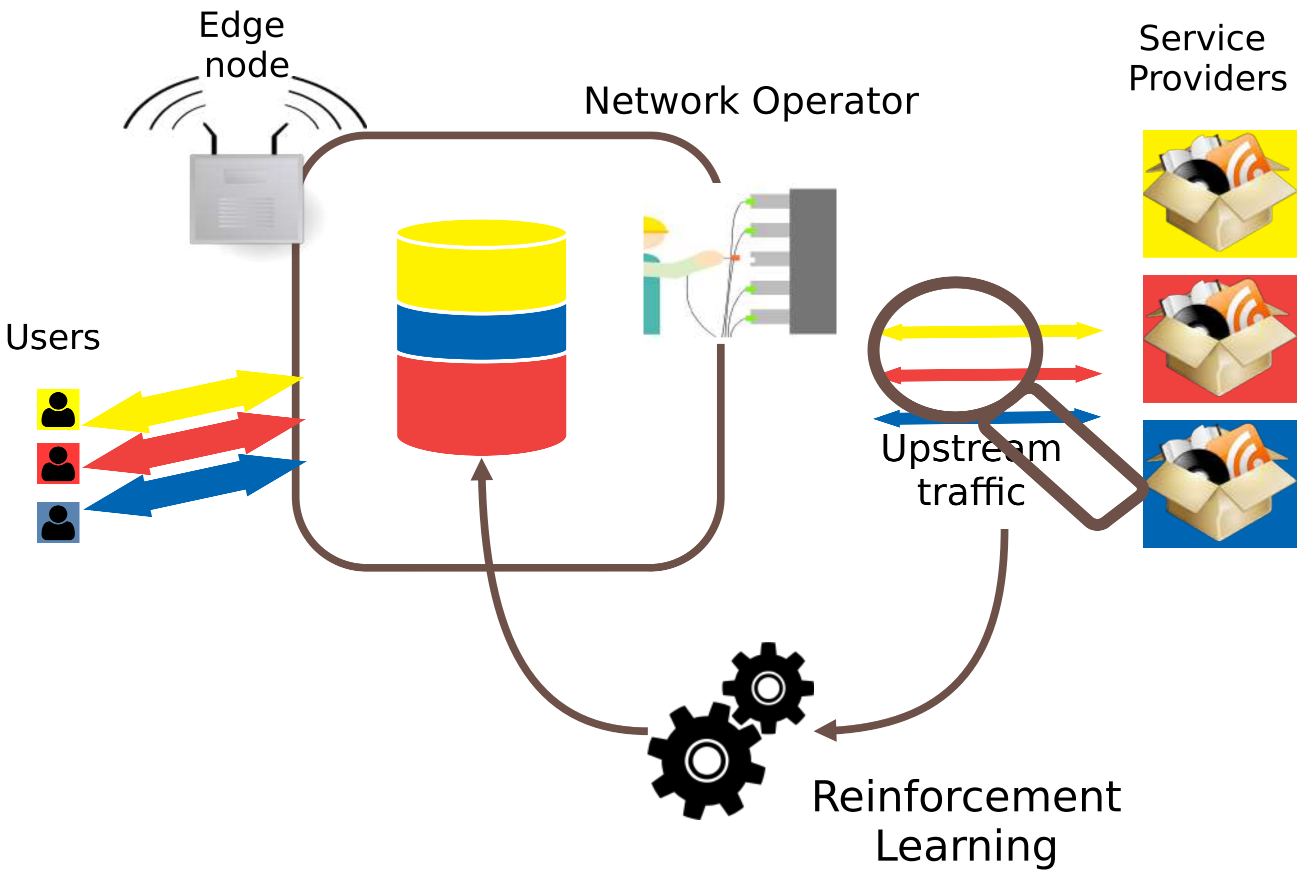
Resource allocation at the Edge



Resource allocation at the Edge



Resource allocation at the Edge



Reviewer's refrain:
"Why don't you apply deep reinforcement learning (DRL)?"

- **Online learning**

  → No long training is possible

  → Experiences come from the running system

  → **Sample efficiency** (only

Resolution: Model-Based Q-Learning

Instantaneous cost:

state (allocation) action (update)

$$\boxed{C_{stoc}\left(\boldsymbol{\theta}^{(k)}, \omega\right)} + \boxed{C_{det}\left(\mathbf{a}^{(k)}\right)}$$

To be learned Deterministic

Resolution: Model-Based Q-Learning

Instantaneous cost:

state (allocation)

action (update)

$$\boxed{C_{stoc}\left(\boldsymbol{\theta}^{(k)}, \omega\right)} + \boxed{C_{det}\left(\mathbf{a}^{(k)}\right)}$$

To be learned

Deterministic

At each timeslot:

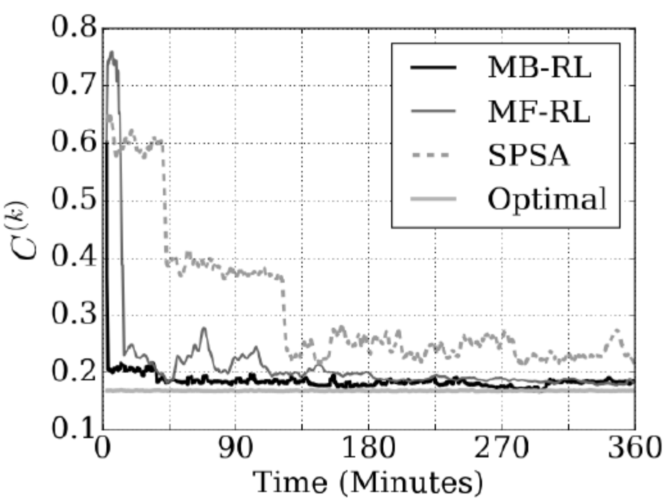
- Train a supervised model $\hat{C}_{\text{stoc}}(\cdot)$ on previous observations
- Apply Q-learning to model $\hat{C}_{stoc}(\cdot)+C_{det}$
- "Simulate" many transitions (200/sec)
  - Without perturbing the real system
  - We can extrapolate $\hat{C}_{\text{stoc}}(\boldsymbol{\theta})$ also at states $\boldsymbol{\theta}$ never visited before

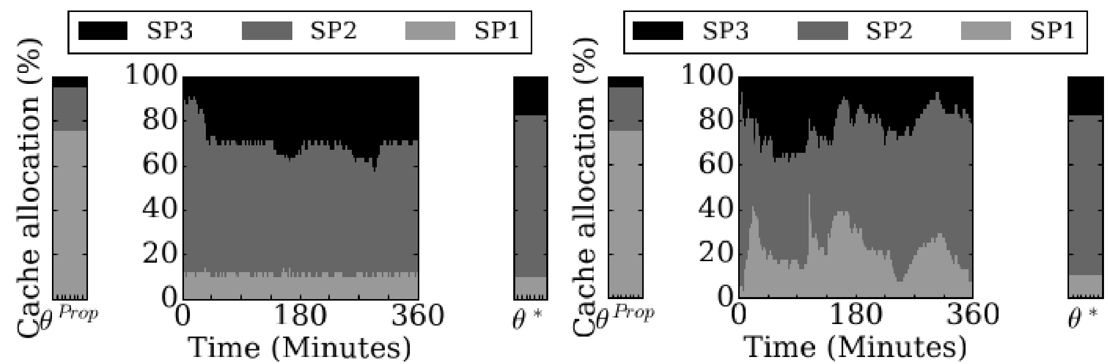
A "good" Q-table is learned very fast

Results

Theorem V-B.1. *If the discount factor γ is sufficiently close to 1 and if the supervised model is an unbiased estimator of the nominal cost*

$$\lim_{k\to\infty} \boldsymbol{\theta}^{(k)} = \hat{\boldsymbol{\theta}}^{*} \textit{ with probability } 1.$$





(a) MB-RL (b) MF-RL

Fig. 8: Evolution of the allocation over time

Conclusion

- AI is effective in managing systems with stochastic and unknown behavior
  - Good practical performance
  - Analytical guarantees
- **TODO**: Test these solutions beyond simulated environments
- Concern
  - “You should compare with deep learning!”
    - Hard to reproduce
    - Deep Learning can become a barrier to nice ideas

Toward **Inference Delivery Networks**:
Distributing Machine Learning with Optimality Guarantees

IEEE Transactions on Networking 2023

Tareq Salem, Gabriele Castellano, Giovanni Neglia, Fabio Pianese, Andrea Araldo

Decision making under uncertainty

- Uncertainty environment
  - User requests,
  - Cost resulting from certain decisions

Decision making under uncertainty

- Uncertainty environment
  - User requests,
  - Cost resulting from certain decisions

- How to make decisions?
  - If we know probability distributions
    - Stochastic / Robust optimization

Decision making under uncertainty

- Uncertainty environment
  - User requests,
  - Cost resulting from certain decisions

