Linha Redes VEICULARES

IC – Instituto de Computação

UNICAMP

Campinas, December 16th, 2025

CMFog-DRL

Integrando Inteligência Artificial para Tomada de Decisão em Ambientes de Fog Computing

Dr. Marcelo Claudio Sousa Araújo
Supervisor: Prof. Dr. Luiz Fernando Bittencourt

IC - Instituto de Computação | UNICAMP

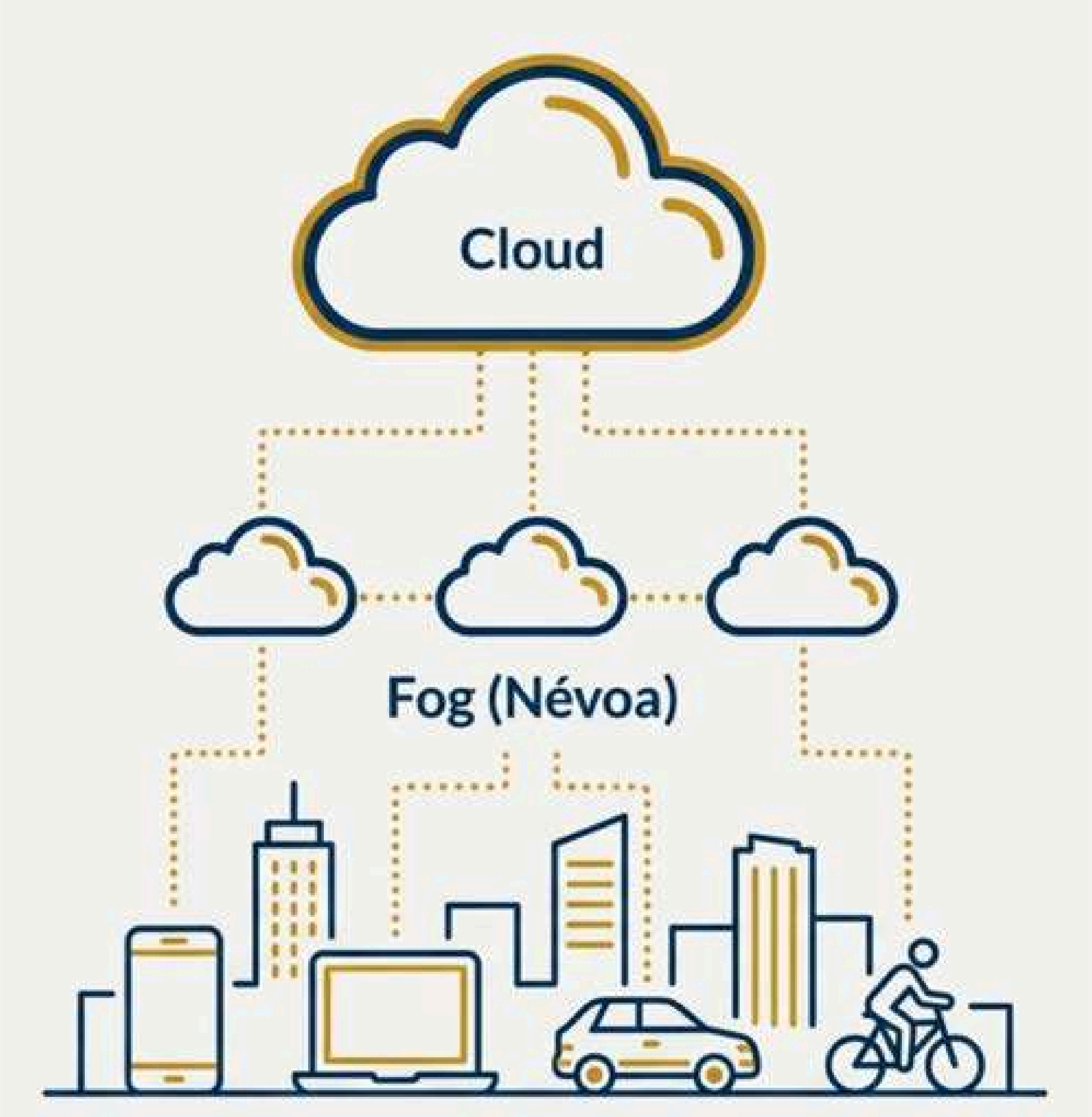
16 de dezembro de 2025

O Desafio da Borda

A explosão de dispositivos IoT cria um ambiente altamente dinâmico.

Processamento na "Névoa" (Fog) é essencial para baixa latência.

- **Alta Mobilidade dos Usuários:** os usuários mudam constantemente seus pontos de acesso à rede.
- **Heterogeneidade de Recursos:** os nós de borda possuem capacidades computacionais e de armazenamento variadas.
- **Cenários Dinâmicos:** as condições da rede e as demandas das aplicações são imprevisíveis.



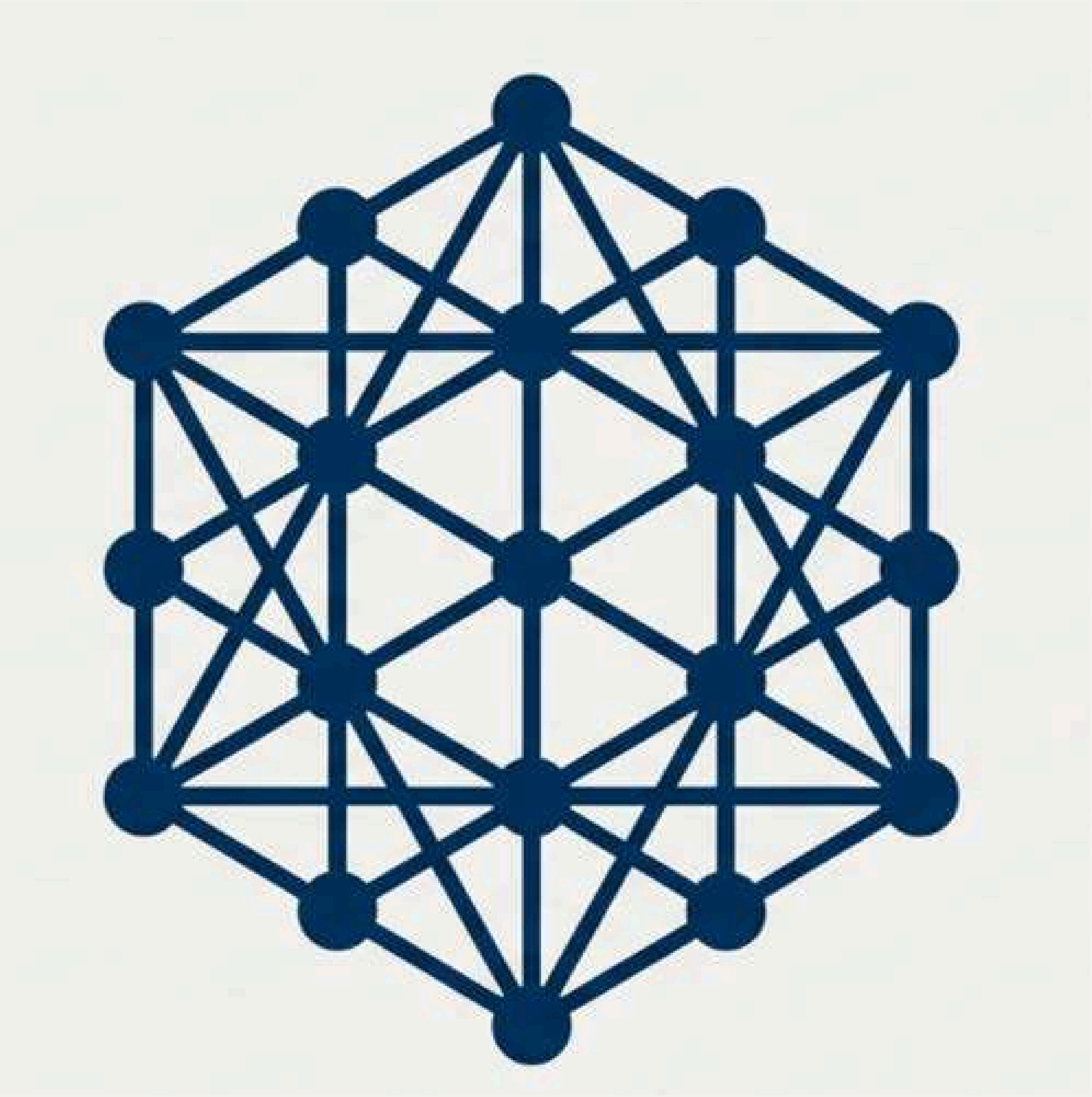
A Abordagem Inicial: CMFog

Estratégia proativa de migração de conteúdo para uma infraestrutura de névoa (fog) multinível.

Seu objetivo é garantir que os dados estejam disponíveis no local e no momento corretos, reduzindo a latência.

Tomada de decisão baseada em:

- **Multiple Attribute Decision Making (MADM):** uso de regras e heurísticas pré-definidas.
- **Predição de Mobilidade com Cadeias de Markov:** modelo para prever o deslocamento dos usuários.



Deep Reinforcement Learning possibilita um sistema adaptativo

O DRL integra Deep Learning e Reinforcement Learning para aprender políticas ótimas por meio da interação contínua com o ambiente, permitindo adaptação dinâmica às mudanças e tomada de decisão com mínima intervenção manual.

1.Adaptacão Contínua

Responde a novos cenários sem intervenção manual.

