ML-enabled Service Discovery for Microservice Architecture: A QoS Approach

Karthik Vaidhyanathan, Mauro Caporuscio, Stefano Florio, and Henry Muccini

The 39th ACM/SIGAPP Symposium On Applied Computing
Track: Software Architecture: Theory, Technology, and Applications

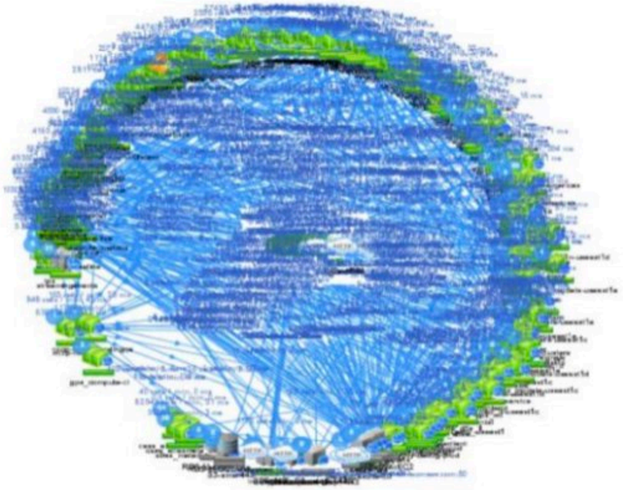
April 9, 2024

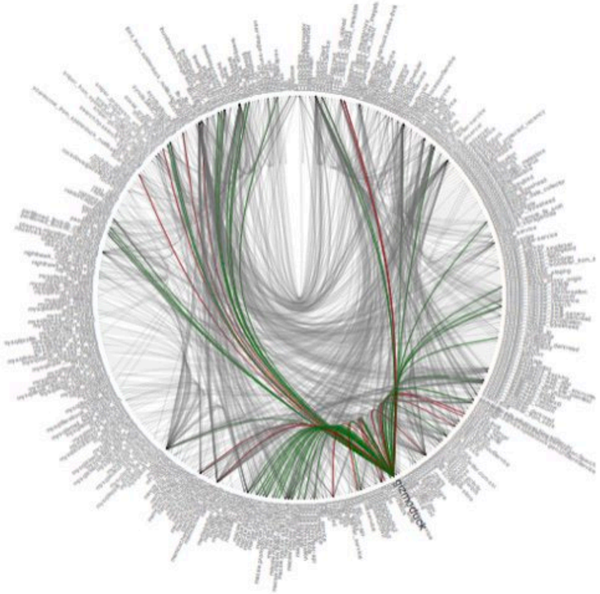
Association for Computing Machinery

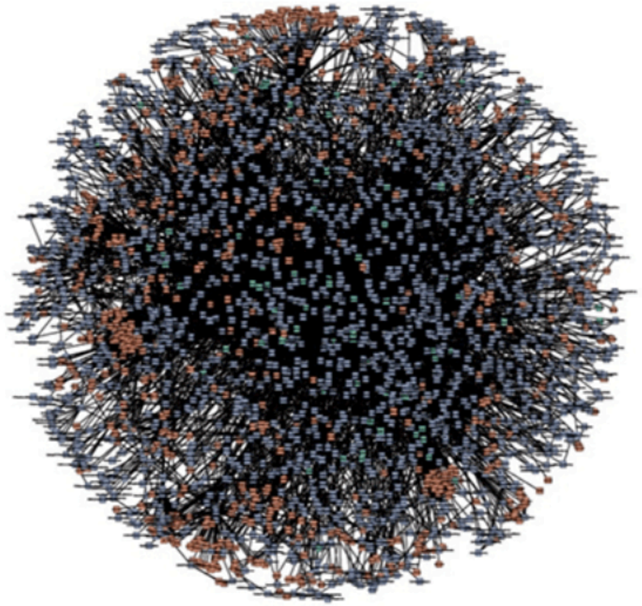
SIGAPP

The World of Microservices

"It is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API" - Martin Fowler