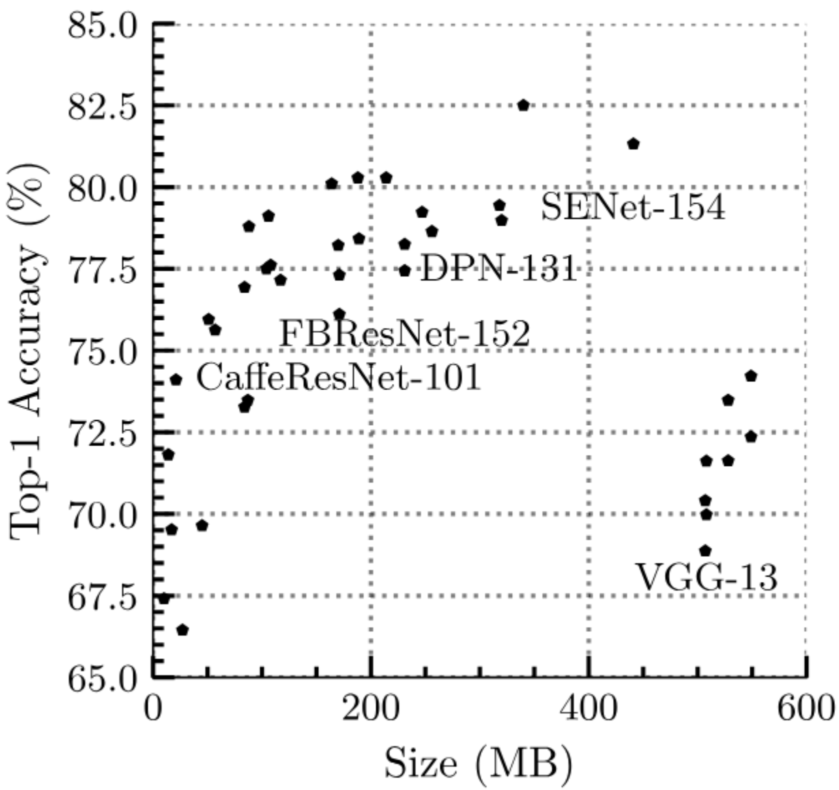
- How to make decisions?
  - If we know probability distributions
    - Stochastic / Robust optimization
  - Otherwise
    - Data-driven approaches, e.g. AI

Definition of online learning

Personal definition inspired in particular from
N. Cesa Bianchi et al.,
Online Learning with Switching Cost and other Adaptive Adversaries
Neurips 2013

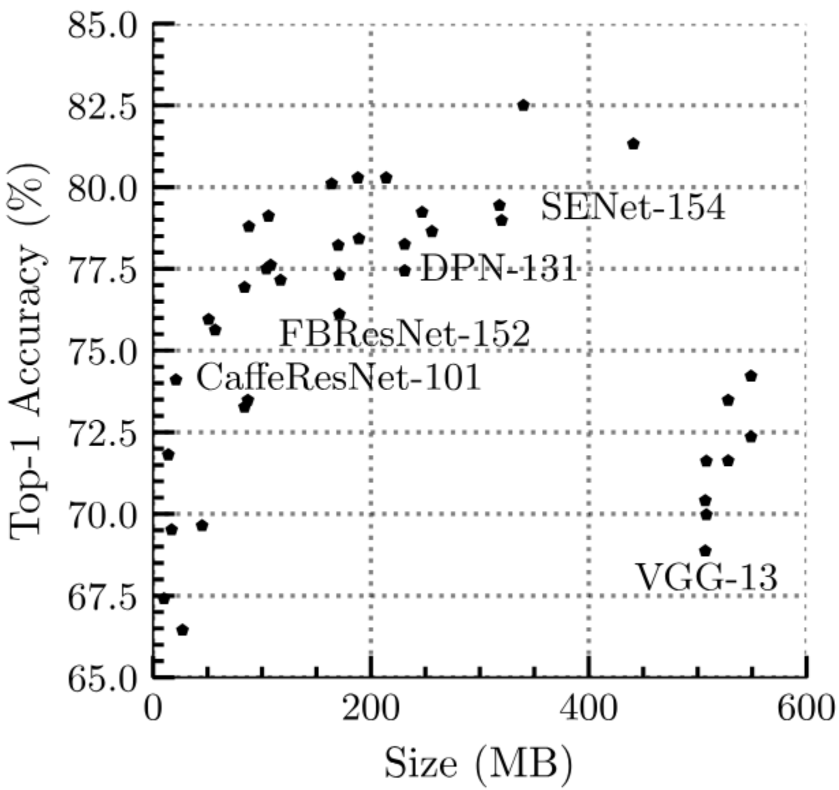
- Sequential decision making setting
- At each round:
  - The environment associates costs to each action
  - The learner selects an action
  - The learner observes the cost of such an action
- The learner aims to minimize the cost
  - The learner implements a strategy
  - Usually, the environment associates the cost so as to minimize the cost under the learner's strategy

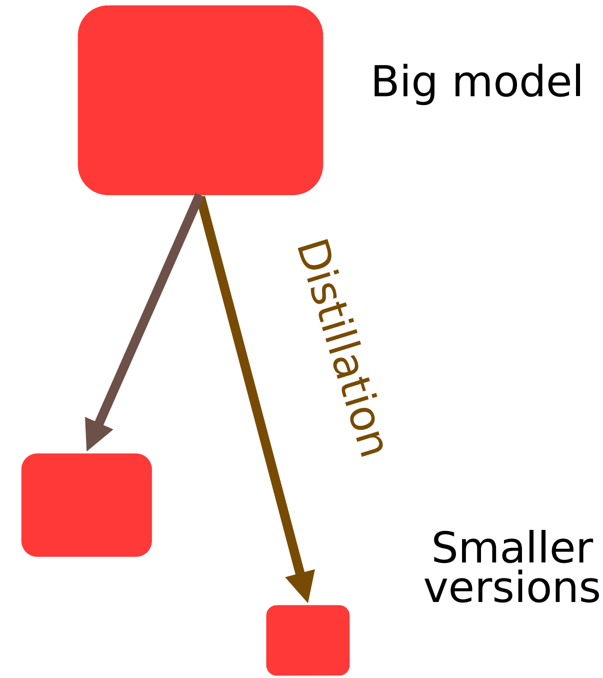
Motivation



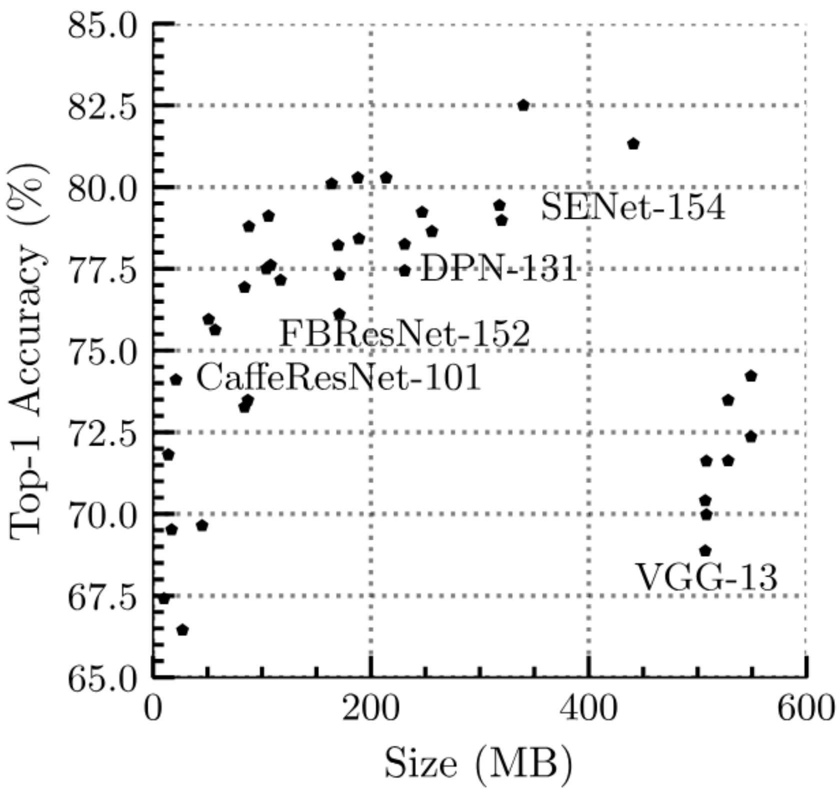
85.0
82.5
80.0
77.5
75.0
72.5
70.0
67.5
65.0
Top-1 Accuracy (%)
0
200
400
600
Size (MB)
SENet-154
DPN-131
FBResNet-152
CaffeResNet-101
VGG-13

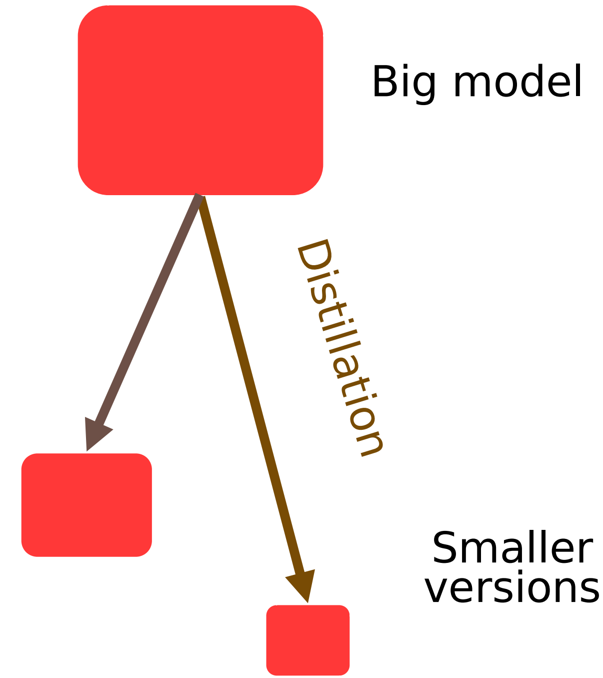
Motivation


85.0
82.5
80.0
77.5
75.0
72.5
70.0
67.5
65.0
Top-1 Accuracy (%)
SENet-154
DPN-131
FBResNet-152
CaffeResNet-101
VGG-13
0
200
400
600
Size (MB)


Big model
Distillation
Smaller versions

Motivation





Research question:

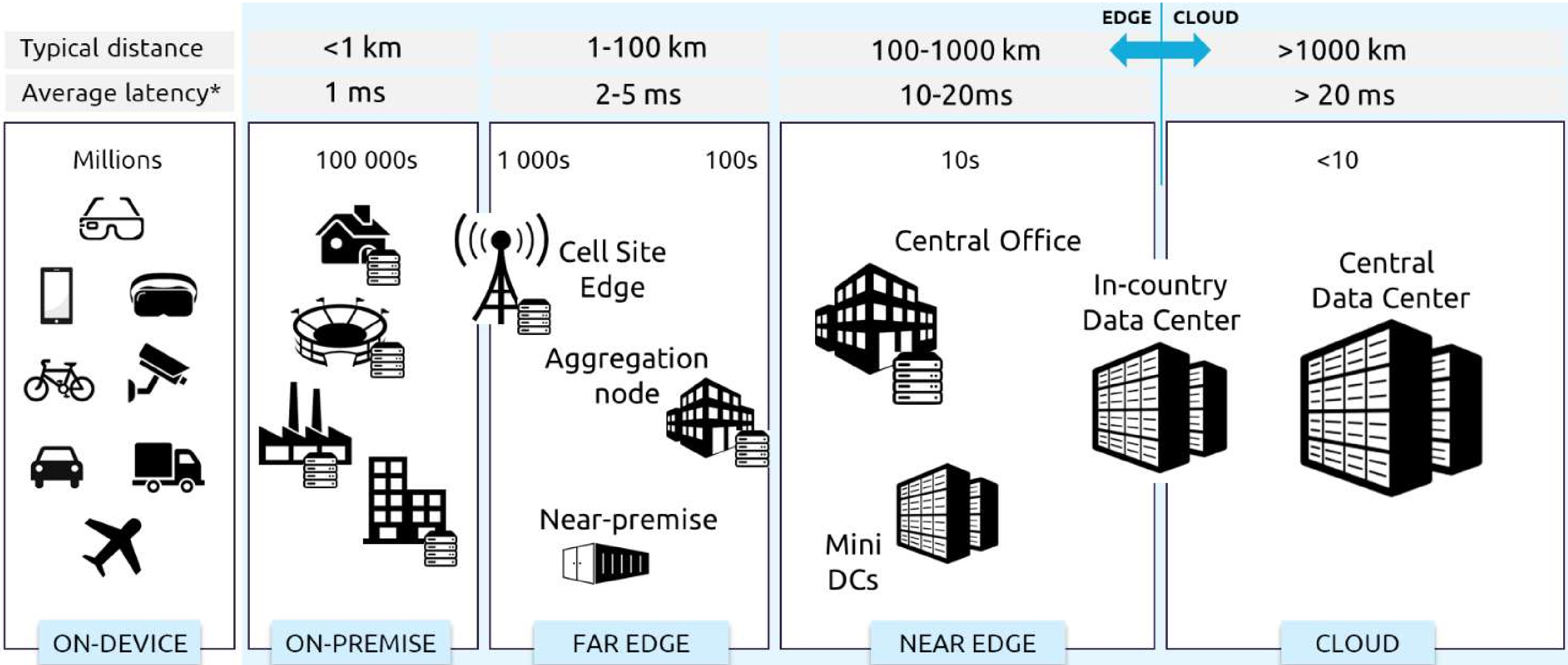
In the cloud-to-edge continuum, **where** to place models and **which version**?

Novel idea:

~~Content~~ Inference Delivery Networks

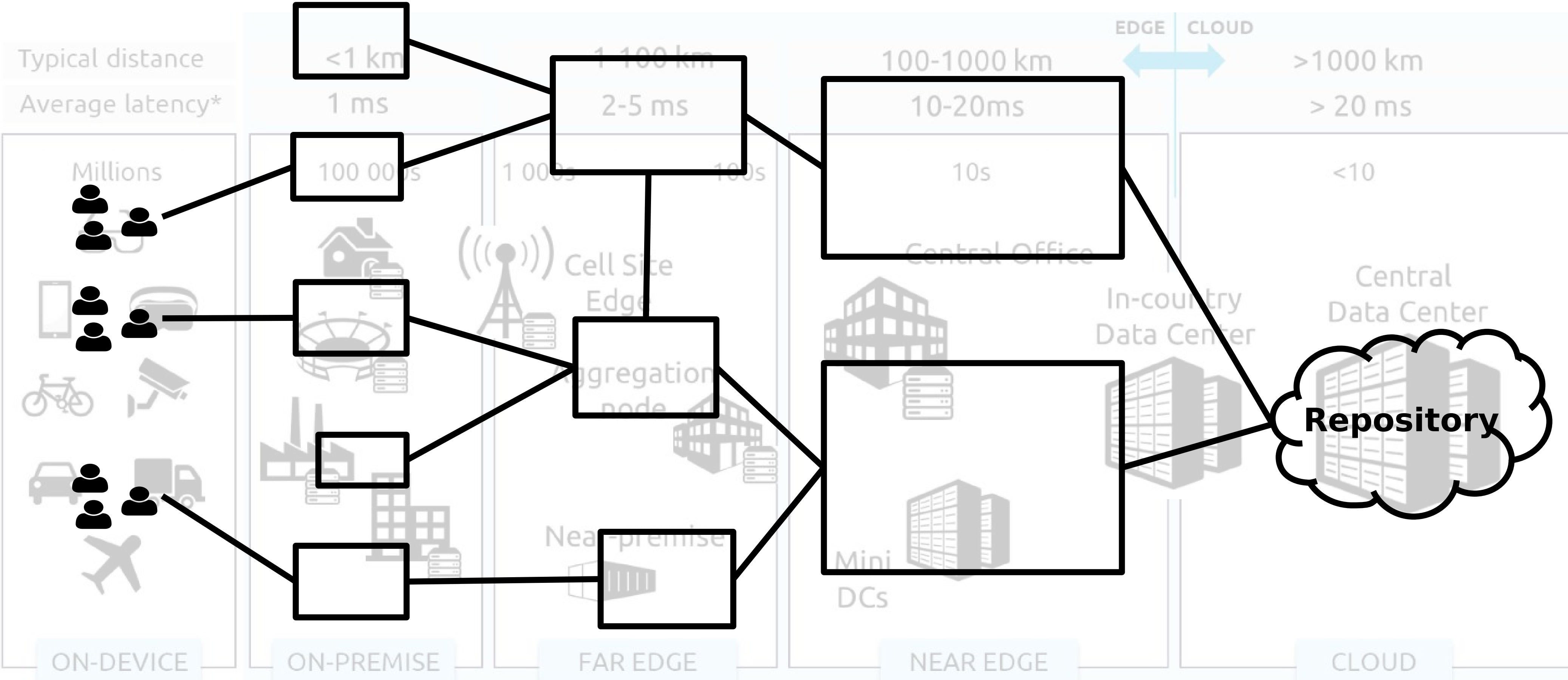
Andrea Araldo

Problem: Inference Delivery Networks



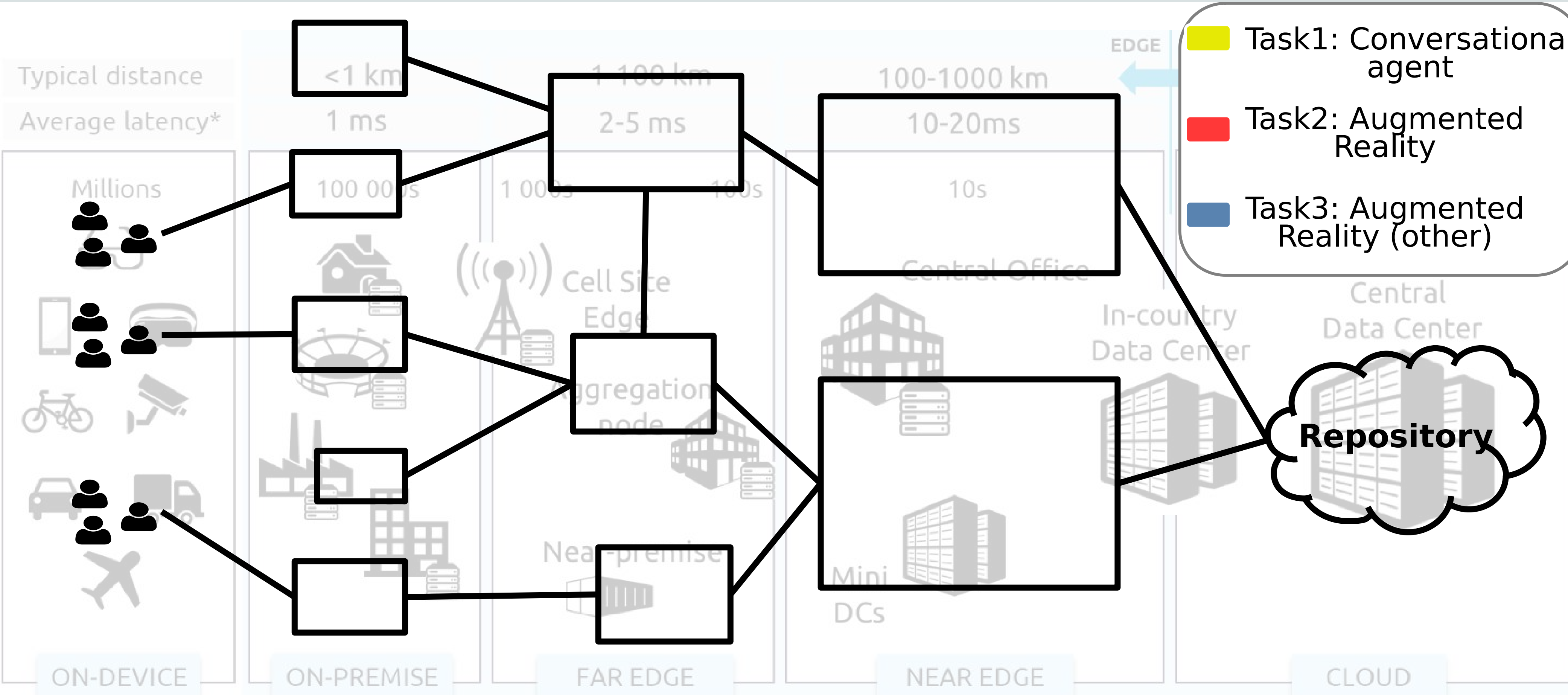
Andrea Araldo

Problem: Inference Delivery Networks



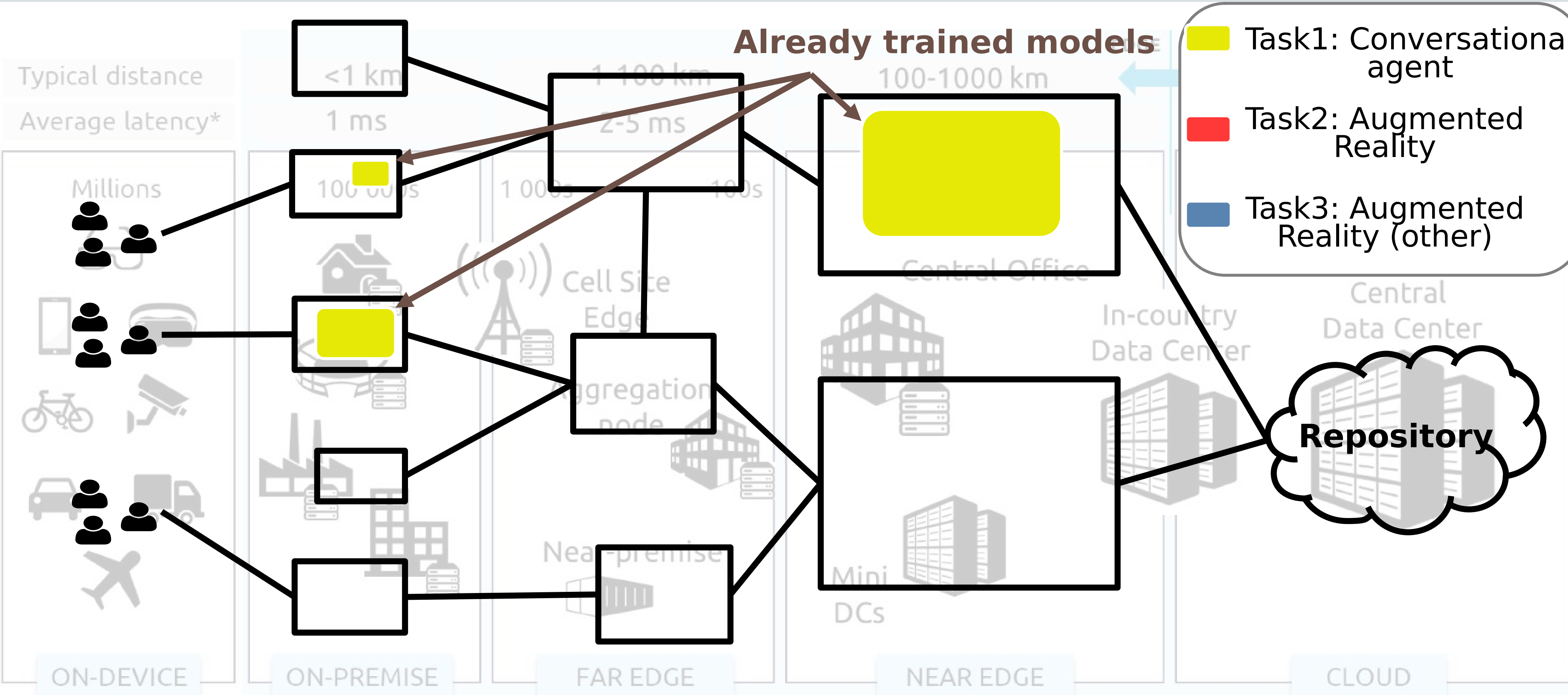
Andrea Araldo

Problem: Inference Delivery Networks

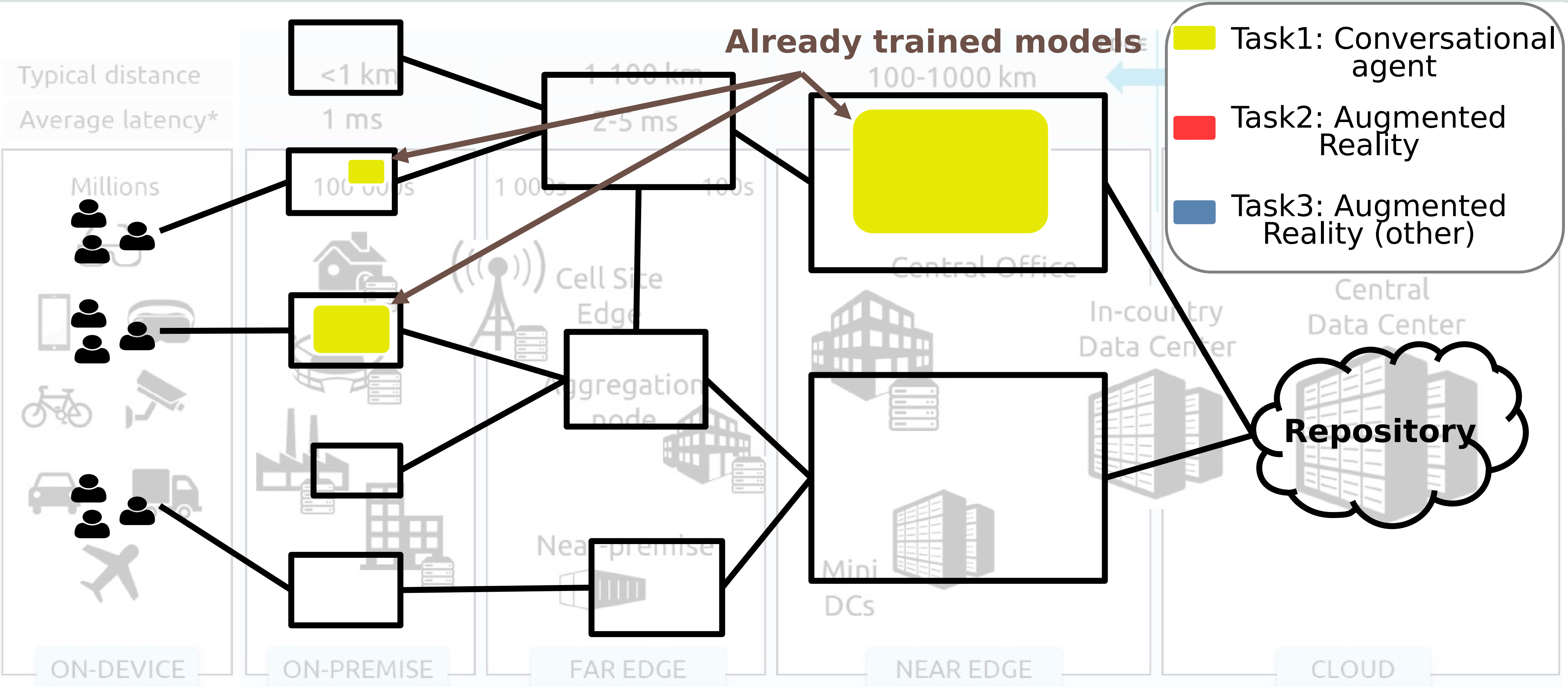


Andrea Araldo

Problem: Inference Delivery Networks



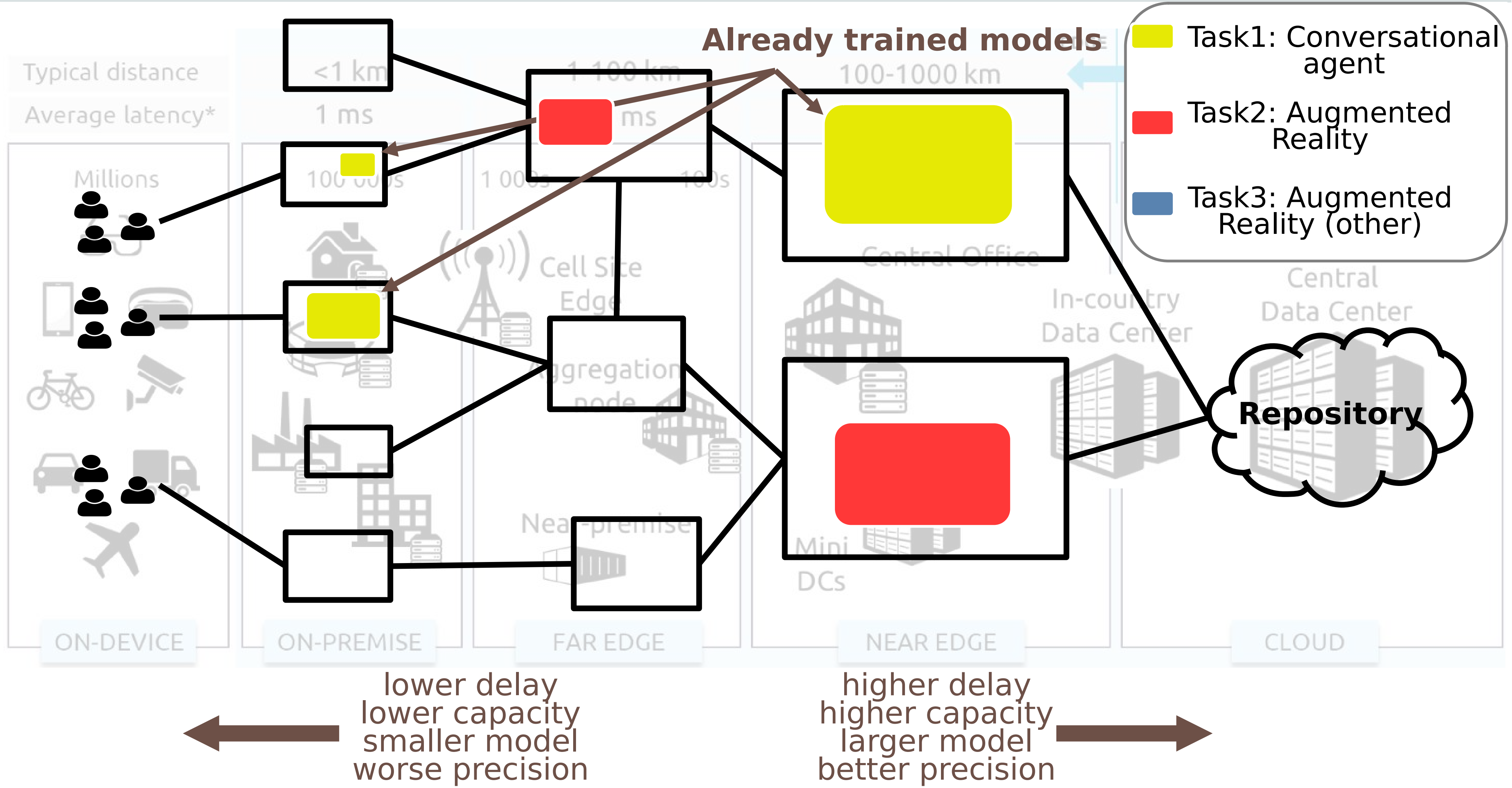
Problem: Inference Delivery Networks



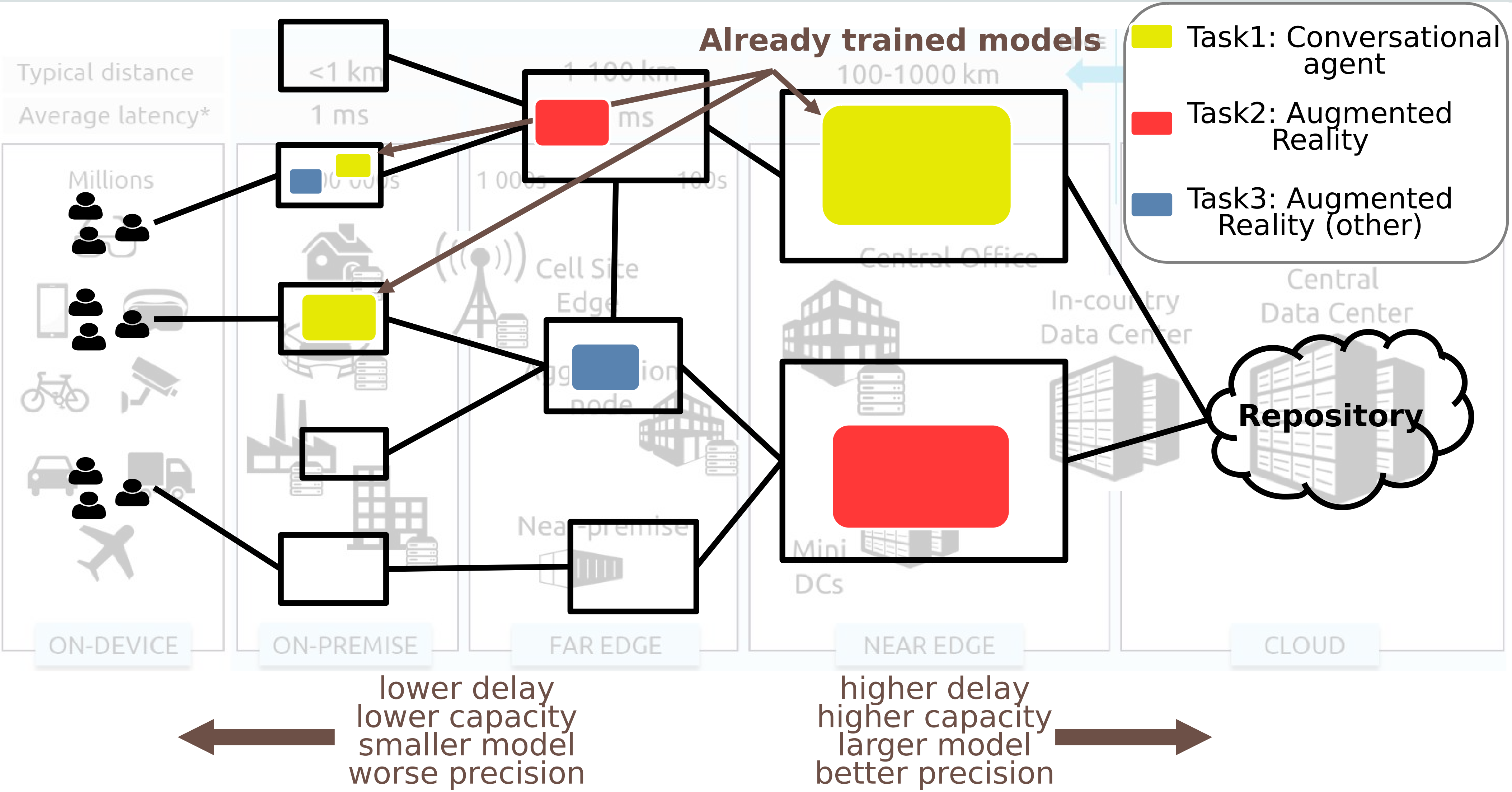
lower delay
lower capacity
smaller model
worse precision

higher delay
higher capacity
larger model
better precision