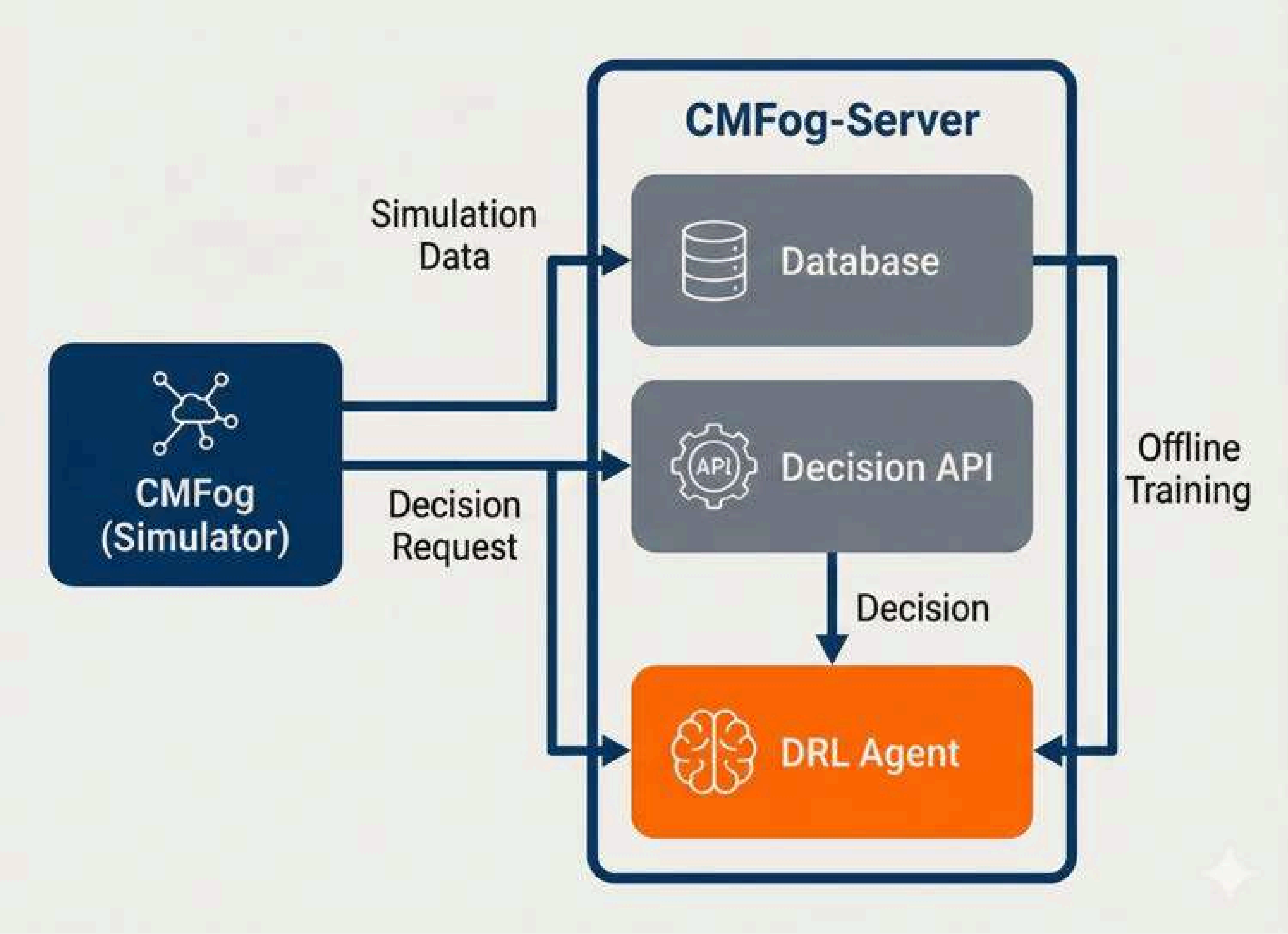
2. Aprendizado Contínuo

Refina decisões com base em novos padrões.

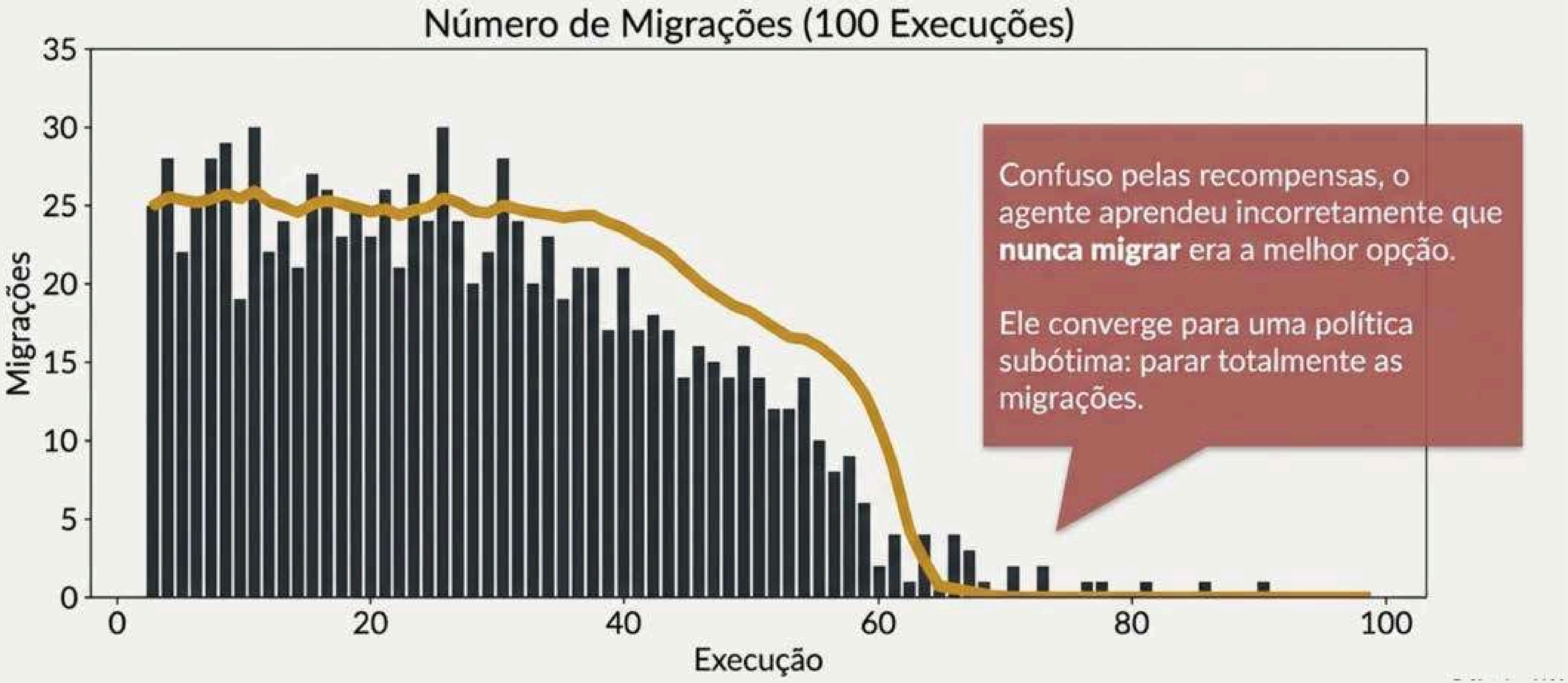
3. Otimização Multiobjetiva

Equilibra múltiplos objetivos (latência, custo) sem regras rígidas.