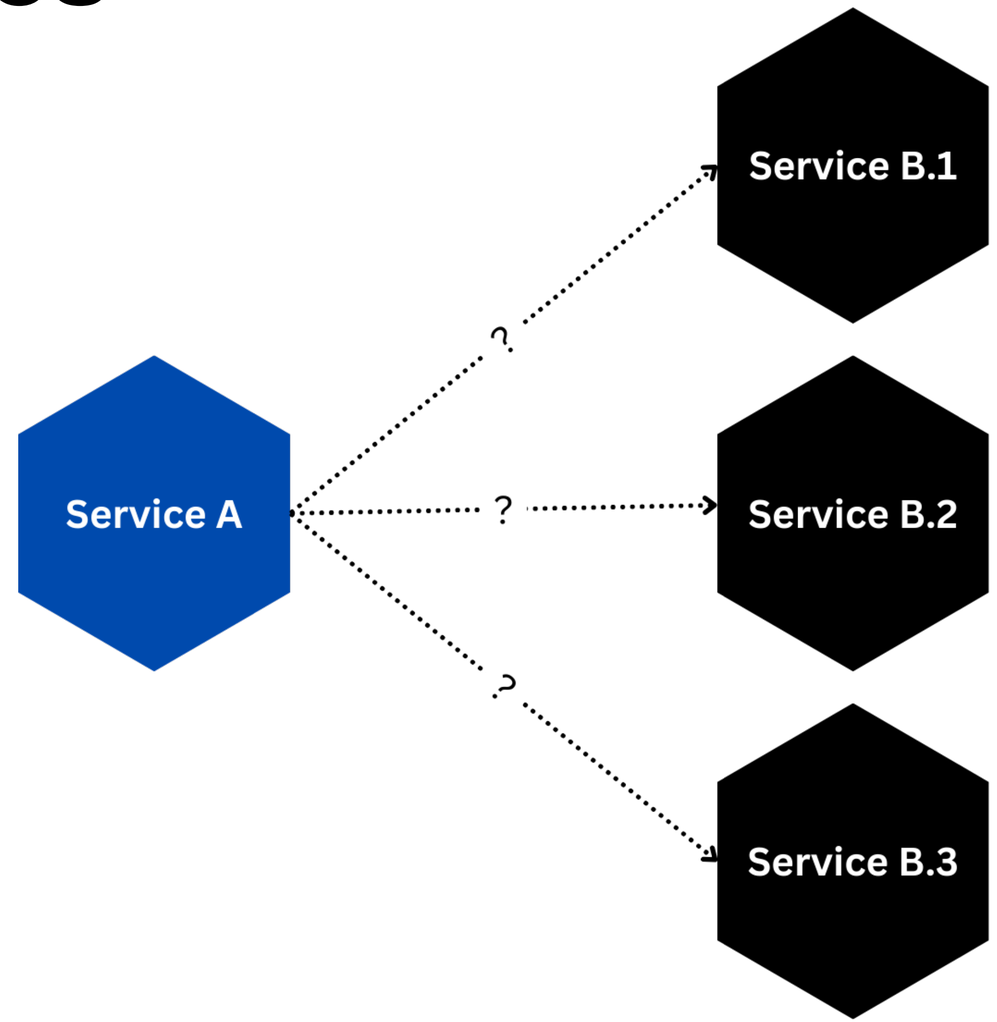
NETFLIX

amazon.com

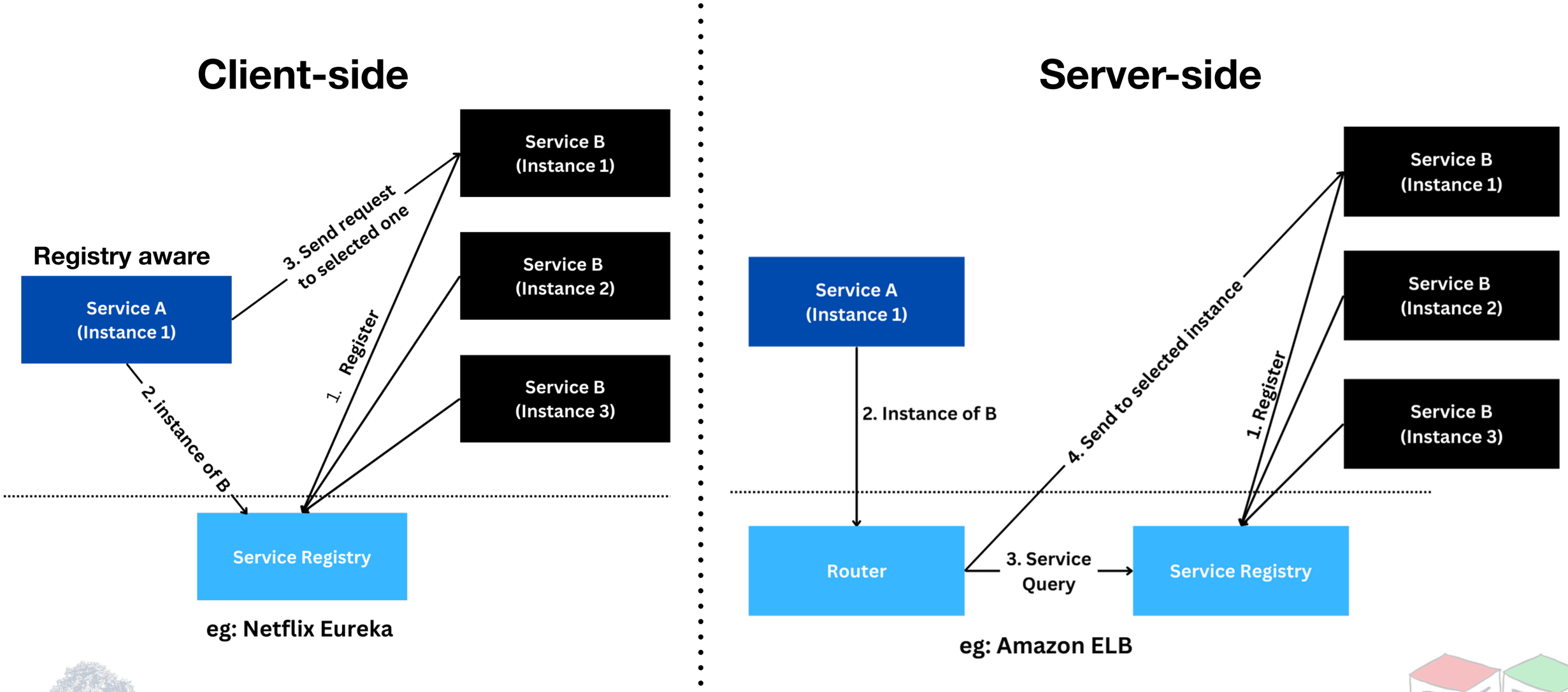
source: tinyurl.com/3kswbtak (Twitter, Google images)

Communication Challenges

- Hundreds of microservices may be composed to form a complex architecture
- Tens of instances of the same microservice can run on different servers
- The number or locations of running instances could change very frequently
- The set of service instances changes dynamically because of autoscaling, failures, and upgrades



