GEdge Platform

GEdge(Griffin-Edge) Platform
- 초저지연 지능형 클라우드 엣지 SW 플랫폼 -

강화학습 기반 멀티 엣지 협업 정책 생성 기술

2023.12.07

GS-Link 프레임워크 코어개발자 (GS-LinkHQ)

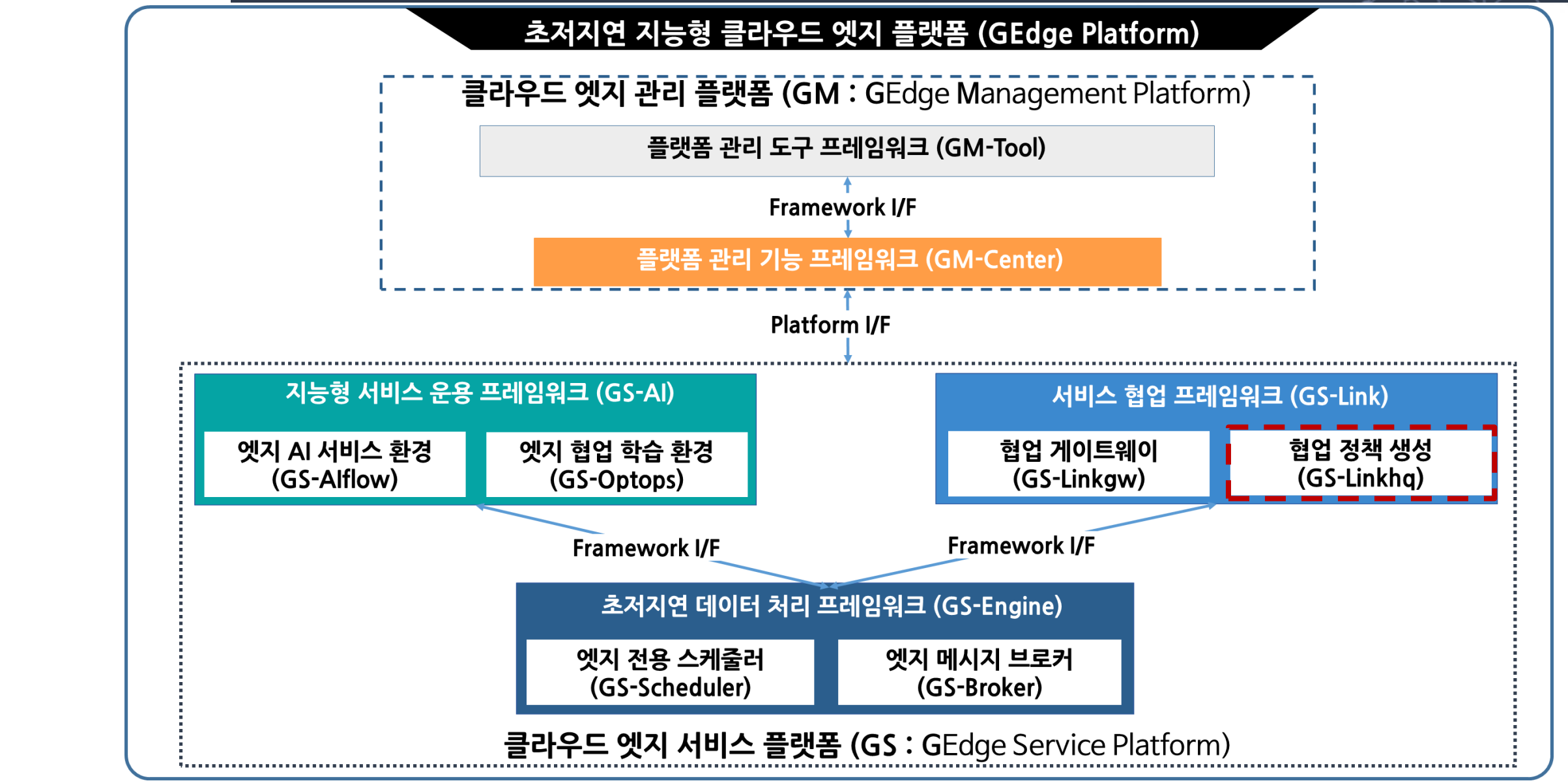
윤주상 (joosang.youn@gmail.com)

"GEdge Platform" 은 클라우드 중심의 엣지 컴퓨팅 플랫폼을 제공하기 위한
핵심 SW 기술 개발 커뮤니티 및 개발 결과물의 코드명입니다.