Uma arquitetura para experimentação reprodutível



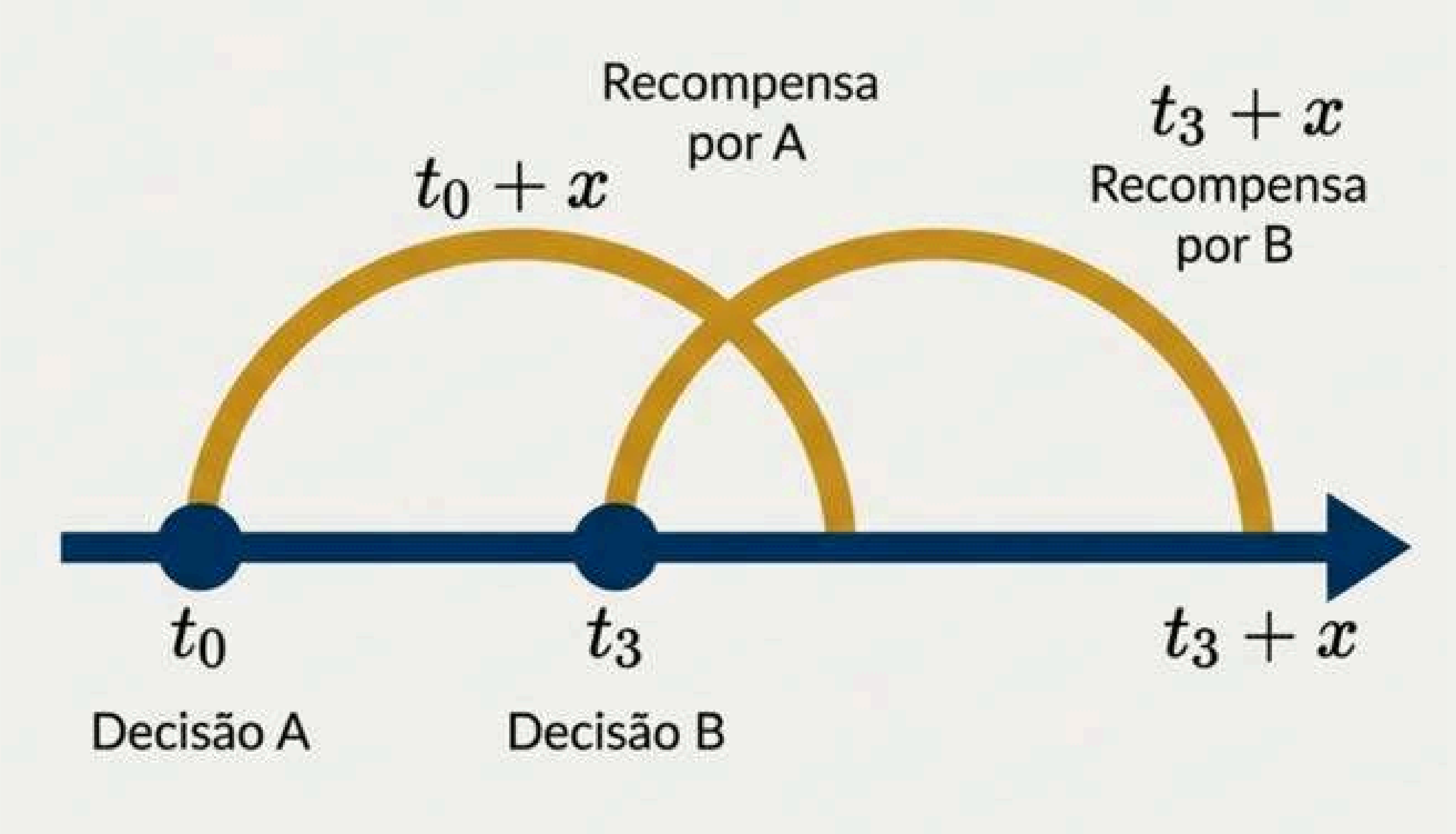
O Obstáculo: Delayed Rewards



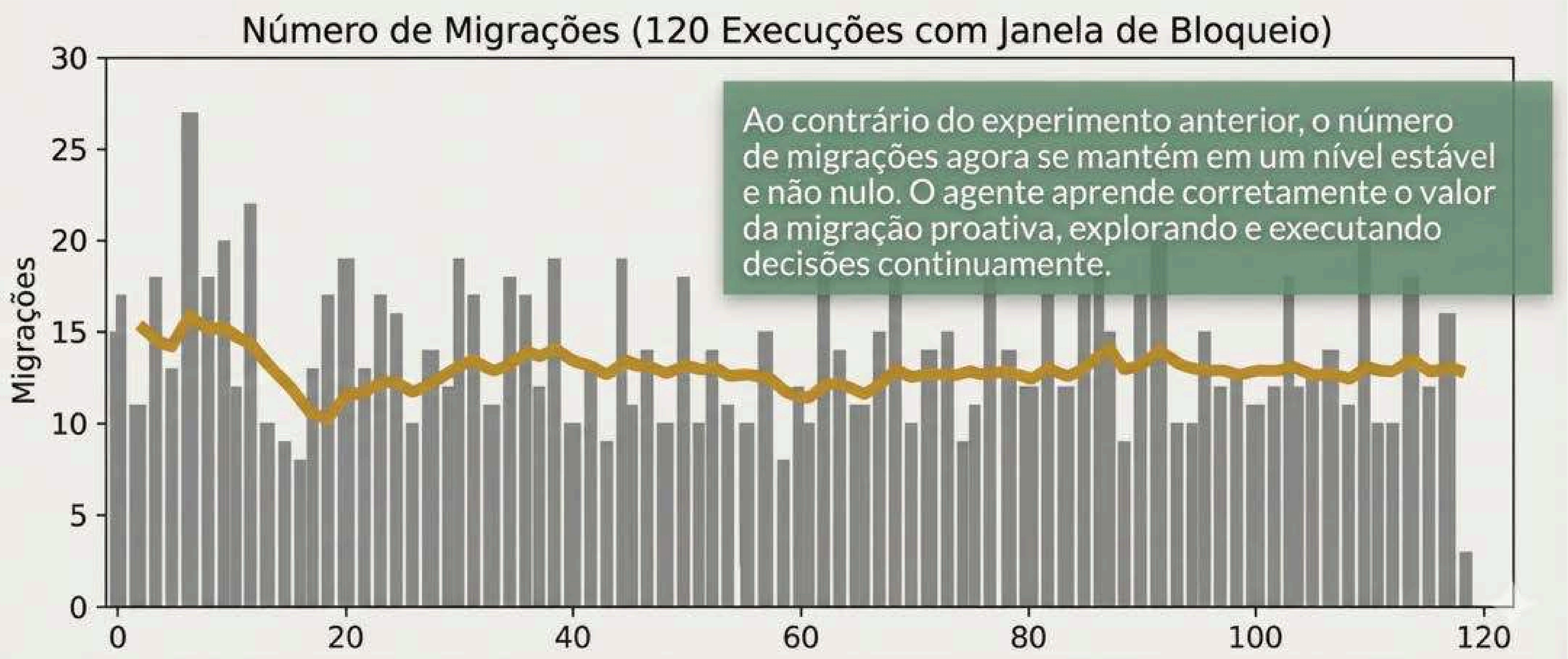
O Obstáculo: Delayed Rewards

O Problema: O impacto de uma decisão (A) só é observado algum tempo depois.

A Complicação: Múltiplas decisões se sobrepõem, tornando ambíguo qual ação merece crédito ou culpa, confundindo o agente.



Solução: Delayed Rewards



Ajustes na Estratégia de Treinamento

Otimização da janela de bloqueio: é necessários fazer fine tuning do tamanho da janela

Ajustes de parâmetros de treinamento: identificar valores para parâmetros que permitam maximizar a qualidade de aprendizado do agente

Função de rewards: ajustar os pesos utilizados na função de recompensa de forma que agente possa aprender o impacto da suas decisões.

Desafios atuais

Datasets de inputs de treinamento: dados gerados artificialmente pode não ter a qualidade necessária para treinamento do agente

Função de recompensa: não existe uma heurística para determinar valores ótimos para os pesos

Obrigado!

Marcelo Claudio Sousa Araújo
marcelo.araujo@ic.unicamp.br

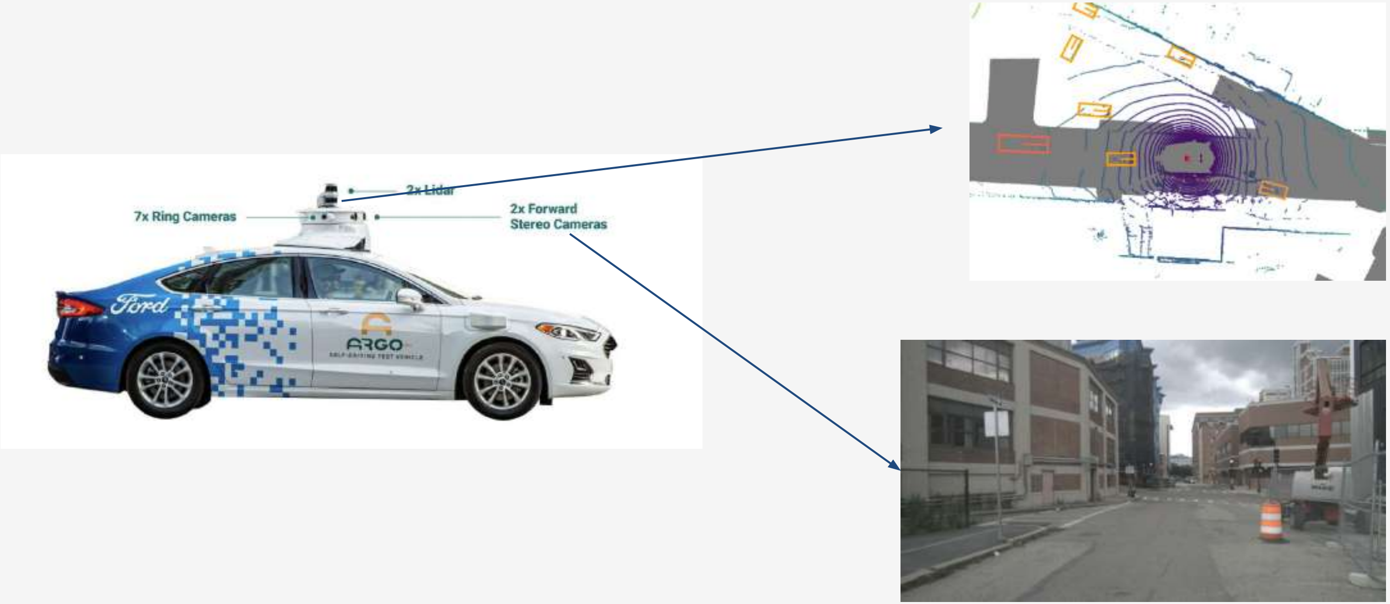
Efficient 2D LiDAR Scene Understanding for Autonomous Driving via Multi-Modal Self-Supervised Learning

Luis F. Solis Navarro[1], Daryel León[2], Assi Chadi[2], Carlos A. Astudillo[1]

- University of Campinas (Brazil)[1]
- Concordia University (Canadá)[2]

Problem Definition

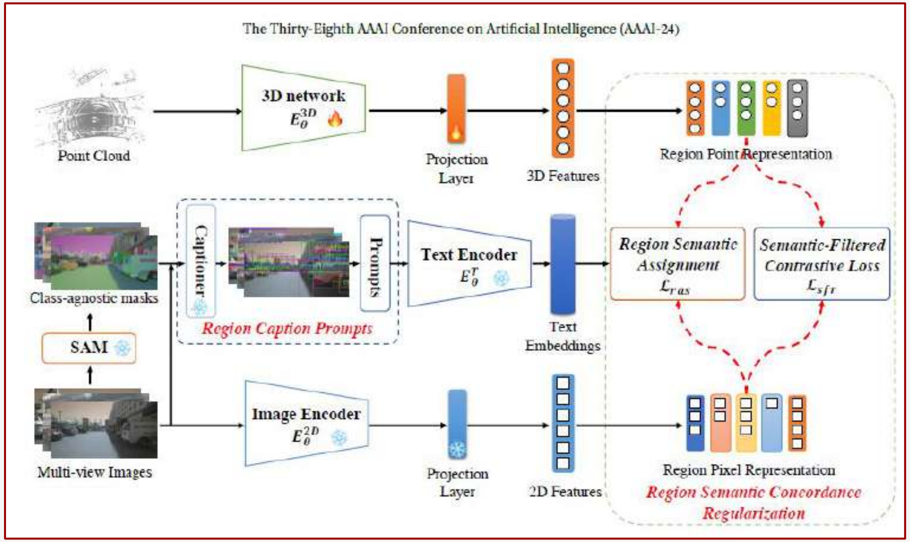
- Synchronizing sensor data types (camera + Lidar) can be a major challenge. Applications: **collision detection, for instance.**
- Lightweight model for running in vehicles.