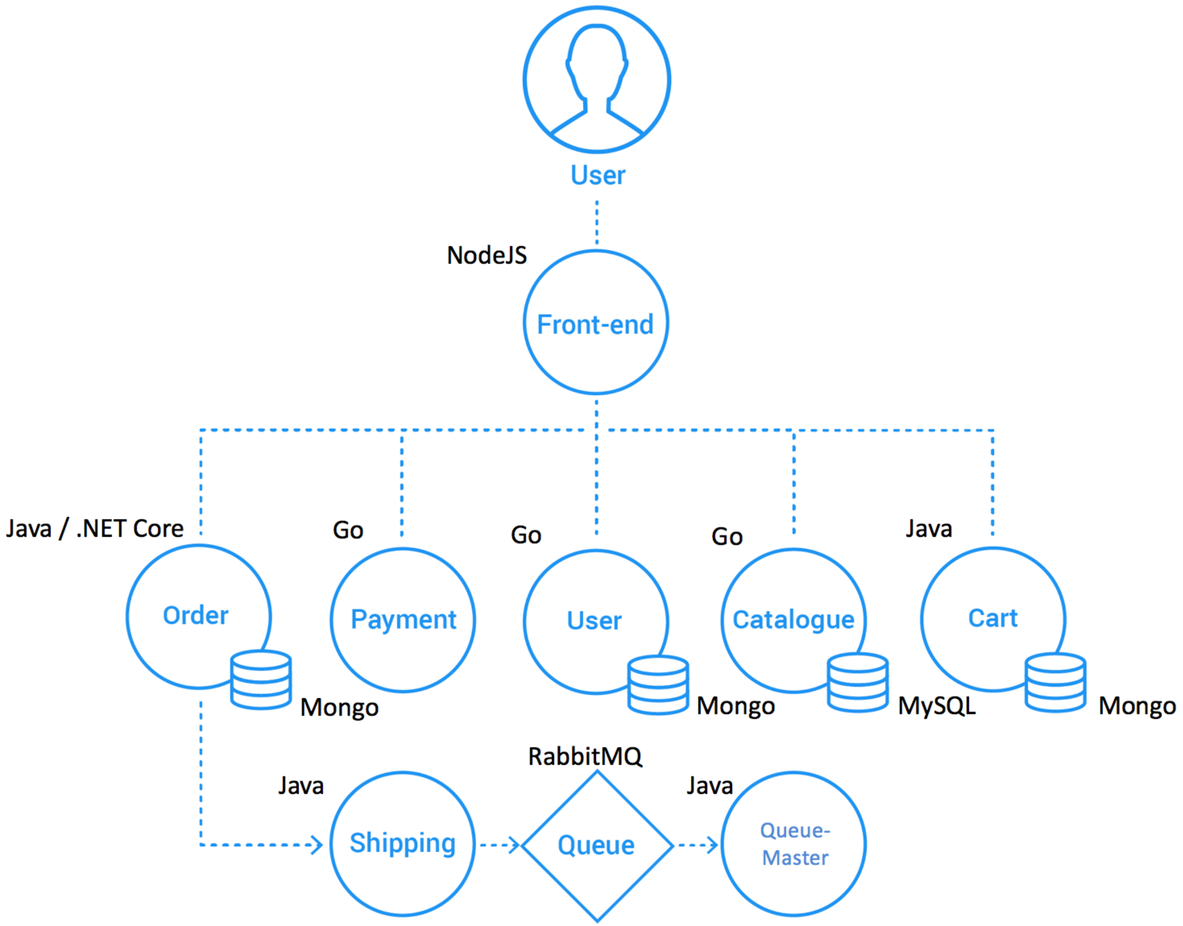
Service Discovery

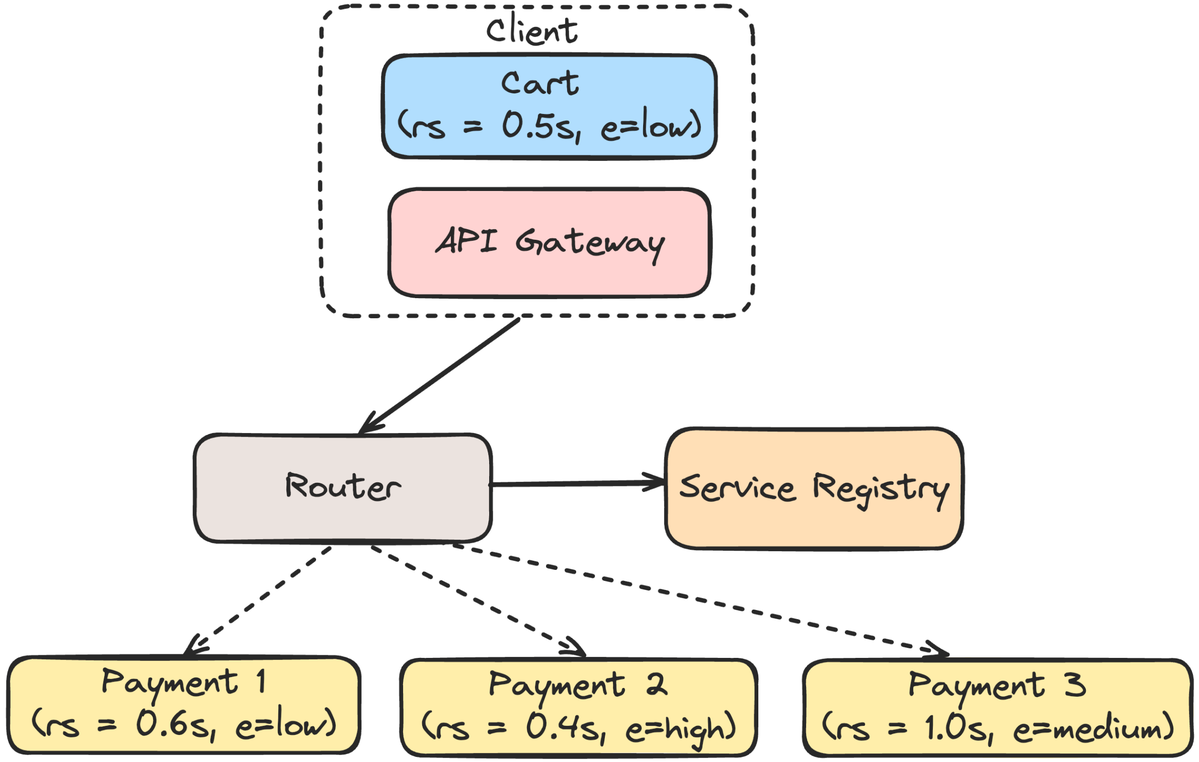


What about the QoS of selected instance?

Lets get more concrete!

The Sockshop System

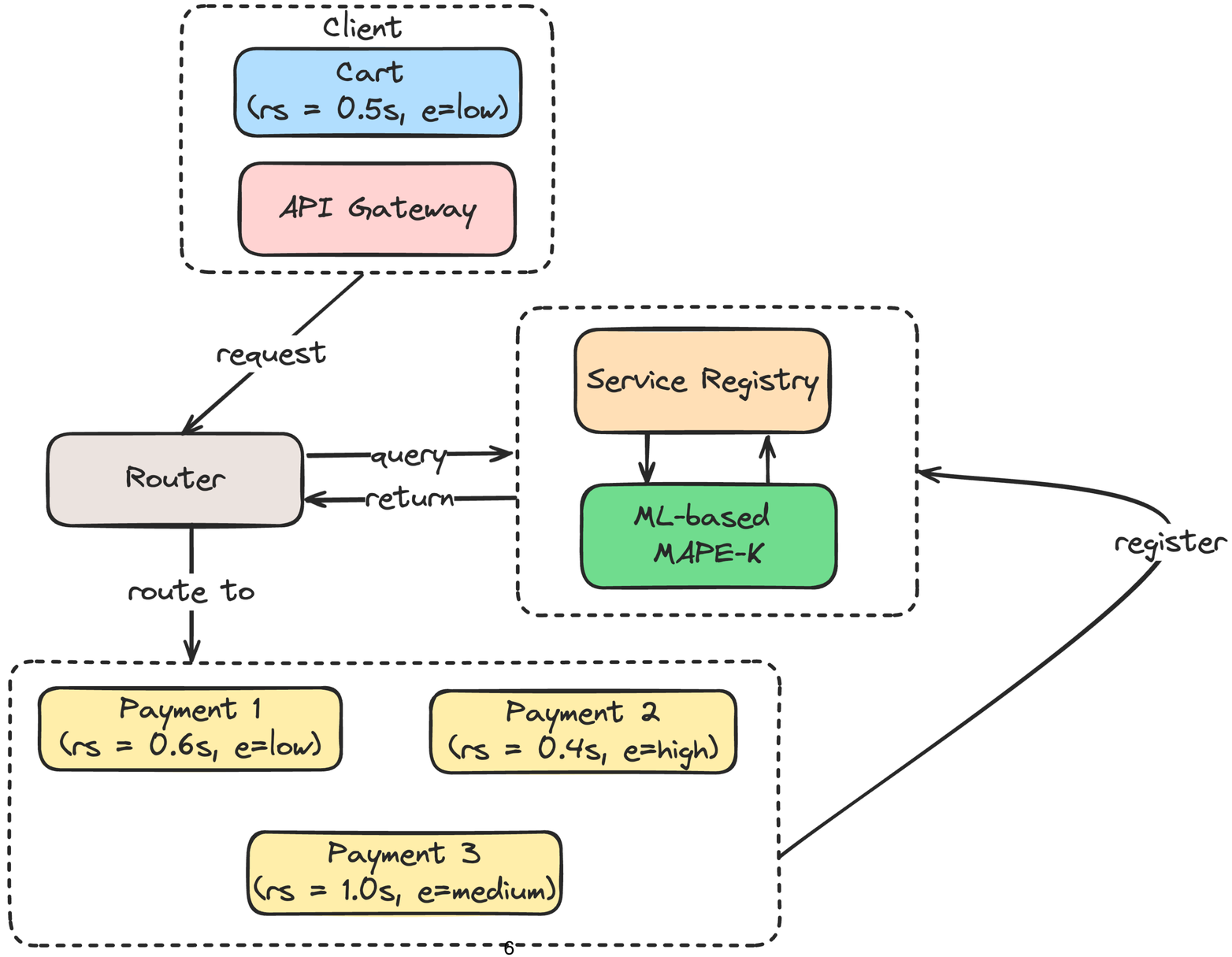




How to guarantee QoS during Service selection (energy vs response time) ?