Problem: Inference Delivery Networks

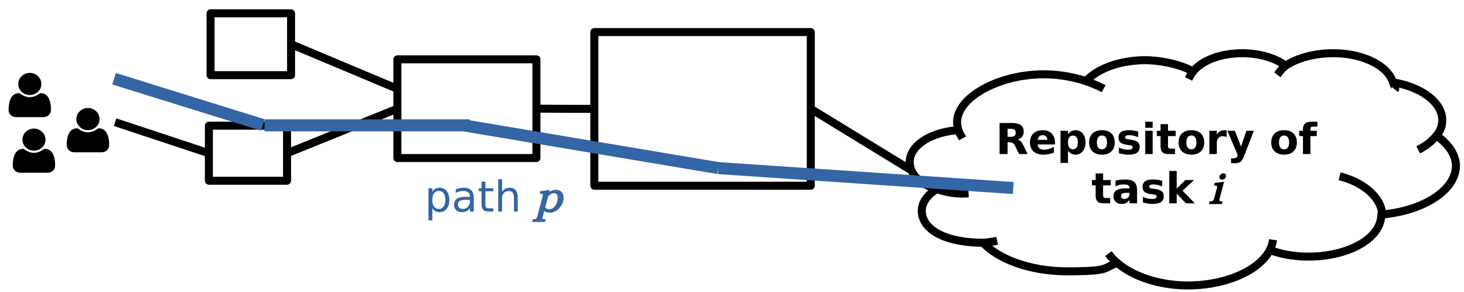


Problem: Inference Delivery Networks



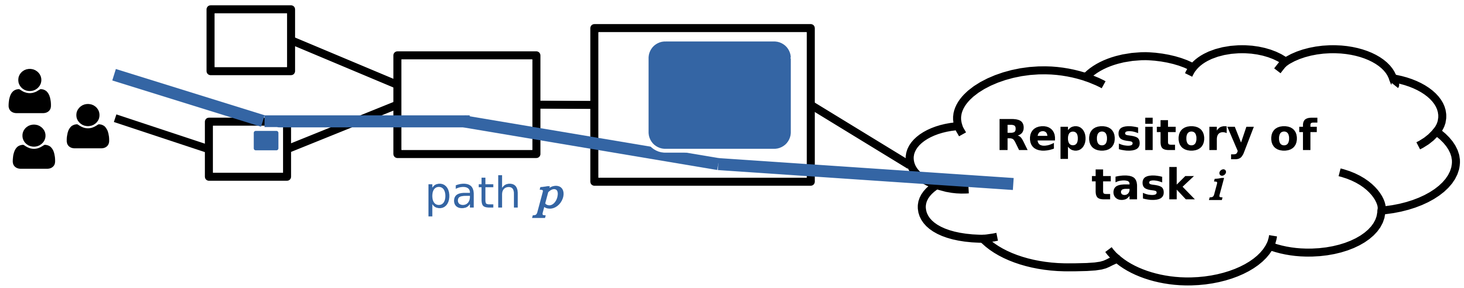
Model

a request for task i going through path $\boldsymbol{p}$, when the request is served via model $m \in \mathcal{M}_i$ on node $p_j \in \boldsymbol{p}$:



Model

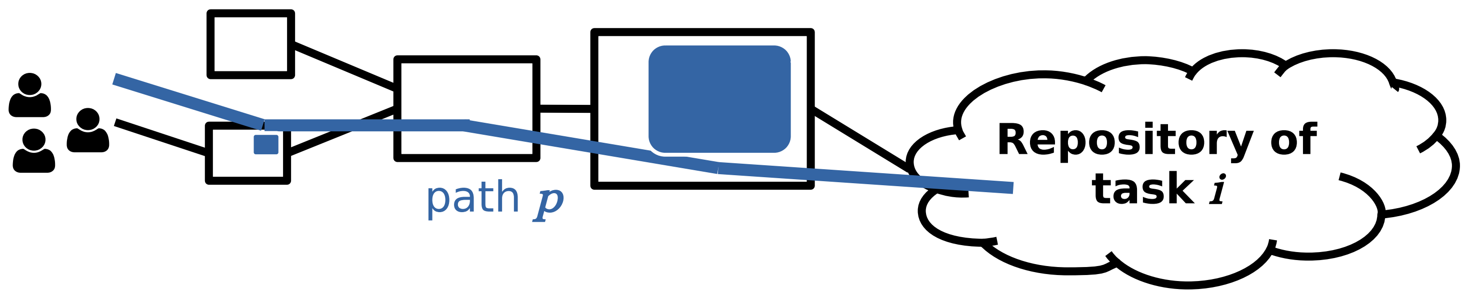
a request for task i going through path $\boldsymbol{p}$, when the request is served via model $m \in \mathcal{M}_i$ on node $p_j \in \boldsymbol{p}$:



Model

- Cost suffered by a request for task i going through path $\boldsymbol{p}$, when the request is served via model $m \in \mathcal{M}_i$ on node $p_j \in \boldsymbol{p}$:

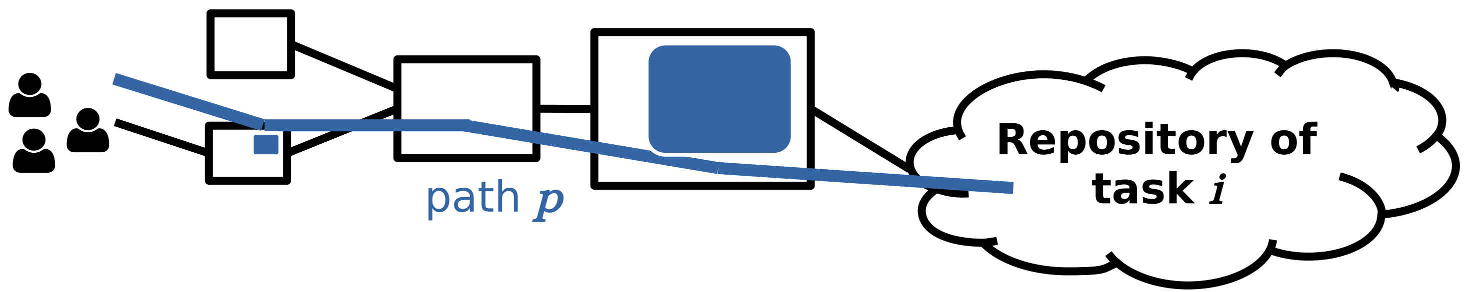
$$C_{\boldsymbol{p},m}^{p_j} = \underbrace{\sum_{j'=1}^{j-1} w_{p_{j'},p_{j'+1}}}_{\text{Round trip time}} + \underbrace{d_m^{p_j}}_{\text{Elaboration time}} + \alpha(1 - \underbrace{a_m}_{\text{Accuracy}})$$



Model

- Cost suffered by a request for task i going through path $\boldsymbol{p}$, when the request is served via model $m \in \mathcal{M}_i$ on node $p_j \in \boldsymbol{p}$:

$$C^{p_j}_{\boldsymbol{p},m} = \sum_{j'=1}^{j-1} \underbrace{w_{p_{j'},p_{j'+1}}}_{\text{Round trip time}} + \underbrace{d^{p_j}_m}_{\text{Elaboration time}} + \alpha(1-\underbrace{a_m}_{\text{Accuracy}})$$



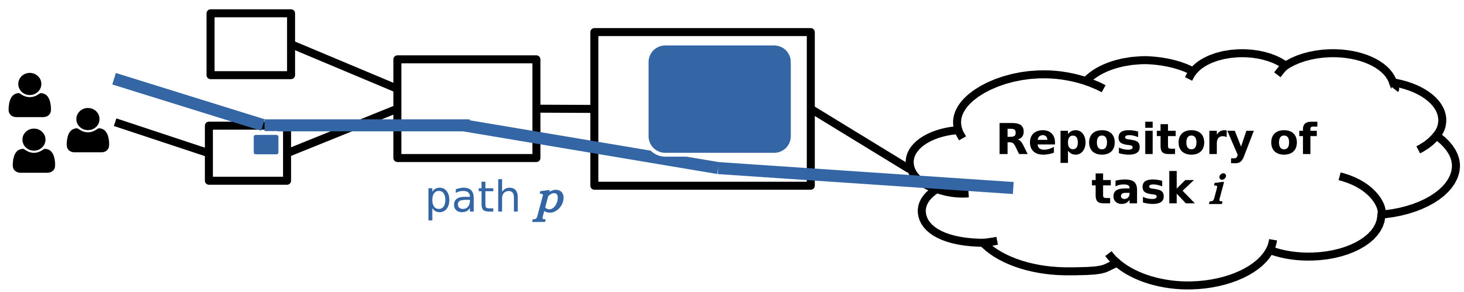
- Input: requests arriving at timeslot t

$$\boldsymbol{r}_t = [r^t_\rho]_{\rho \in \mathcal{R}}$$ $\forall \rho=(i,\boldsymbol{p})$, r_ρ^t is the number of requests for model i following path $\boldsymbol{p}$

Model

- Cost suffered by a request for task i going through path $\boldsymbol{p}$, when the request is served via model $m \in \mathcal{M}_i$ on node $p_j \in \boldsymbol{p}$:

$$C^{p_j}_{\boldsymbol{p},m} = \underbrace{\sum_{j'=1}^{j-1} w_{p_{j'},p_{j'+1}}}_{\text{Round trip time}} + \underbrace{d^{p_j}_m}_{\text{Elaboration time}} + \alpha(1-\underbrace{a_m}_{\text{Accuracy}})$$



- Input: requests arriving at timeslot t

$$\boldsymbol{r}_t = [r^t_\rho]_{\rho\in\mathcal{R}}$$ $\forall\rho=(i,\boldsymbol{p})$, r_ρ^t is the number of requests for model i following path $\boldsymbol{p}$

- Decision at timeslot t

$$\mathbf{x}_t = [X^v{}_{t,m}]_{t,v,m}$$ $X^v{}_{t,m;}=1$ iff model $m \in \mathcal{M}_i$ is placed on node v.

Online learning formulation

Exogenous
(environment)

Placement
decision

$$C(\boldsymbol{r}_t, \quad \boldsymbol{x}_t)$$

Requests arrived

Model placement

Online learning formulation

Exogenous (environment)

Placement decision

$$\underbrace{C(\boldsymbol{r}_t, \boldsymbol{\omega})}_{\text{Cost with the default placement (only in the repository)}} - C(\boldsymbol{r}_t, \boldsymbol{x}_t)$$

Requests arrived

Model placement

Online learning formulation

Exogenous (environment)

Placement decision

$$\underbrace{C(\boldsymbol{r}_t, \quad \boldsymbol{\omega})}_{\text{Cost with the default placement (only in the repository)}} - C(\boldsymbol{r}_t, \quad \boldsymbol{x}_t) = G(\boldsymbol{r}_t, \quad \boldsymbol{x}_t)$$: Instantaneous gain

Requests arrived

Model placement

Online learning formulation

Exogenous (environment)

Placement decision

$$\underbrace{C(\boldsymbol{r}_t,\ \boldsymbol{\omega})}_{\text{Cost with the default placement (only in the repository)}} - C(\boldsymbol{r}_t,\ \boldsymbol{x}_t) = G(\boldsymbol{r}_t,\ \boldsymbol{x}_t)$$: Instantaneous gain

Requests arrived

Model placement

$\psi \in (0, 1]$

What we lose compared to the best placement in hindsight

$$\psi\text{-Regret}_{T,\mathcal{X}} \triangleq \sup_{\{\boldsymbol{r}_t\}_{t=1}^T \in \mathcal{A}^T} \left\{ \psi \sum_{t=1}^{T} G(\boldsymbol{r}_t,\ \boldsymbol{x}_*) - \mathbb{E}\left[\sum_{t=1}^{T} G(\boldsymbol{r}_t,\ \boldsymbol{x}_t) \right] \right\}$$

Best placement in hindsight

$$\boldsymbol{x}_* \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} G(\boldsymbol{r}_t,\ \boldsymbol{x})$$

Andrea Araldo

Online learning formulation

Exogenous (environment)

Placement decision

$$\underbrace{C(\boldsymbol{r}_t, \boldsymbol{\omega})}_{\text{Cost with the default placement (only in the repository)}} - C(\boldsymbol{r}_t, \boldsymbol{x}_t) = G(\boldsymbol{r}_t, \boldsymbol{x}_t)$$: Instantaneous gain

Requests arrived

Model placement

$\psi \in (0,1]$

What we lose compared to the best placement in hindsight

$$\psi\text{-Regret}_{T,\mathcal{X}} \triangleq \sup_{\{\boldsymbol{r}_t\}_{t=1}^{T} \in \mathcal{A}^T} \left\{ \psi \sum_{t=1}^{T} G(\boldsymbol{r}_t, \boldsymbol{x}_*) - \mathbb{E}\left[\sum_{t=1}^{T} G(\boldsymbol{r}_t, \boldsymbol{x}_t) \right] \right\}$$

Worst case analysis:
The environment chooses inputs to maximize our loss

Best placement in hindsight

$$\boldsymbol{x}_* \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=1}^{T} G(\boldsymbol{r}_t, \boldsymbol{x})$$

Andrea Araldo

Algorithm and theoretical guarantee

Algorithm 1 INFIDA Distributed Allocation on Node v

1: **procedure** INFIDA($\boldsymbol{y}_1^v = \underset{\boldsymbol{y}^v \in \mathcal{Y}^v \cap \mathcal{D}^v}{\arg\min} \Phi^v(\boldsymbol{y}^v)$, $\boldsymbol{x}_1^v = \text{DEPROUND}(\boldsymbol{y}_1^v)$, $\eta \in \mathbb{R}_+$)
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Compute $\boldsymbol{g}_t^v \in \partial_{\boldsymbol{y}^v} G(\boldsymbol{r}_t, \boldsymbol{l}_t, \boldsymbol{y}_t)$ through (18).
4:
5: $\hat{\boldsymbol{h}}_{t+1}^v \leftarrow \hat{\boldsymbol{y}}_t^v + \eta \boldsymbol{g}_t^v$ ▷ Take gradient step
6:
7: $\boldsymbol{y}_{t+1}^v \leftarrow \mathcal{P}_{\mathcal{Y}^v \cap \mathcal{D}^v}^{\Phi^v}(\boldsymbol{h}_{t+1}^v)$ ▷ Project new state onto the feasible region using Algorithm 2
8: $\boldsymbol{x}_{t+1}^v \leftarrow \text{DEPROUND}(\boldsymbol{y}_{t+1}^v)$ ▷ Sample a discrete allocation

Theorem 5.1: INFIDA has a sublinear $(1 - 1/e)$-regret w.r.t. the time horizon T, i.e., there exists a constant A such that:

$$(1 - 1/e)\text{-Regret}_{T,\mathcal{X}} \leq A\sqrt{T}, \tag{21}$$

Inference delivery network - proof

Salem, Castellano, Neglia, Pianese, Araldo, Toward Inference Delivery Networks: Distributing Machine Learning With Optimality Guarantees, **IEEE Trans. Net.** 2023

Sketch of the proof

Proof. To prove the ψ-regret guarantee: *(i)* we first establish an upper bound on the regret of the INFIDA policy over its fractional allocations domain $\mathcal{Y}$ against a fractional optimum, then *(ii)* we use it to derive a corresponding ψ-regret guarantee over the integral allocations domain $\mathcal{X}$.