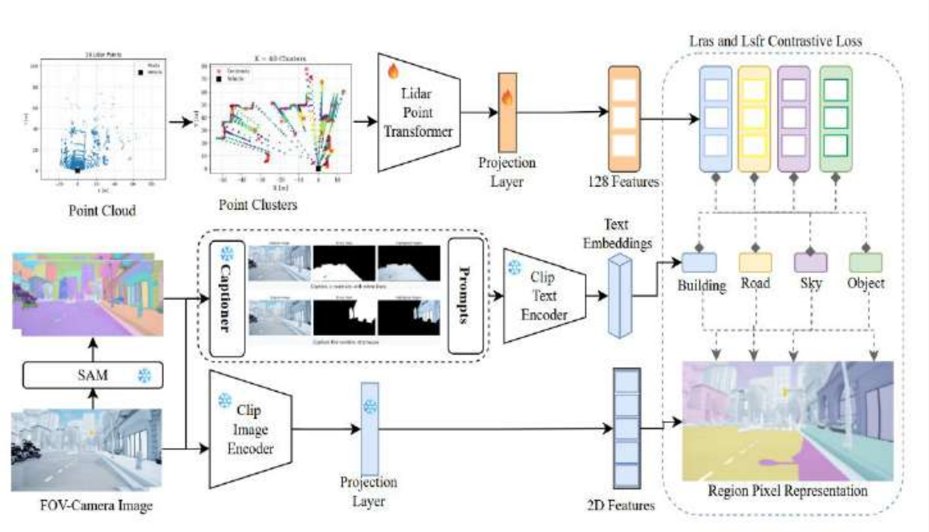
Lidar Point Transformer

VLM2Scene: Self-Supervised Image-Text-LiDAR Learning with Foundation Models for Autonomous Driving Scene Understanding

Learning without Labels: Cross-Modal Pre-Training for LiDAR Transformers via Image-Text Supervision



(Liao et al., 2024)



Ours

Model: Minkowski U-Net
Lidar Point: All points - one by one

Model: Transformer Model
Lidar Point: Clustered lidar point in **sequential batch (10)**

Objectives

INSTITUTO DE COMPUTAÇÃO

Develop a **Transformer architecture** designed to process 2D LiDAR point clouds. The model will convert raw coordinates into **enriched vector representations** that capture geometry and global context.

Implement a self-supervised pre-training framework using images and text to generate pseudo-labels. This allows the model to learn robust semantic features without costly manual annotations.

Validate the pre-trained model by fine-tuning it for the task of **object detection in LiDAR**. Success will be measured by evaluating its accuracy, thereby proving the learned representations' effectiveness.

Pre Training

Loss Function

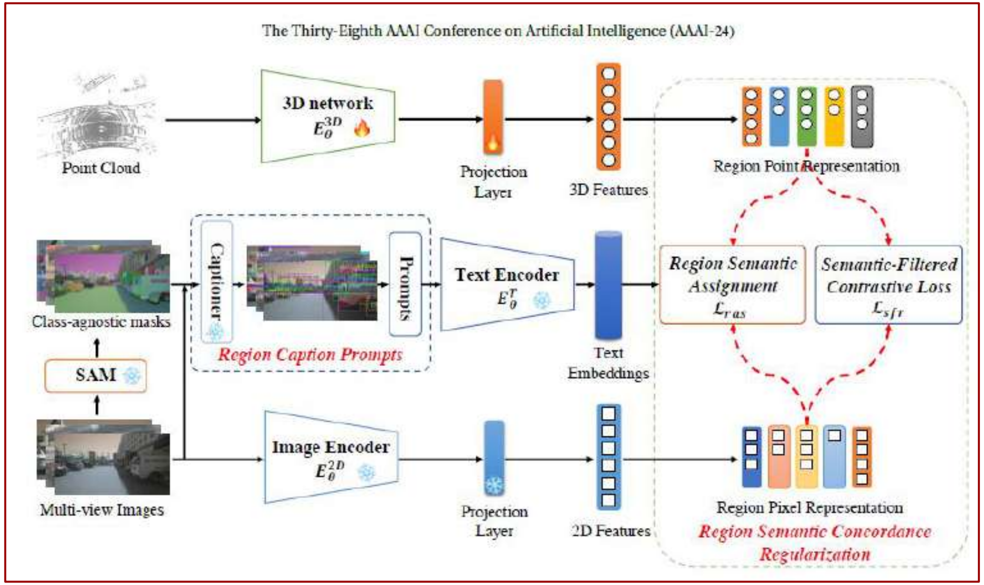
Semantic-Filtered Contrastive Loss

$$\mathcal{L}_{sfr}(P,Q) = -\frac{1}{M}\sum_{i=0}^{M}\log\left(\frac{e^{((p_i\cdot q_i)/\tau)}}{\sum_{j\neq i} T_{ij}\cdot e^{((p_i\cdot q_j)/\tau)} + e^{((p_i\cdot q_i)/\tau)}}\right)$$

Semantic-Filtered Contrastive Loss refines representation learning by preventing "false negatives." Guided by **text descriptions**, it identifies when different segments belong to the same object—like a truck's cab and trailer. It then filters out this incorrect negative pair, ensuring the model learns a coherent object representation.

(Liao et al., 2024)

VLM2Scene: Self-Supervised Image-Text-LiDAR Learning with Foundation Models for Autonomous Driving Scene Understanding

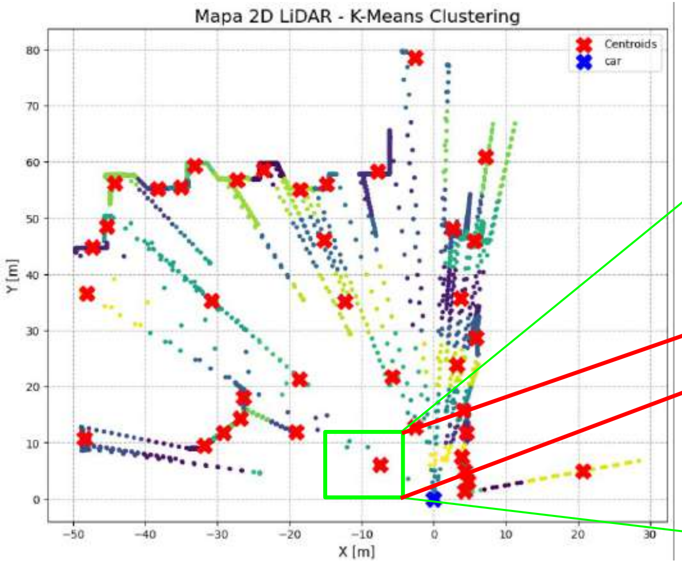


(Liao et al., 2024)

Pre Training

Loss Function

Traditional Contrastive Loss

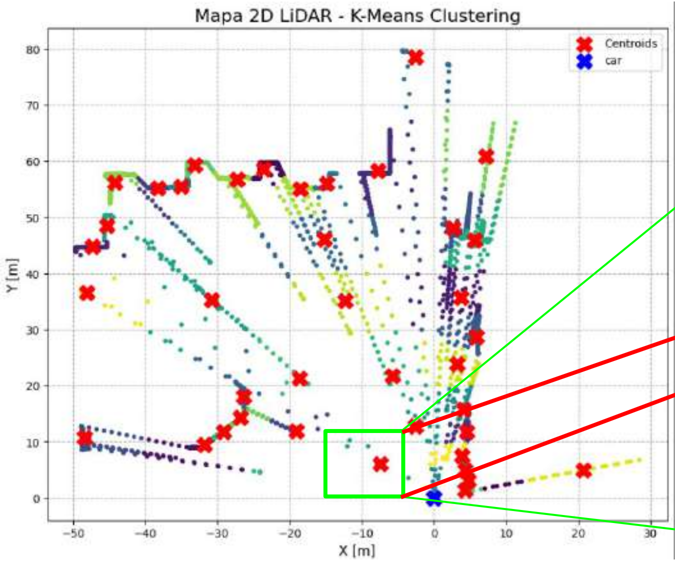


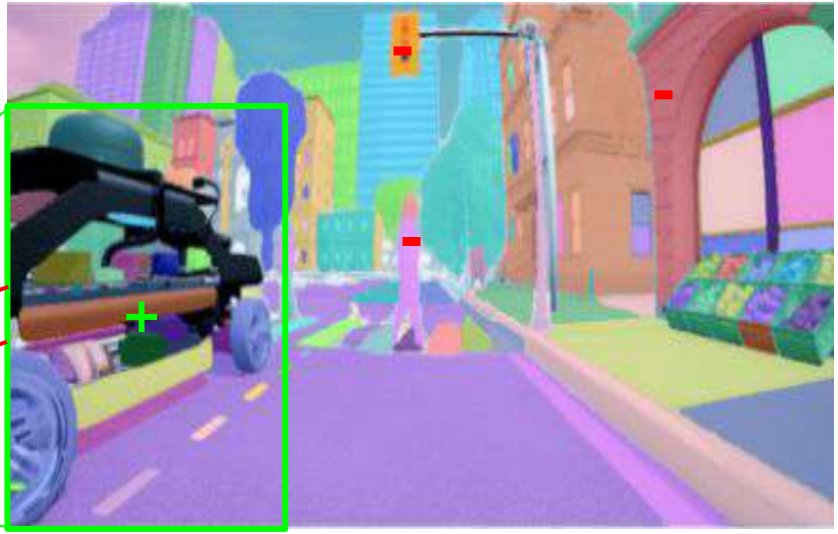
Multiple Sam Segments, but the same class (Car).
Negative Falses Issue