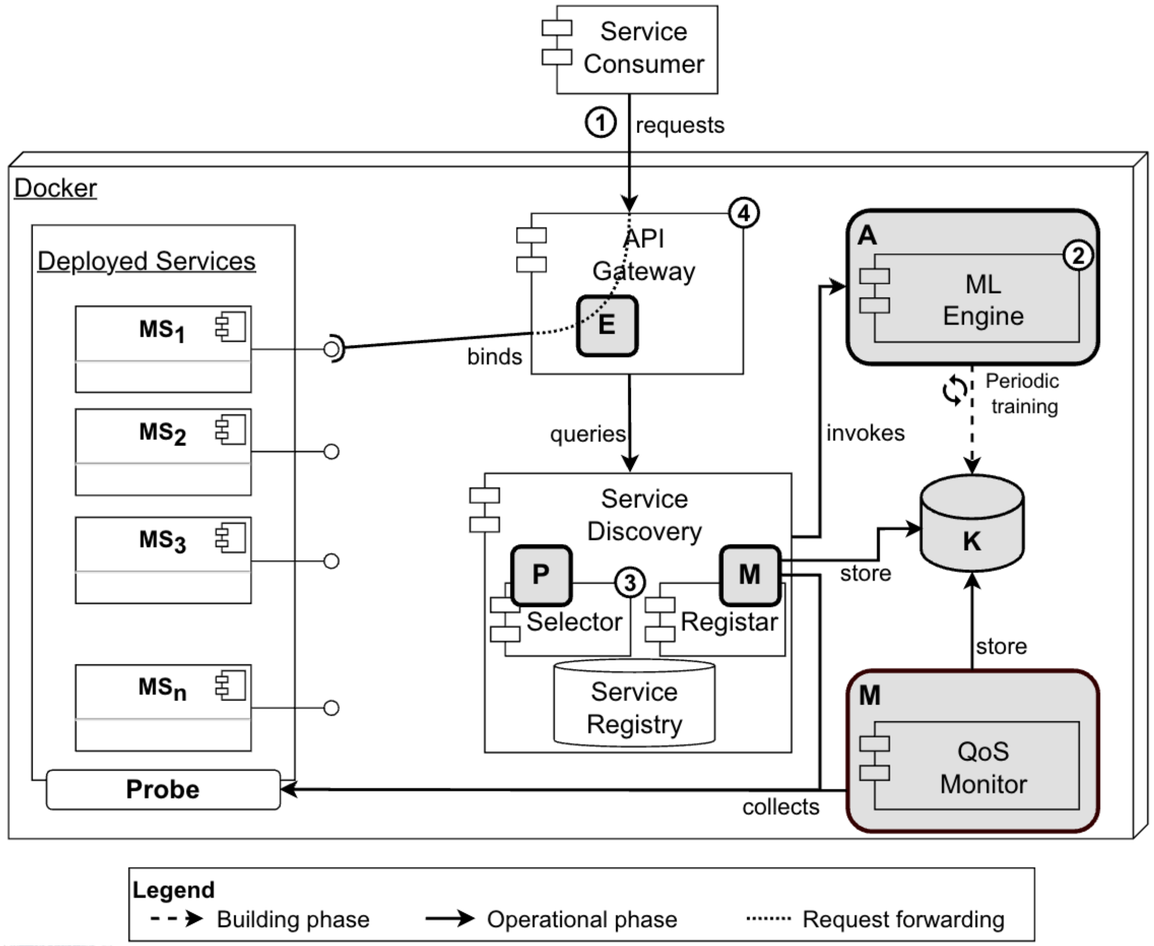
More info: https://github.com/microservices-demo/microservices-demo

Self-adaptation in the action!



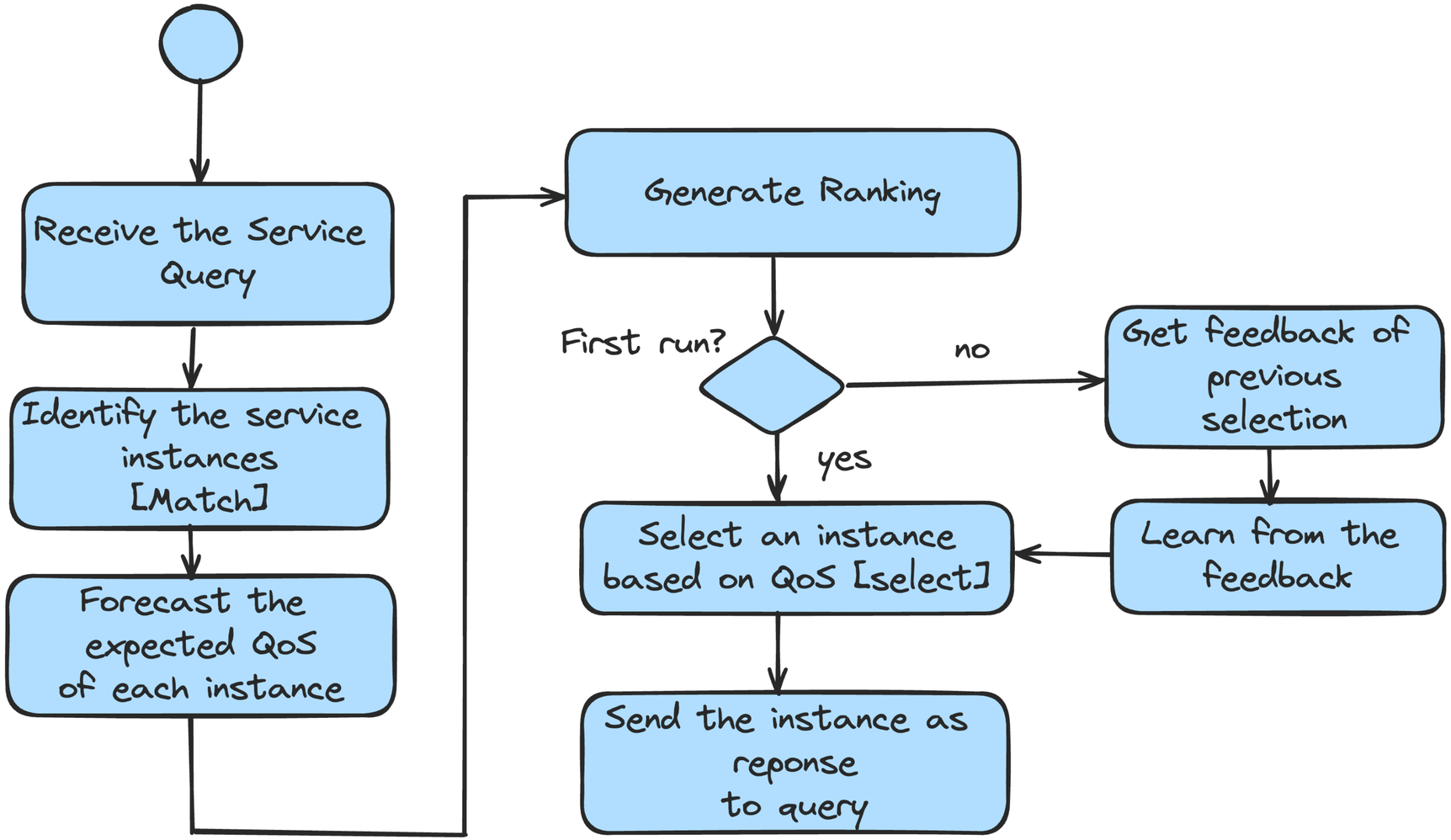
What we propose?