- New Leap Forward of
GEdge Platform Community 7th Conference (GEdge Platform v4.0 Release) -

Contents

GEdge 플랫폼 내 LinkHQ의 역할

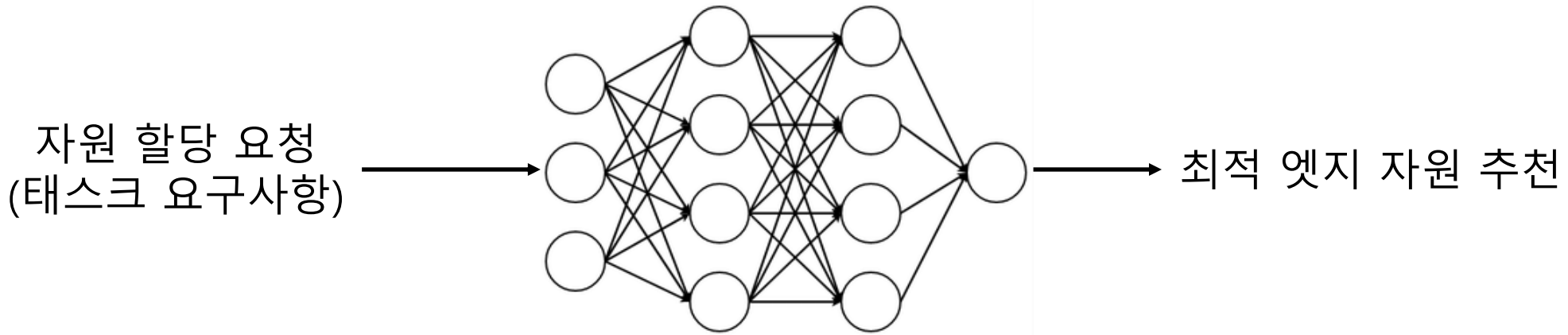


강화학습 기반 지능형 오프로딩 정책 생성 기술

1 엣지-클라우드 자원 할당 정책

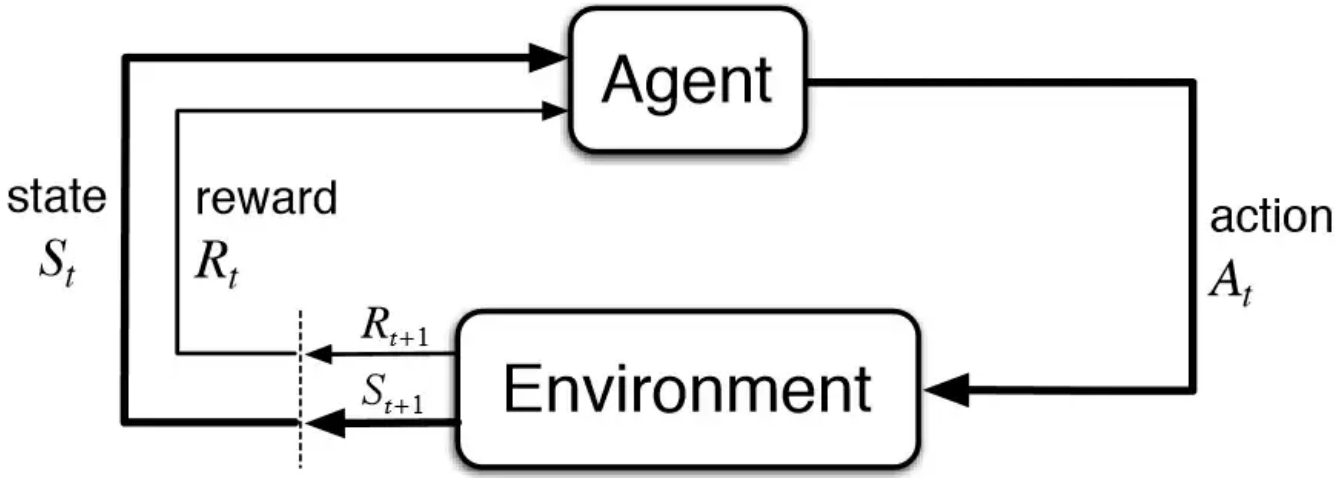
- **기존 자원 할당 정책**
  - Random, Least Load, Round-Robin
  - Rule Based