Pre Training

Loss Function

Traditional Contrastive Loss

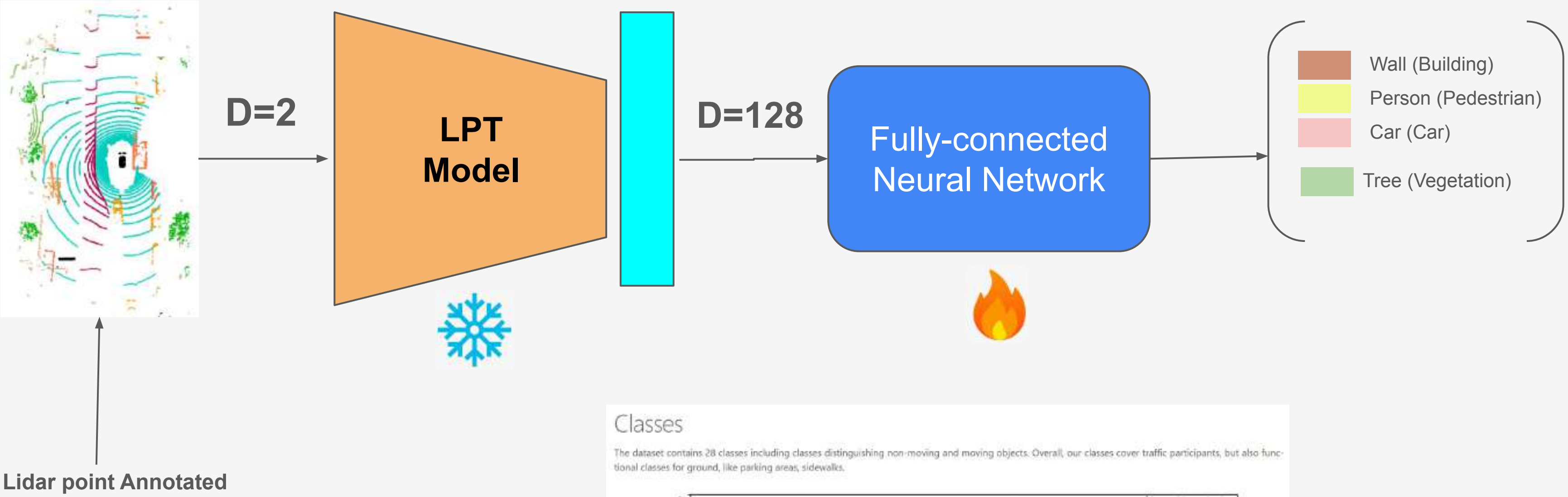


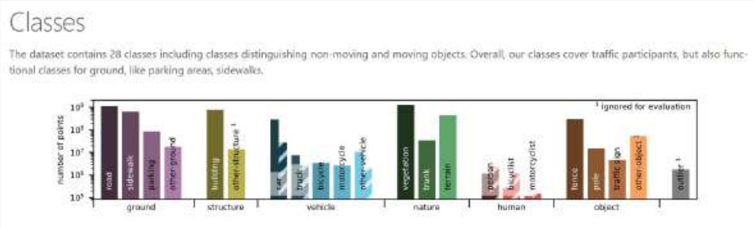


Multiple Sam Segments, but the same class (Car).
Negative Falses Issue

Fine Tuning

Get a labeled cloud points from nuScenes dataset for fine tune the model





Results | Quantitative

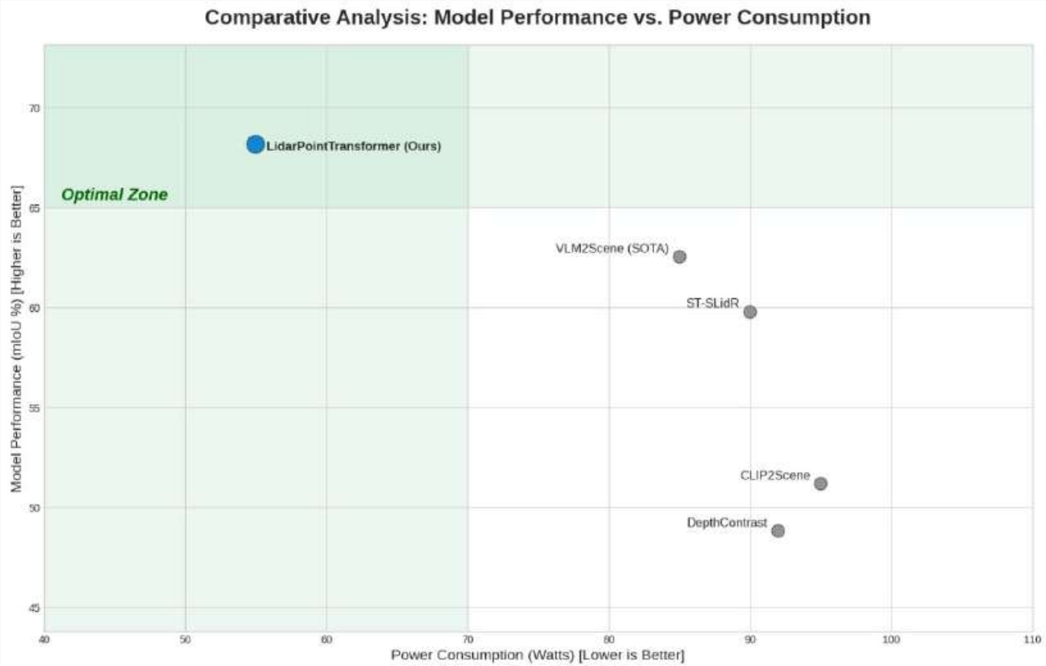
Per-class 2D semantic segmentation IoU performance on the nuScenes vaild set when fine-tuning with **1 % labels**.

Method	**mIoU**	car	pedestrian	vegetation (tree)	wall (Building)
DepthContrast	48.8	67.4	31.6	74.6	21.9
ST-SLidR	59.75	74.5	50.9	82.9	30.7
CLIP2Scene	51.17	70.2	41.3	77.6	15.6
VLM2Scene	62.52	77.7	53.2	83.0	36.2
LidarPointTransformer (ours)	**68.15**	**82.1**	**71.2**	**79.0**	**40.3**

Comparison with other approaches

Aproach	Primary 3D Data	3D Architecture	2D Supervision Source	Learning Level	Main Pretext Task	Trainable Parameters
DepthContrast	3D LiDAR	PointNet++ & Sparse U-Net	3D Data Augmentatio ns	Instance-level	Instance Discrimination	~2.5 M
ST-SLidR	3D LiDAR	Minkowski U-Net	SSL Image Features	Region-level (Superpixels)	Tolerant Contrastive Loss	~25 M
CLIP2Scene	3D LiDAR	SPV CNN / Minkowski Net	Image (CLIP) & Text (Templates)	Point/Pixel-level	Point-Text Contrastive	~28 M
VLM2Scene	3D LiDAR	Minkowski U-Net	Image (CLIP) & Text (BLIP-2)	Region-level (SAM Masks)	Filtered Contrastive + RSA	~25 M
LidarPointTransformer (ours)	**2D LiDAR**	**Point Transformer**	**Image (CLIP) & Text (Captions)**	**Region-level (SAM Masks)**	**Contrastive + Pseudo-Labeling**	**~0.6 M**

Results | Quantitative



$$C_E \approx \eta \cdot (\alpha \cdot f(N) + \beta \cdot g(P_M))$$

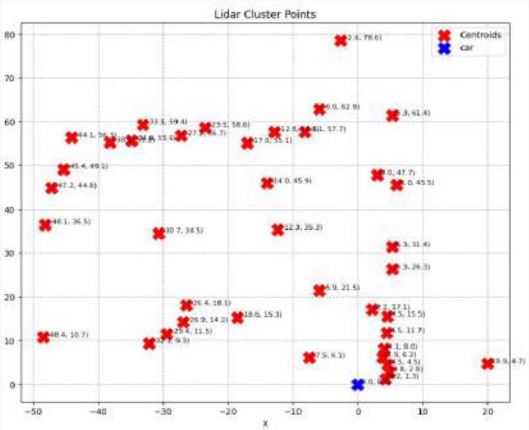
C_E: (Estimated Energy Cost)
n: Hardware efficiency coefficient
N: Number of input elements
P_M: Number of trainable parameters
f(N): Computational complexity

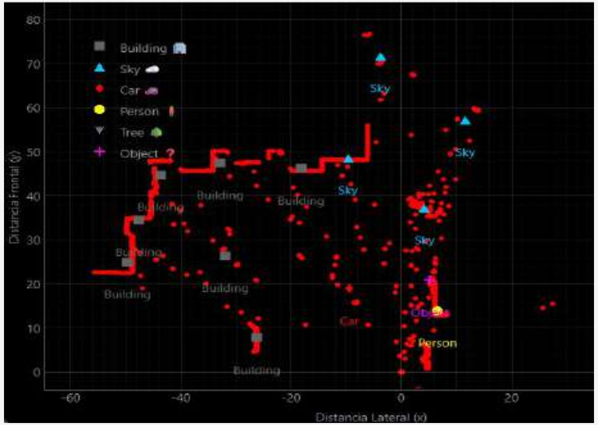
```
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 570.133.07             Driver Version: 570.133.07     CUDA Version: 12.8     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Quadro RTX 6000                Off |   00000000:01:00.0 Off |                  Off |
| 52%   74C    P0            151W /  260W |   18921MiB /  24576MiB |     88%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
```