ML-enabled Service Discovery through MAPE-K loop

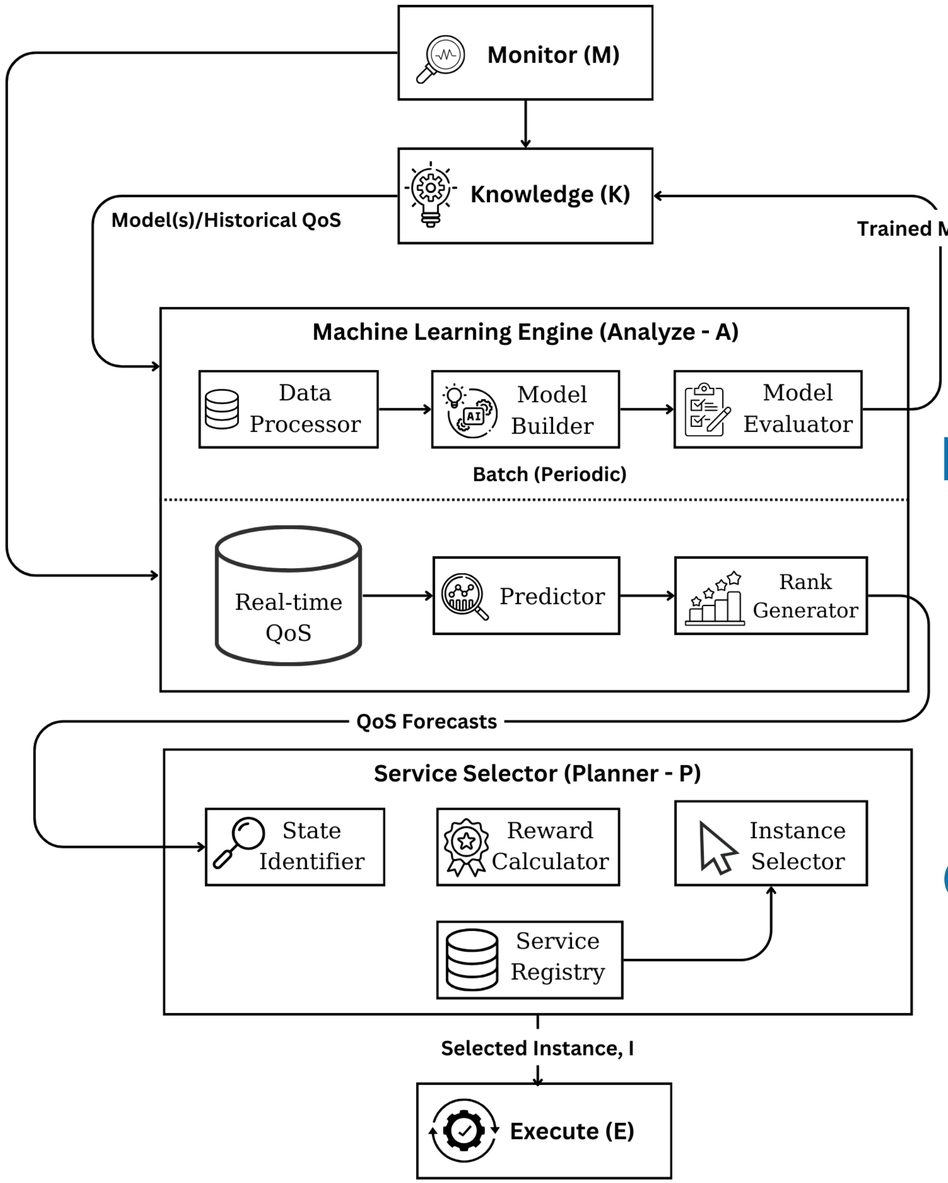


- Microservices deployed as docker containers
- **Monitor** the QoS values and store in **Knowledge (K)**
- **Analyze** using the support of ML engine
- ML Engine works in two modes
  - Build phase (training)
  - Operational phase (inference) - Analyze
  - K stores the QoS values and ML models
- **Plan** involves selection through RL
- **Execution** is the binding to new instance