Fractional domain regret guarantee. To establish the regret guarantee of running Algorithm 1 at the level of each computing node $v \in \mathcal{V}$, we showed that the following properties hold:

1) The function G is concave over its domain $\mathcal{Y}$ (Lemma F.1).
2) The mirror map $\Phi : \mathcal{D} \to \mathbb{R}$ is θ-strongly convex w.r.t. the norm $\|\cdot\|_{l_1(\boldsymbol{s})}$ over $\mathcal{Y} \cap \mathcal{D}$, where θ is equal to Eq. (95) (Lemma F.2).
3) The gain function $G : \mathcal{Y} \to \mathbb{R}$ is σ-Lipchitz w.r.t $\|\cdot\|_{l_1(\boldsymbol{s})}$: the subgradients are bounded under the norm $\|\cdot\|_{l_\infty(\frac{1}{\boldsymbol{s}})}$ by σ, i.e., the subgradient of $G(\boldsymbol{r}_t, \boldsymbol{l}_t, \boldsymbol{y})$ at point $\boldsymbol{y}_t \in \mathcal{Y}$ is upper bounded ($\|\boldsymbol{g}_t\|_{l_\infty(\frac{1}{\boldsymbol{s}})} \leq \sigma$) for any $(\boldsymbol{r}_t, \boldsymbol{l}_t) \in \mathcal{A}$ (Lemma F.3).
4) $\|\cdot\|_{l_\infty(\frac{1}{\boldsymbol{s}})}$ is the dual norm of $\|\cdot\|_{l_1(\boldsymbol{s})}$ (Lemma F.4).
5) The Bregman divergence $D_\Phi(\boldsymbol{y}_*, \boldsymbol{y}_1)$ in Eq. (63) is upper bounded by a constant $D_{\max}$ where $\boldsymbol{y}_* = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \sum_{t=1}^{T} G(\boldsymbol{r}_t, \boldsymbol{l}_t, \boldsymbol{y})$ and $\boldsymbol{y}_1 = \underset{\boldsymbol{y} \in \mathcal{Y} \cap \mathcal{D}}{\arg\min}\, \Phi(\boldsymbol{y})$ is the initial allocation (Lemma F.5).

We then apply results from [1]

[1] S. Bubeck, “Convex Optimization: Algorithms and Complexity,” Foundations and Trends® in Machine Learning, Nov. 2015.

Markov Decision Process

- State at timeslot k

$$\boldsymbol{\theta}^{(k)} = (\theta_1^{(k)},\ldots,\theta_P^{(k)})$$

$\boldsymbol{\theta}^{(k)}$

Markov Decision Process

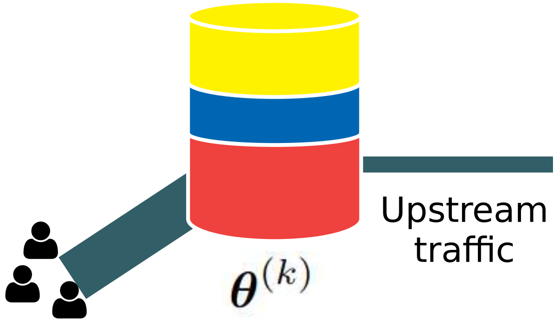
- State at timeslot k

$$\boldsymbol{\theta}^{(k)} = (\theta_1^{(k)},\dots,\theta_P^{(k)})$$

- Nominal cost (upstream traffic)

$$C_{\text{nom}}(\boldsymbol{\theta}^{(k)}, \omega)$$

Random (it depends on users' requests)



Markov Decision Process

- State at timeslot k

$$\boldsymbol{\theta}^{(k)} = (\theta_1^{(k)}, \ldots, \theta_P^{(k)})$$

- Nominal cost (upstream traffic)

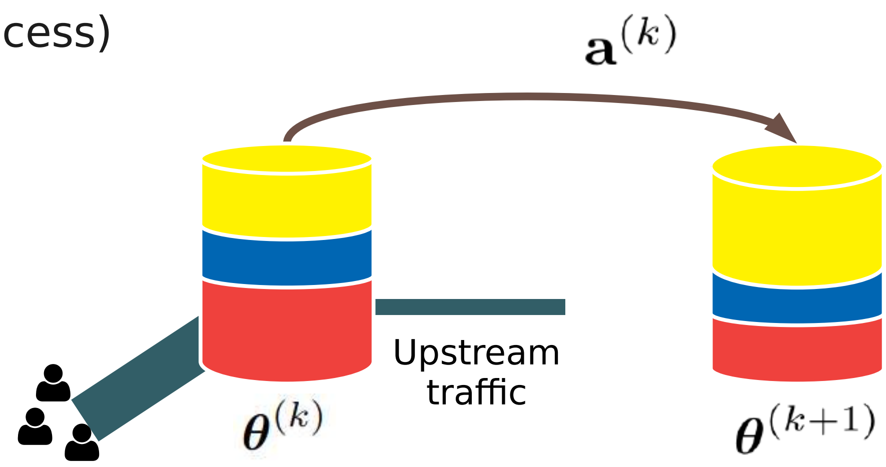
$$C_{\text{nom}}(\boldsymbol{\theta}^{(k)}, \omega)$$

Random (it depends on users' requests)

- Action

$$\mathbf{a}^{(k)} = \boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}$$

(Deterministic Markov Decision Process)



Markov Decision Process

- State at timeslot k

$$\boldsymbol{\theta}^{(k)} = (\theta_1^{(k)},\ldots,\theta_P^{(k)})$$

- Nominal cost (upstream traffic)

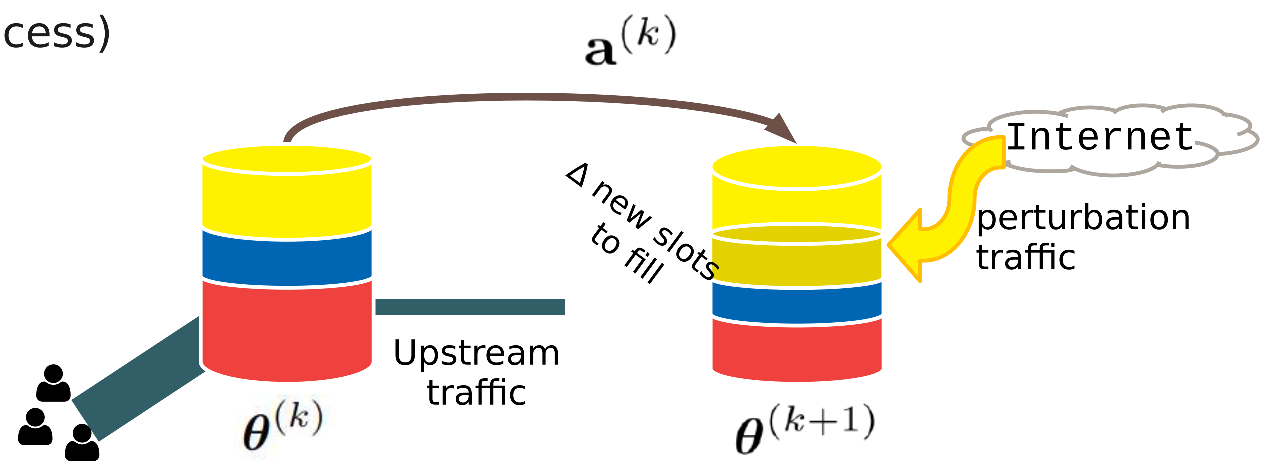
$$C_{\text{nom}}(\boldsymbol{\theta}^{(k)}, \omega)$$ Random (it depends on users' requests)

- Action

$$\mathbf{a}^{(k)} = \boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(\bar{k})}$$

(Deterministic Markov Decision Process)

- Perturbation Cost



Markov Decision Process

- State at timeslot k

$$\boldsymbol{\theta}^{(k)} = (\theta_1^{(k)},\dots,\theta_P^{(k)})$$

- Nominal cost (upstream traffic)

$$C_{\text{nom}}(\boldsymbol{\theta}^{(k)}, \omega)$$

Random (it depends on users' requests)

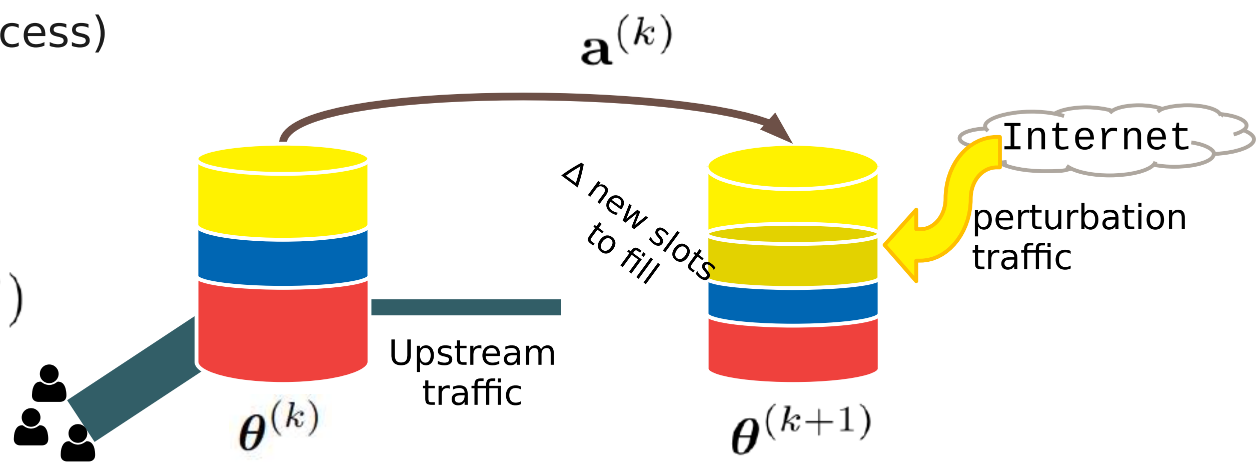
- Action

$$\mathbf{a}^{(k)} = \boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}$$

(Deterministic Markov Decision Process)