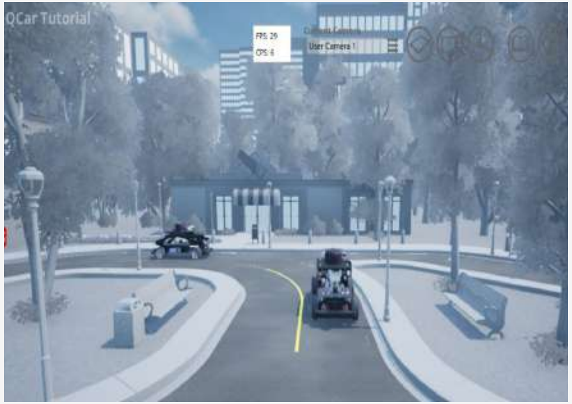
gpu used for training process

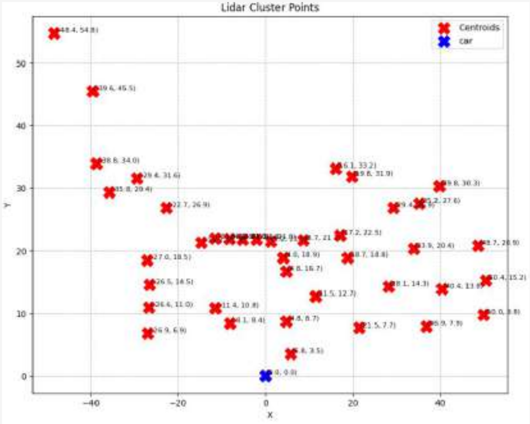
Results | Qualitative

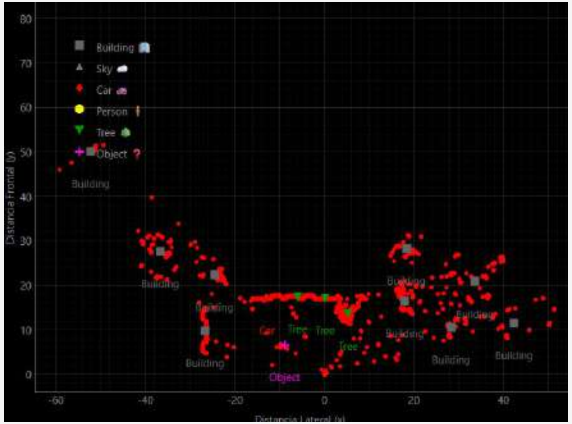
On digital twin data





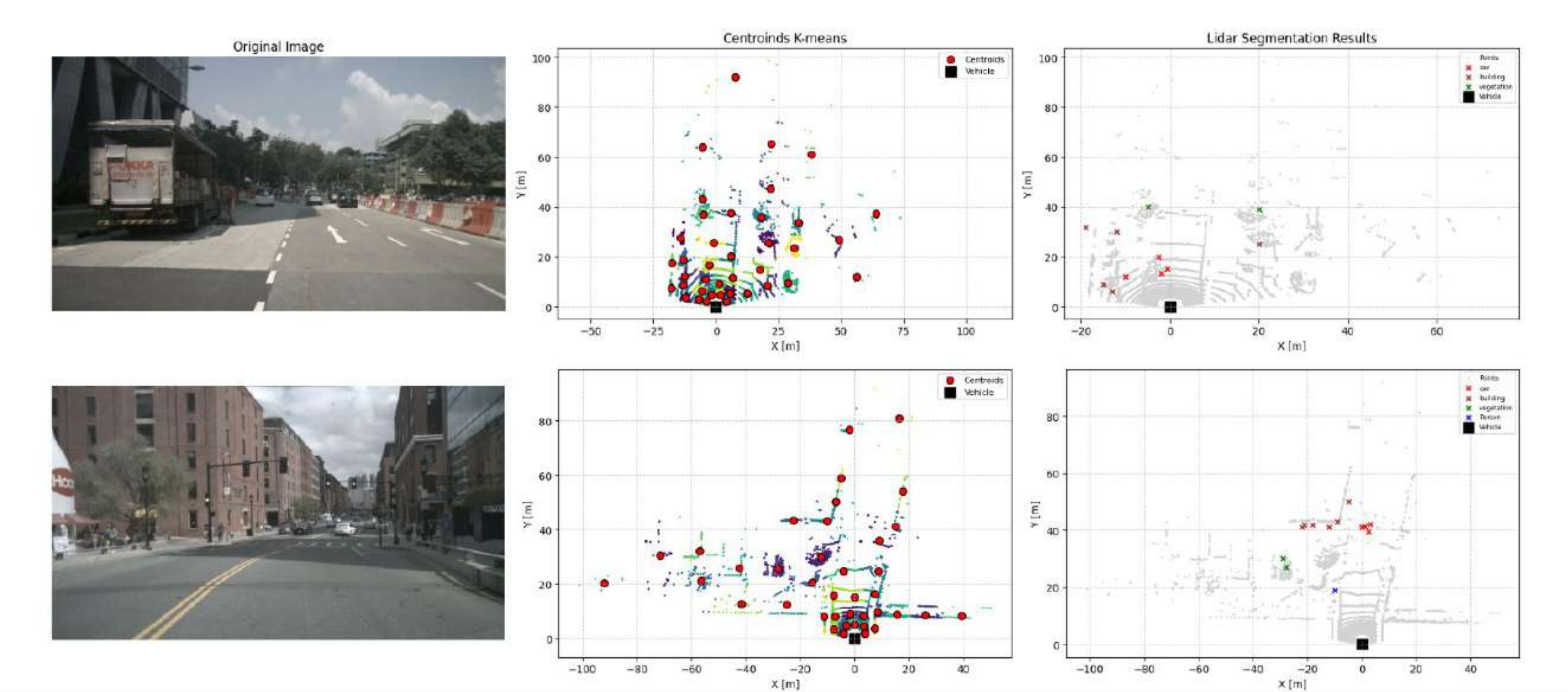






Results | Qualitative

On nuscenes data



Next Activities

- Training the model directly inside the vehicles, ensuring the privacy of LiDAR and image data.
  - Use of federated learning (FL) over our experimental 5G network, reducing communications and computing costs.
- Research papers describing our results.

Blockchain-based reputation system with aggregation auditing centers in smart cities with 5G

Contents

1 - Objective

2 - Motivation and contribution

3 - Basic concept

4 - Related work

5 - Proposal

6 - Experiment

7 - Result

8 - Conclusion

9 - Next steps

Objective

- Distributed system for registering and managing urban incidents.
- Edge architecture with blockchain to ensure data integrity.
- Efficient reputation mechanism to improve the city.
- Improve the quality of life in urban communities.
- Use of sensitive IoT mesh for collecting and managing urban data

Motivation

- Smart cities face challenges with excessive data
- Identifying meaning in the voluminous data for city management is a challenge
- There is high latency for data processing, which makes it outdated and conflicting
- Obtaining reliable and consistent information from various regions is a challenge.
- Sensitive mesh contribute with sustainable in the city

Challenge

- Data generated from diverse sources
- Excess of data transferred
- Mixture of volatile and long-term information replicated in the cloud
- Need to filter and process high volume of data
- Final data may be outdated and conflicting for the application's needs
- Reliability data from multiple sources

Contribution

- Smart Cities will have information to improve the quality of life of citizens even with high regional demands
- Our propose focus on sensitive mesh capable of standardizing information from IoT/machines and humans
- The edge will not only transmit data but also generate information
- The edges will provide more consistent information, avoiding cloud traffic overload and excess storage of volatile data
- Blockchain will help with the consistency of information between components
- Auditing city using report and vote mechanism