Approach Overview



Supervised combined with Reinforcement

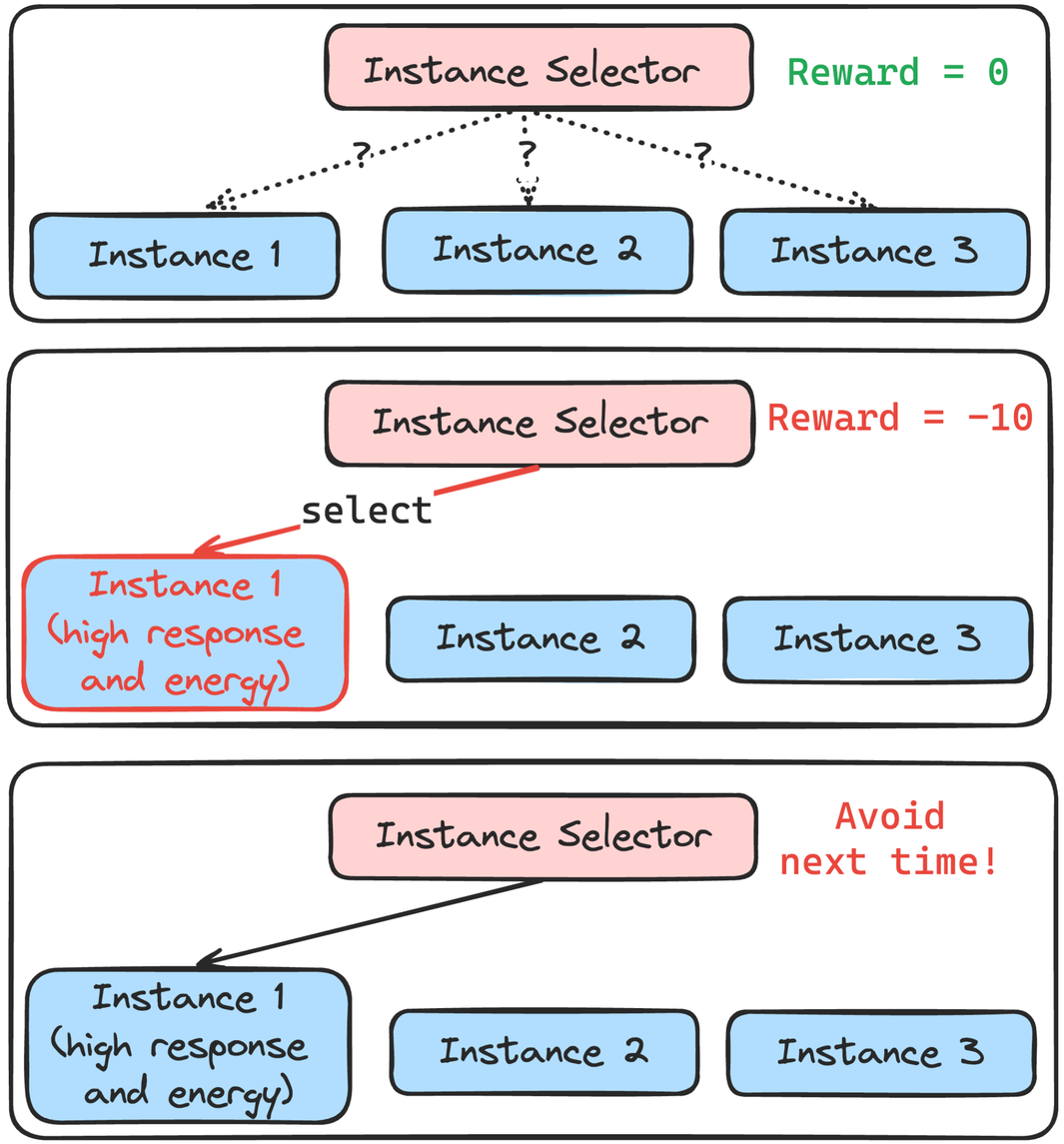


LSTM - Time series forecasts

Q-learning - selection of instance

Caporuscio, M., Toma, M.D., Muccini, H. and Vaidhyanathan, K.. **A Machine Learning Approach to Service Discovery for Microservice Architectures.** ECSA 2021

Q-Learning for Selection



1 Observe

Initialize
Q-table
SXA table
S: states
A: actions

2 Select an instance using a policy (Epsilon Greedy)

3 Return the instance

R = x1*r(p1) + x2*r(p2) +.. xn*r(pn)

4 Get reward or penalty from the next forecast

5 Update the learning Q[s,a]

6 Continue the process

Experiment Setup and Goals

- Each microservice in Sockshop replicated to six instances - total 30 microservices
- Machine with 16 GB RAM and 3.2 Ghz processor
- FIFA 98 world cup trace for request simulation, training data: 1 week (each min)
- Five evaluation candidates
  - **Naive** - always select the same instance
  - **Round robin** - keep moving across the instances
  - Three strategies that use **Q-learning (balanced, response time, energy)**
- **For q-learning, learning factor: 0.08 and discount factor: 0.9**

Git link: https://github.com/karthikv1392/ML_SD_Tradeoff

Evaluations - Defining the Utility

$$U_q = w_e \cdot E_q + w_r \cdot R_q,$$

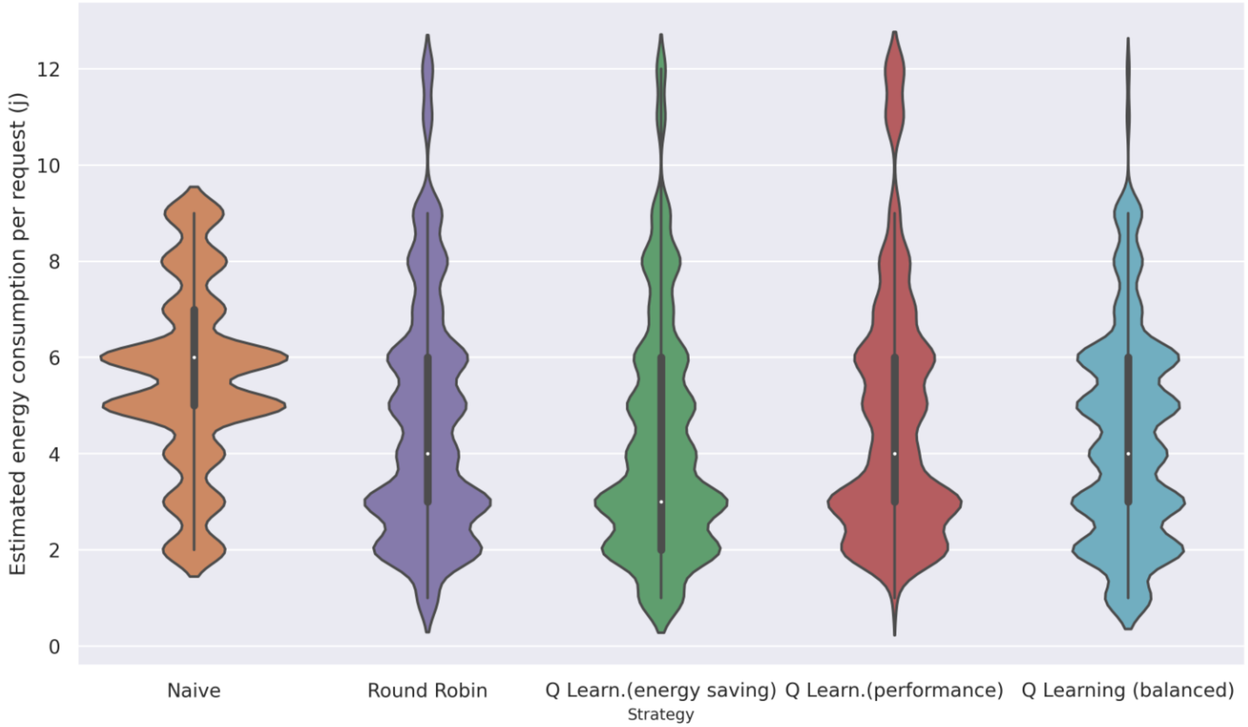
$$R_q = \begin{cases} (R_{max} - r_q) \cdot p_{rv} & \text{if } r_q \geq R_{max} \\ r_q - R_{min} & \text{if } R_{max} > r_q > R_{min} \\ (r_q - R_{min}) \cdot p_{rv} & \text{if } r_q \leq R_{min} \end{cases}$$