- Perturbation Cost
- Instantaneous cost

$$C^{(k)} \triangleq C_{\text{nom}}(\boldsymbol{\theta}^{(k)}, \omega) + C_{\text{pert}}(\mathbf{a}^{(k)})$$



Andrea Araldo

Markov Decision Process

- State at timeslot k

$$\boldsymbol{\theta}^{(k)} = (\theta_1^{(k)},...,\theta_P^{(k)})$$

- Nominal cost (upstream traffic)

$$C_{\text{nom}}(\boldsymbol{\theta}^{(k)}, \omega)$$

Random (it depends on users' requests)

- Action

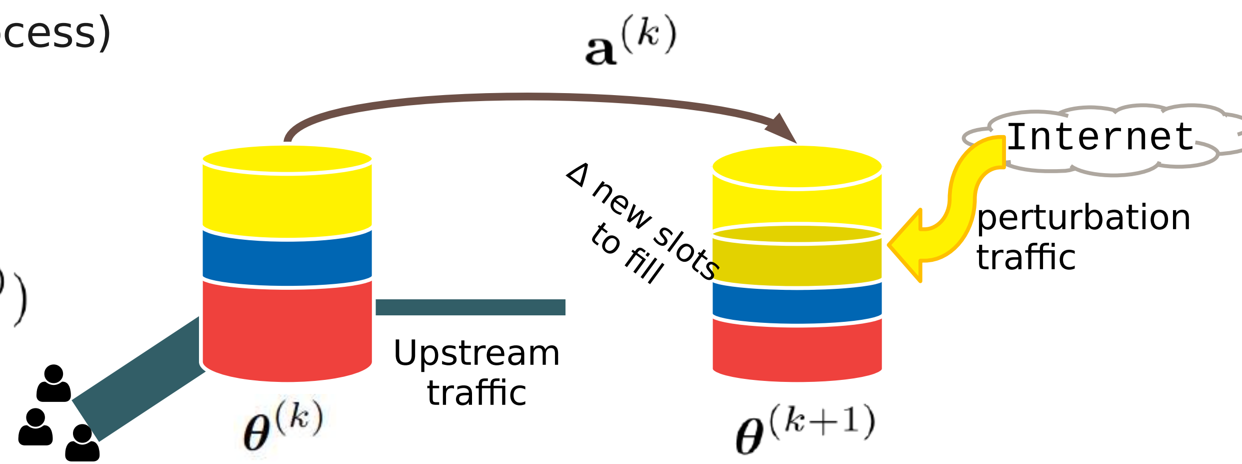
$$\mathbf{a}^{(k)} = \boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(\bar{k})}$$

(Deterministic Markov Decision Process)

- Perturbation Cost
- Instantaneous cost

$$C^{(k)} \triangleq C_{\text{nom}}(\boldsymbol{\theta}^{(k)}, \omega) + C_{\text{pert}}(\mathbf{a}^{(k)})$$

$$C_{\text{cum}}^{\gamma} = \lim_{Z\to\infty} \mathbb{E}\left[\sum_{k=0}^{Z} \gamma^{(k)} \cdot \underbrace{\left(C_{\text{nom}}(\boldsymbol{\theta}^{(k)}, \omega) + C_{\text{pert}}(\mathbf{a}^{(k)})\right)}_{\text{Instantaneous cost } C^{(k)}}\right]$$



Resolution: Model-Based Q-Learning

Algorithm 1: k-th step of Model-based RL

$\alpha^{(k)} \leftarrow$ calculate the value of α ; // formula (17)

$\epsilon^{(k)} \leftarrow$ calculate the value of ϵ ; // formula (18)

with probability $\epsilon^{(k)}$: $\mathbf{a}^{(k)} \leftarrow$ random action ; // ϵ-greedy policy

with probability $1-\epsilon^{(k)}$: $\mathbf{a}^{(k)} \leftarrow$ best action from $Q^{(k)}(\boldsymbol{\theta}, \mathbf{a})$;

$\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k)} + \mathbf{a}^{(k)}$;

$C^{(k)} \leftarrow C_{\text{nom}}(\boldsymbol{\theta}^{(k)}, \omega) + C_{\text{pert}}(\mathbf{a}^{(k)})$;

$Q^{(k)}(\boldsymbol{\theta}^{(k)}, \mathbf{a}^{(k)}) \leftarrow (1-\alpha^{(k)}) \cdot Q^{(k)}(\boldsymbol{\theta}^{(k)}, \mathbf{a}^{(k)}) + \alpha^{(k)} \cdot \left(C^{(k)} + \gamma \min_{\mathbf{a} \in \mathcal{A}_{\boldsymbol{\theta}^{(k+1)}}} Q^{(k)}(\boldsymbol{\theta}^{(k+1)}, \mathbf{a})\right)$; // update $Q^{(k)}$

///////////////

/// Memory replay

$\mathcal{M}^{(k)} \leftarrow \mathcal{M}^{(k-1)} \cup \{(\boldsymbol{\theta}^{(k)}, \mathbf{a}^{(k)}, C_{\text{nom}}(\boldsymbol{\theta}^{(k)}))\}$;

for N_{memory} *times* ; // N_{memory} is the size of the memory mini batch

do

- $(\boldsymbol{\theta}^{\text{rd}}, \mathbf{a}^{\text{rd}}, C^{\text{rd}}_{\text{nom}}) \leftarrow$ random element from $\mathcal{M}^{(k)}$;
- $\boldsymbol{\theta}'^{\text{rd}} \leftarrow \boldsymbol{\theta}^{\text{rd}} + \mathbf{a}^{\text{rd}}$;
- $Q^{(k)}(\boldsymbol{\theta}^{\text{rd}}, \mathbf{a}^{\text{rd}}) \leftarrow (1-\alpha^{(k)}) \cdot Q^{(k)}(\boldsymbol{\theta}^{\text{rd}}, \mathbf{a}^{\text{rd}}) + \alpha^{(k)} \cdot \left(C^{\text{rd}}_{\text{nom}} + C_{\text{pert}}(\mathbf{a}^{\text{rd}}) + \gamma \min_{\mathbf{a} \in \mathcal{A}_{\boldsymbol{\theta}'^{\text{rd}}}} Q^{(k)}(\boldsymbol{\theta}'^{\text{rd}}, \mathbf{a})\right)$; // update $Q^{(k)}$

end

///////////////

/// Model training and inference

$\mathcal{D}_p^{(k)} \leftarrow \mathcal{D}_p^{(k-1)} \cup \{(\theta_p^{(k)}, C_{\text{nom,p}}(\theta_p^{(k)}))\}$; // collect realization of $C_{\text{nom,p}}(\theta_p^{(k)})$ for each SP p

$\hat{C}^{(k)}_{\text{nom},p}(\theta_p) \leftarrow$ estimate model from $\mathcal{D}_p^{(k)}$ **for** N_{model} *times* ; // N_{model} is the size of the model mini batch

do

- $\boldsymbol{\theta}^{\text{rd}} \leftarrow$ random state from $\mathcal{S}$;
- $\mathbf{a}^{\text{rd}} \leftarrow$ random action from $\mathcal{A}_{\boldsymbol{\theta}^{\text{rd}}}$;
- $\boldsymbol{\theta}'^{\text{rd}} \leftarrow \boldsymbol{\theta}^{\text{rd}} + \mathbf{a}^{\text{rd}}$;
- Compute $\hat{C}^{(k)}_{\text{nom,p}}(\theta_p^{\text{rd}})$; // predict the nominal cost using the model
- $\hat{C} \leftarrow \sum_{\text{p}=1}^{P} \hat{C}_{\text{nom,p}}(\theta_p^{\text{rd}})) + C_{\text{pert}}(\mathbf{a}^{\text{rd}})$;
- $Q^{(k)}(\boldsymbol{\theta}^{\text{rd}}, \mathbf{a}^{\text{rd}}) \leftarrow (1-\alpha^{(k)}) \cdot Q^{(k)}(\boldsymbol{\theta}^{\text{rd}}, \mathbf{a}^{\text{rd}}) + \alpha^{(k)} \cdot \left(\hat{C} + \gamma \min_{\mathbf{a} \in \mathcal{A}_{\boldsymbol{\theta}'^{\text{rd}}}} Q^{(k)}(\boldsymbol{\theta}'^{\text{rd}}, \mathbf{a})\right)$; // update $Q^{(k)}$

end

RL for caching - proof

Ben-Ameur, Araldo, Chahed, Cache Allocation in Multi-tenant Edge Computing: An Online Model-based Reinforcement Learning Approach, **IEEE ICC** 2022 & **Major Revision in IEEE Trans. Cloud Comp.**

Theorem V-B.1. *If the discount factor γ is sufficiently close to 1*

$$\lim_{k\to\infty} \boldsymbol{\theta}^{(k)} = \hat{\boldsymbol{\theta}}^{*} \text{ with probability } 1.$$

Sketch of the proof

- We prove that our Q-table $Q^{(k)}$ converges to the optimal Q-table Q^* with probability 1.
- We prove that the sequence of actions and states induced by Q^* has an absorbing state that is the discretely optimal state $\hat{\boldsymbol{\theta}}^*$.
- We prove that the sequence of actions and states induced by our Q-table $Q^{(k)}$ also follows Q^*.
- We prove that the sequence of actions and states that we take online converges with probability 1 to the sequence induced by our Q-table $Q^{(k)}$ (assuming no more exploration).
- Finally, we show that this sequence converges with probability 1 to a sequence induced by Q^*.