Basic Concept

Smart City

- **Definição Smart City**

  Aveiro Tech City Living Lab

- **Vehicular Ad Hoc Network**

  Road side Unit

  On board side Unit

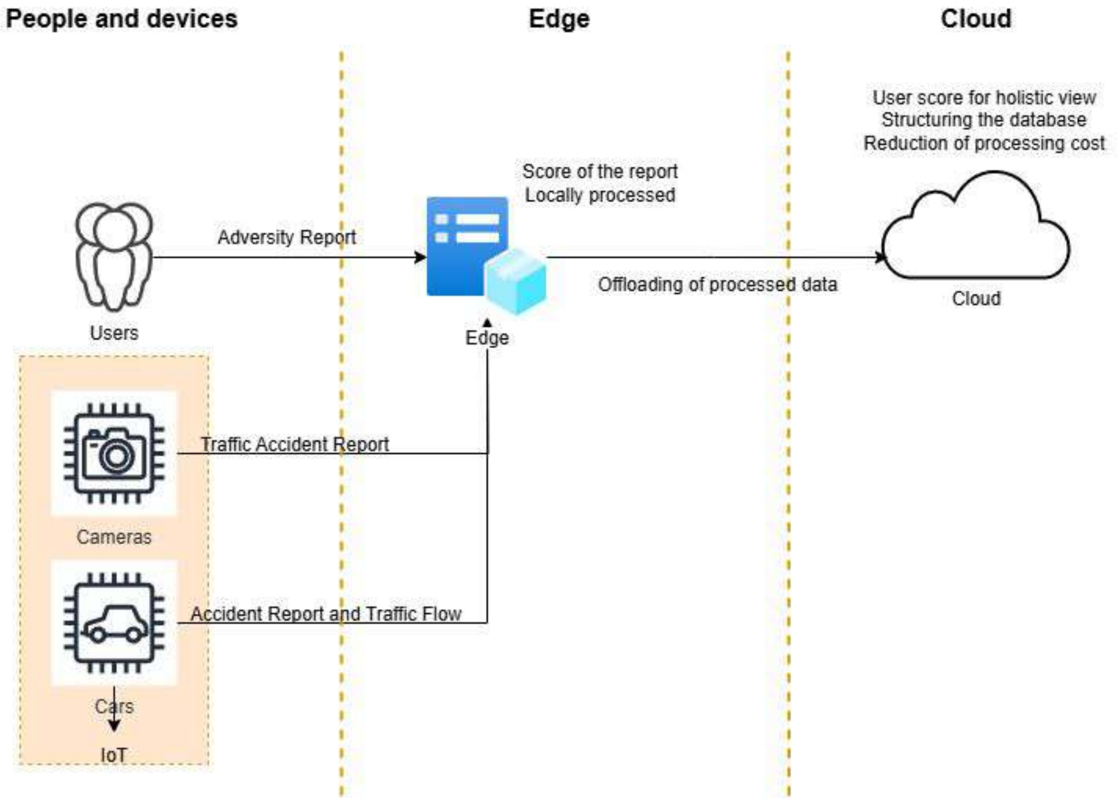
- **Communication technologies**

  ITS-G5 (WAVE), WiFi and Ethernet

Proposal

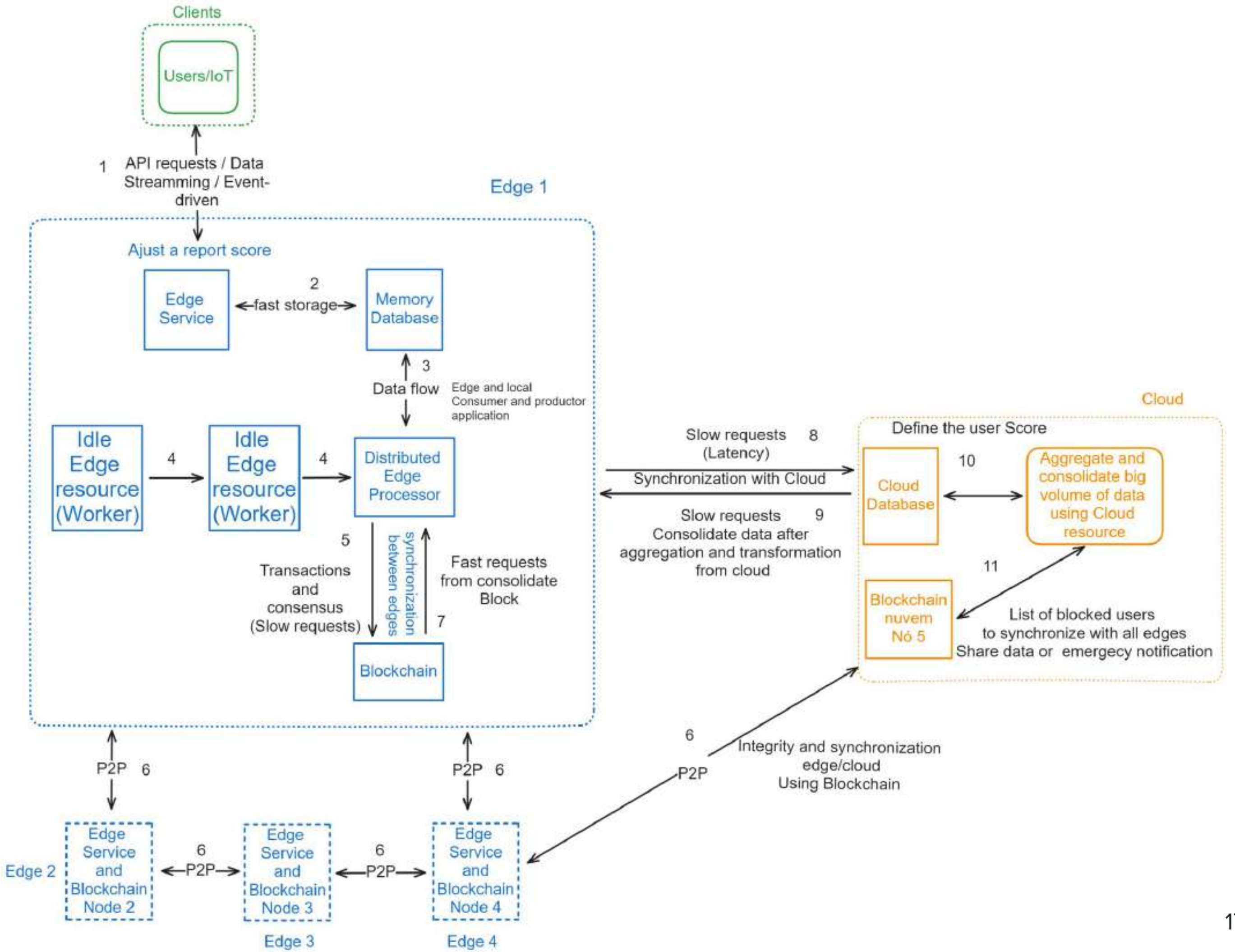
- Distributed system for registering and managing urban incidents.
- Edge architecture with blockchain to ensure data integrity.
- Efficient reputation mechanism to improve the city.
- Improve the quality of life in urban communities.
- Use of sensitive IoT mesh for collecting and managing urban data

Proposal



People and devices
Edge
Cloud
User score for holistic view
Structuring the database
Reduction of processing cost
Score of the report
Locally processed
Adversity Report
Offloading of processed data
Cloud
Users
Edge
Traffic Accident Report
Cameras
Accident Report and Traffic Flow
Cars
IoT

Proposal


Clients
Users/IoT
1 API requests / Data Streamming / Event-driven
Edge 1
Ajust a report score
Edge Service
2 fast storage
Memory Database
3 Data flow
Edge and local Consumer and productor application
Idle Edge resource (Worker)
4
Idle Edge resource (Worker)
4
Distributed Edge Processor
5 Transactions and consensus (Slow requests)
synchronization between edges
Fast requests from consolidate Block
7
Blockchain
Slow requests (Latency) 8
Synchronization with Cloud
Slow requests 9 Consolidate data after aggregation and transformation from cloud
Cloud
Define the user Score
Cloud Database
10
Aggregate and consolidate big volume of data using Cloud resource
11
Blockchain nuvem Nó 5
List of blocked users to synchronize with all edges
Share data or emergecy notification
P2P 6
P2P 6
6 P2P
Integrity and synchronization edge/cloud Using Blockchain
Edge 2
Edge Service and Blockchain Node 2
6 P2P
Edge Service and Blockchain Node 3
6 P2P
Edge Service and Blockchain Node 4
Edge 3
Edge 4

Proposal

Topology

- **Cloud components**

  On-premise

  Cloud services

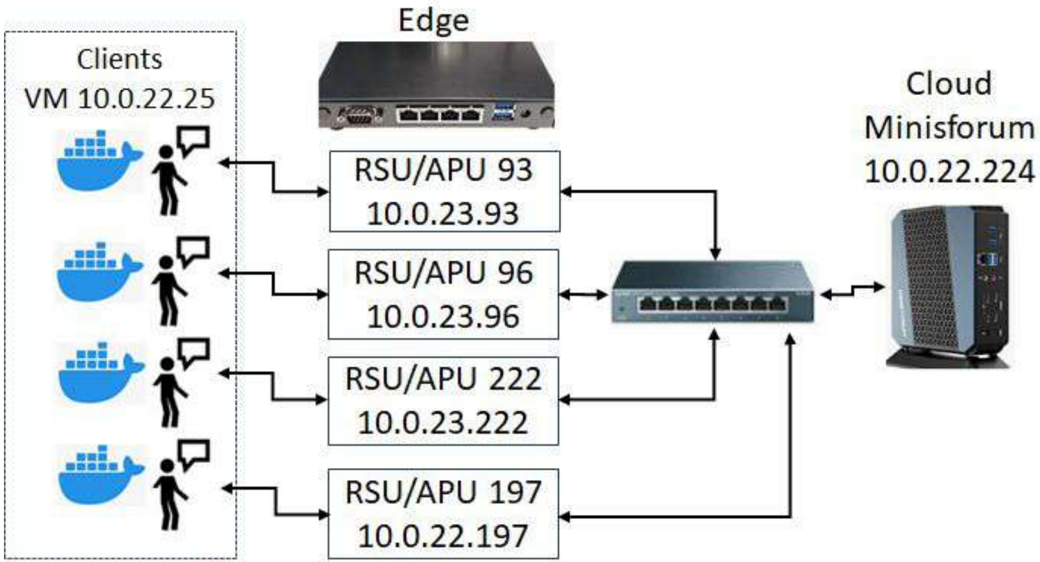
- **Edge components**

  Road side Unit

- **Client components**

  On board side Unit

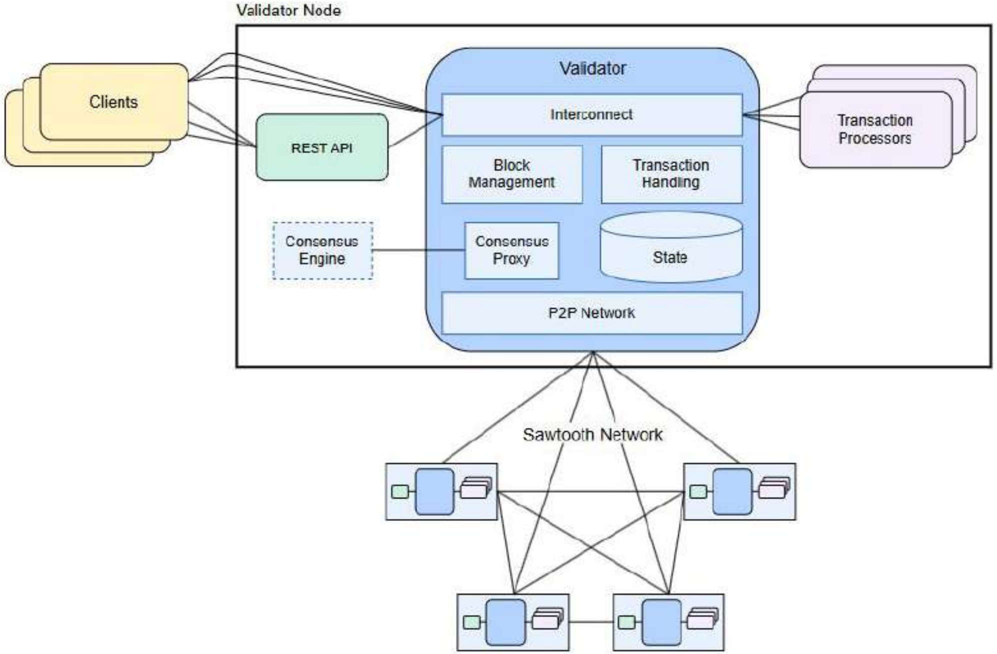
  Smartphones



Proposal

Blockchain - Sawtooth Hyperledger

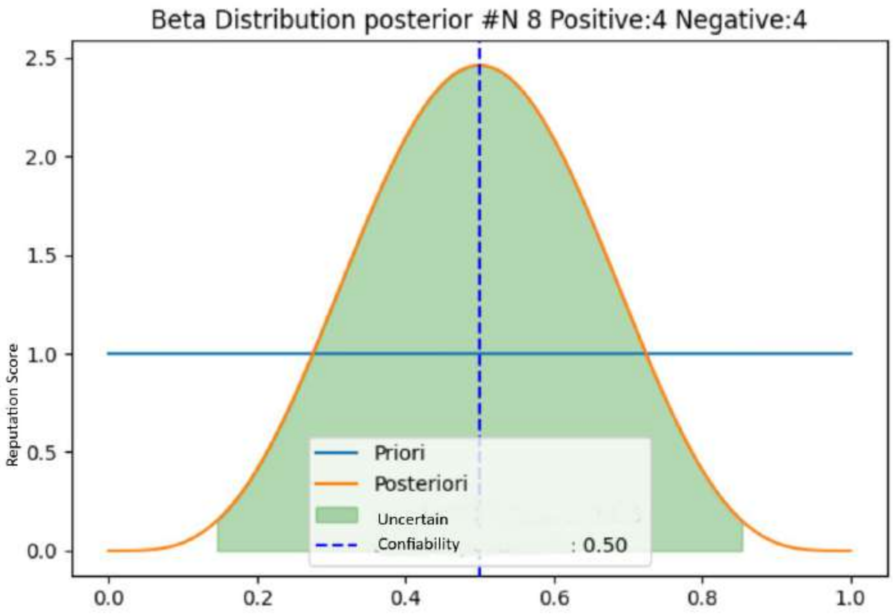
- Transaction Processors
- REST API
- Validator
- Consensus Engine
- Container shell