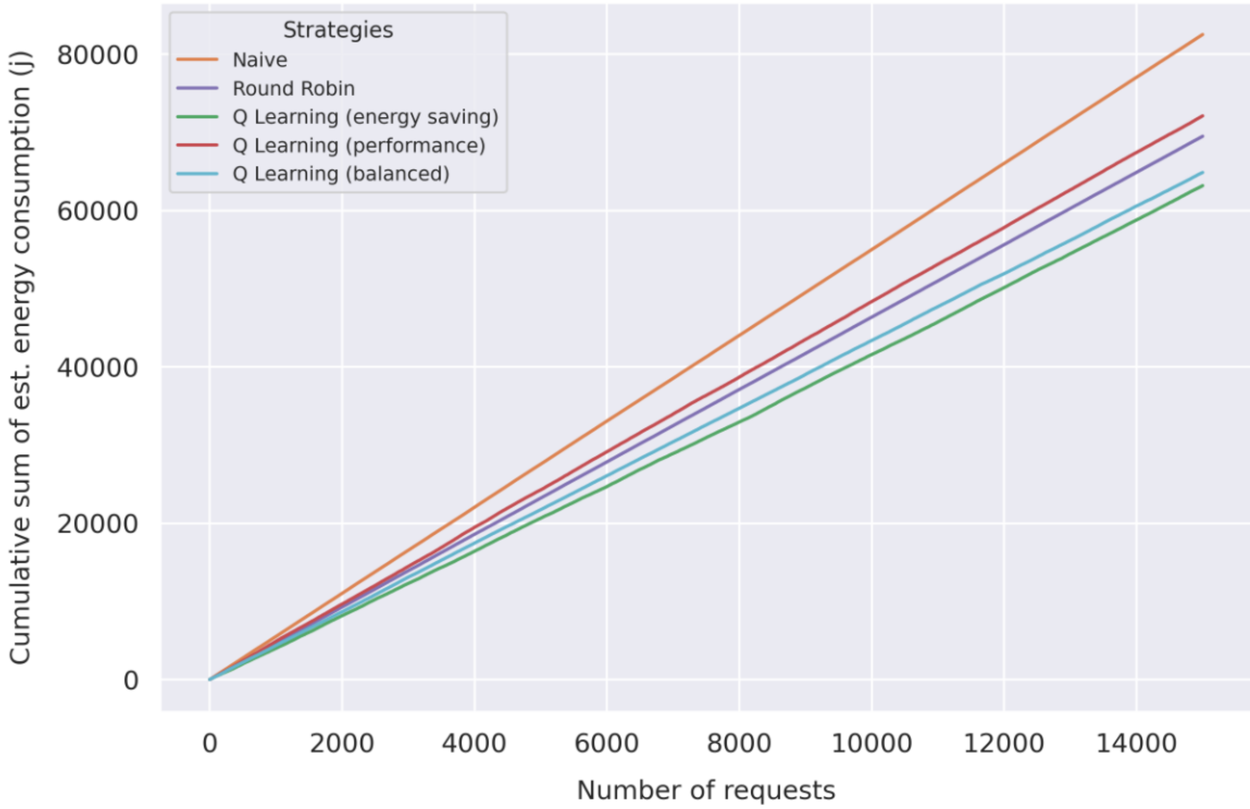
$$E_q = \begin{cases} E_{max} - e_q & \text{if } e_q < E_{max} \\ (E_{max} - e_q) \cdot p_{ev} & \text{otherwise} \end{cases}$$

Parameter	Description	Value
τ	Time period	60 secs
p_{ev}	Penalty for energy violation	0.8
p_{rv}	Penalty for Response time	0.8
R_{max}	Maximum response time	0.8
R_{min}	Minimum Response time	0.2
E_{max}	Maximum energy	1.45 joules
w_r	Weight on response time	0.4
w_e	Weight on energy	0.6

Energy Consumption Effectiveness

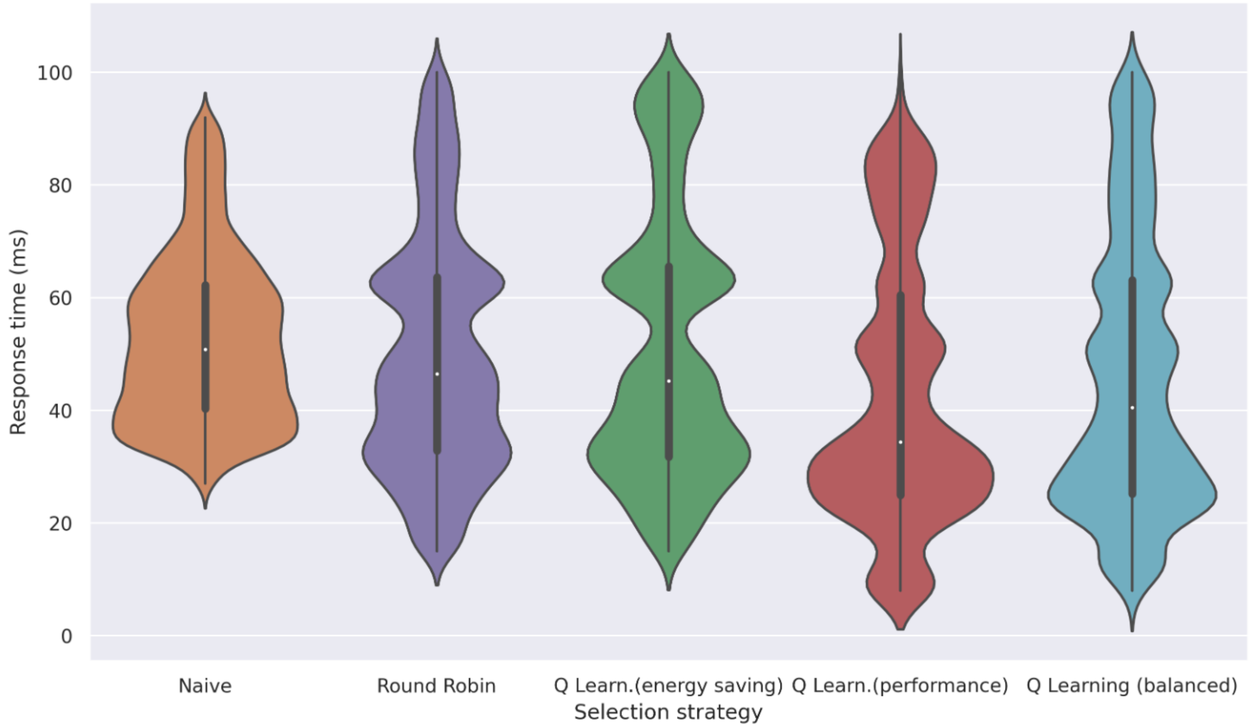
Comparison of energy consumption while using each approach