Proposal

Reputation Score

- Report reputation score using Bayes Statistic
- Update posterior for each time see new data
- This report score compose the user reputation score

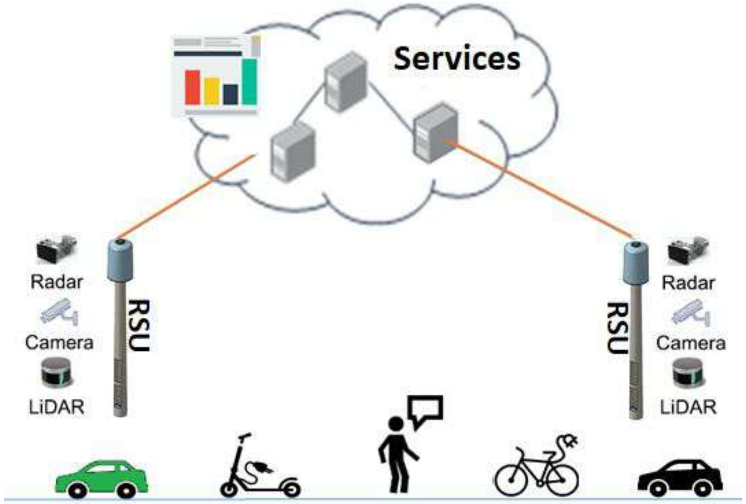


Application Scenario

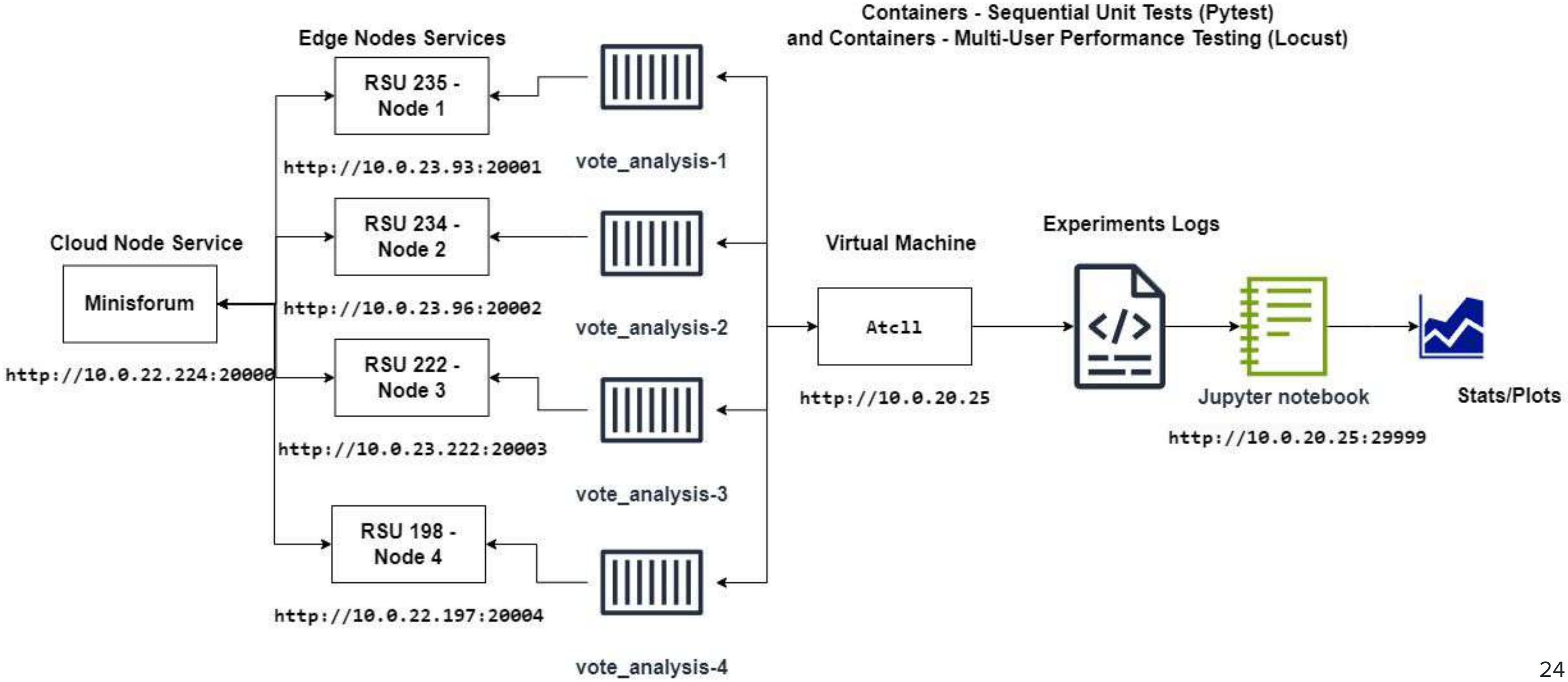
Baseline: Single interaction test comparing system performance without communication stress and evaluating memory vs. SQL database communication.

Concurrent Edge Interaction: Performance analysis using pytest to measure concurrent interactions, comparing stress under report creation and voting scenarios, and testing parallelism across nodes.

IoT Client Evaluation: Stress test using Locust to simulate 50 users interacting with edge systems, analyzing behavior under high-load conditions.



Application Scenario



Result

Prioritize In-Memory Databases: Use in-memory databases for real-time operations (e.g., voting, reporting) to reduce latency.

Hybrid Architecture: Implement a hybrid solution to balance speed and reliability.

Stress Testing: Continuously monitor and test performance under burst traffic to ensure alignment with system goals.

Automate Scaling: Deploy auto-scaling mechanisms to dynamically allocate resources based on load, especially during peak times.

Table 2. Operation time of container interaction on desktop.

Operation	count	mean	median	std	min	max
user	753.0	0.24	0.24	0.02	0.01	0.37
report	1.0	0.01	0.01	0.00	0.01	0.01
reputation/vote	751.0	0.01	0.01	0.0	0.01	0.04
wallet/test2	1.0	0.0	0.0	0.00	0.0	0.0

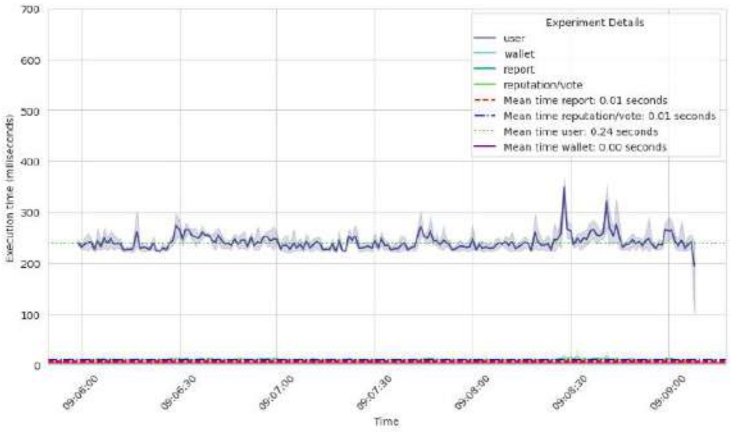


Figure 4. Sequential interaction in personal desktop.

Result

Concurrency Efficiency: The system handles concurrent edge interactions without significant performance degradation, even under high load.

Database Scalability: Memory-based databases (e.g., in-memory DB) are robust to parallelism, making them suitable for distributed edge architectures.

Design Implication: This result supports the use of scalable, memory-optimized databases in edge systems to ensure consistent performance during bursts of concurrent requests.

Table 4. Testing 1000 interactions with a client, edge, and cloud.

Omnimo	Operation	Count	Mean (s)	Std (s)
APU 93	user	1001.0	1.18	0.02
	reputation/vote	1000.0	0.07	0.01
	report	1.0	0.04	0.00
	wallet	1.0	0.02	0.00

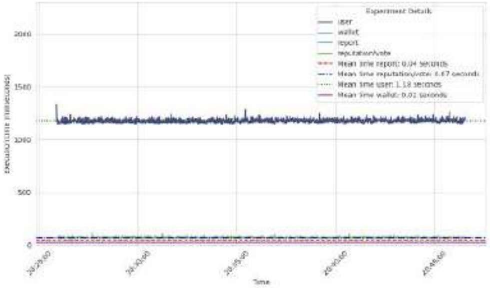


Figure 5. Performance over 1000 interactions with a client, edge, and cloud.

Table 5. Statistics for 4 clients concurrency.

Omnimo	Operation	Count	Mean (s)	Std (s)
APU 93	user	1001.0	1.16	0.01
	reputation/vote	1000.0	0.07	0.01
	report	1.0	0.05	0.00
APU 96	user	1001.0	1.07	0.01
	reputation/vote	1000.0	0.08	0.01
	report	1.0	0.04	0.00
APU 222	user	1001.0	1.07	0.01
	reputation/vote	1000.0	0.08	0.01
	report	1.0	0.04	0.00
APU 197	user	1001.0	1.28	0.07
	reputation/vote	1000.0	0.08	0.01
	report	1.0	0.05	0.00

Conclusion

- Distributed system for registering and managing urban incidents.
- Edge architecture with blockchain to ensure data integrity.
- Efficient reputation mechanism to improve the city.
- Improve the quality of life in urban communities.
- Use of sensitive IoT mesh for collecting and managing urban data

Next Step

- Testing new topology architectures
- Analysis wave communication
- Street test and integration with Waze
- Integration with accident detection

Questions?

Thanks !!!