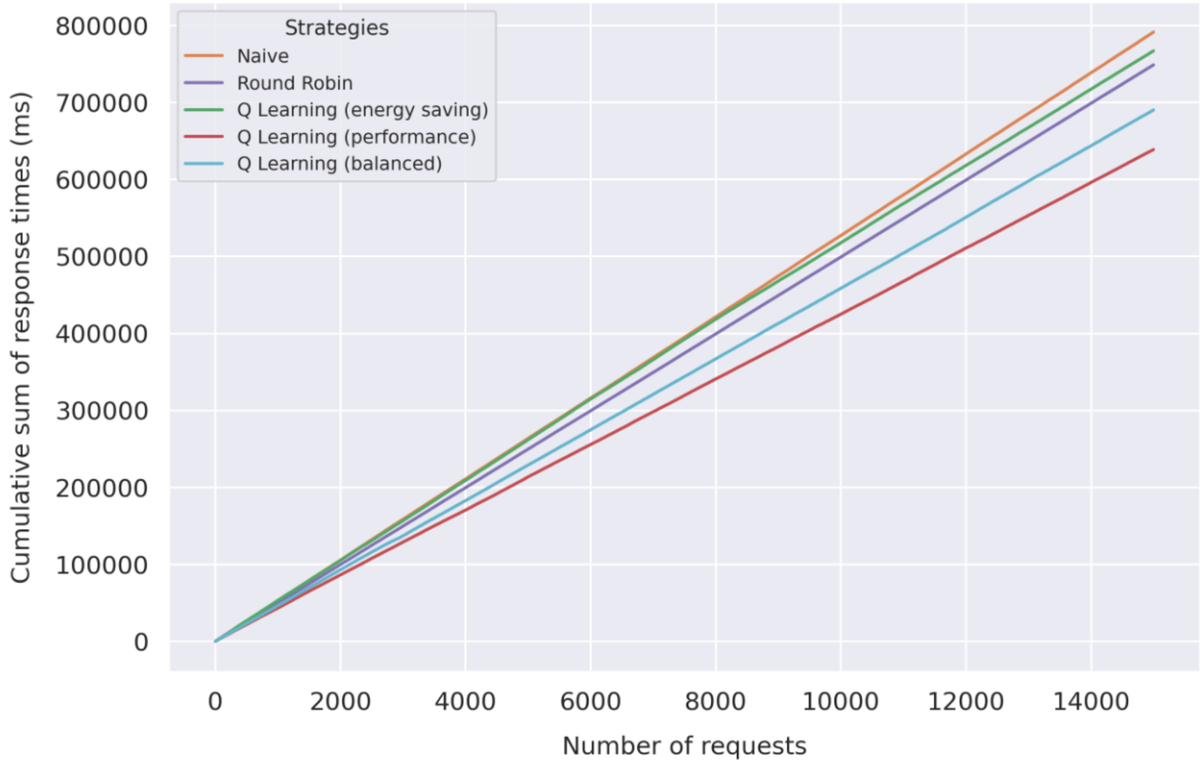




Response Time Effectiveness

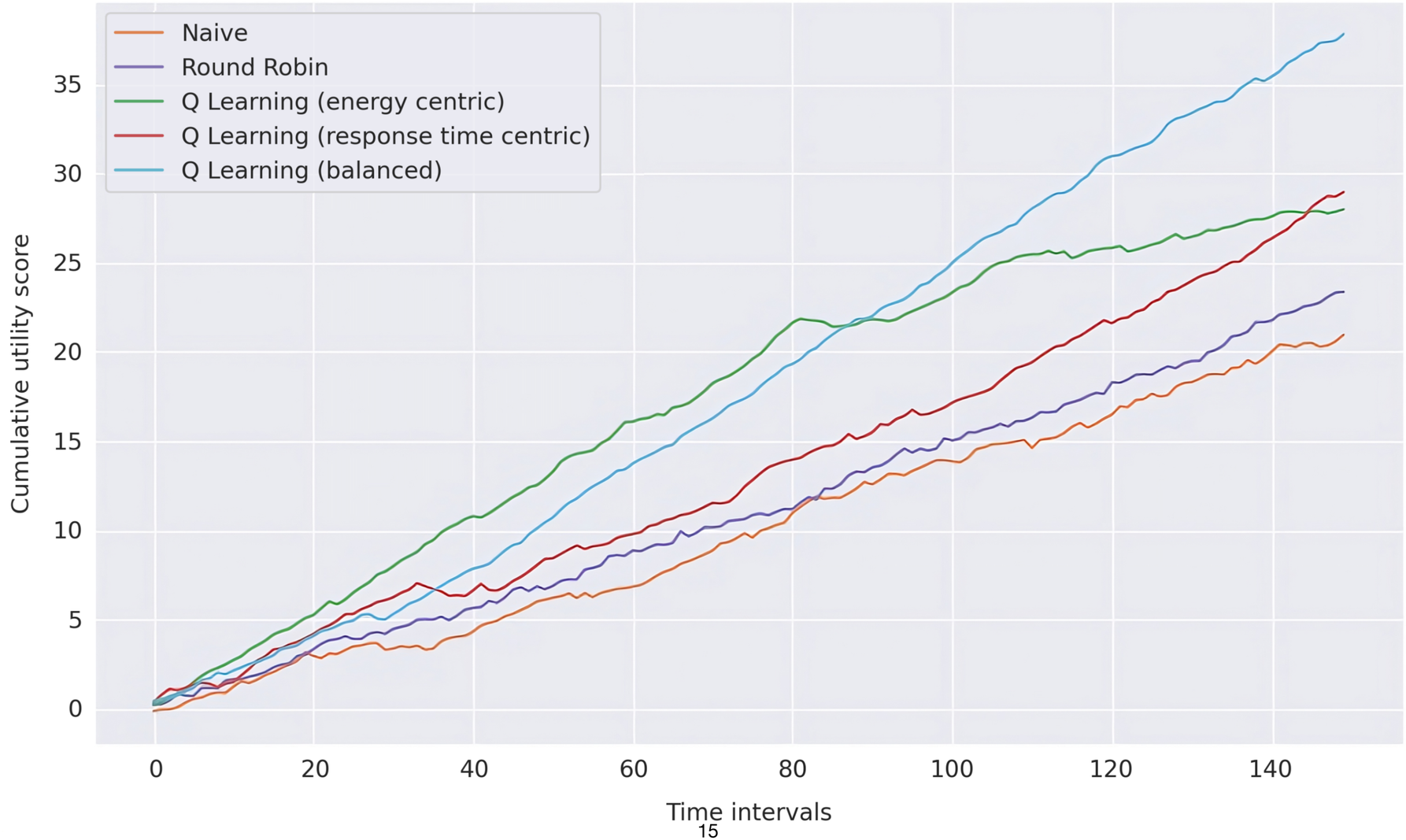
Comparison of response time while using each approach





Overall effectiveness

Cumulative plot of Utility over time



Conclusions and Future Work 

- Self-adaptation powered by ML can enable **sustainable** service discovery
- Development of **tool for ML-enabled service discovery** is under progress
- Expanding to large scale microservice systems
- **Distributed RL** working per service level may enhance scalability
- Potential of **Deep-Q and transfer learning** needs to be explored
- **ML requires data:** Need for test beds and simulators for microservice system
- Open to new collaborations as well!

SCAN ME

Thank you

More info: karthikvaidhyanathan.com
Email: karthik.vaidhyanathan@iiit.ac.in
Twitter: @karthi_ishere