- **심층강화학습 기반 자원 할당 정책**

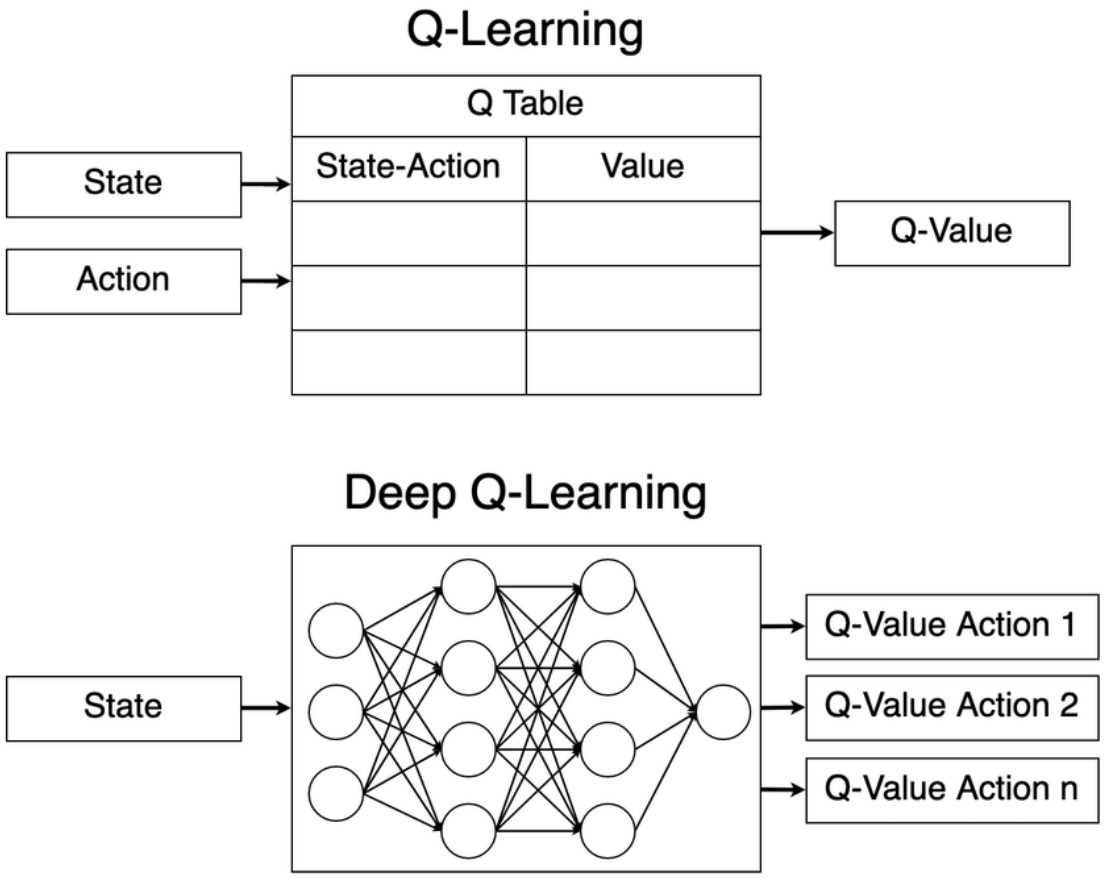


2 강화학습

- **강화학습**
  - 에이전트가 환경과 상호작용하며 해당 경험을 통해 특정 상태에서 **최적 행동을 학습**하는 기법

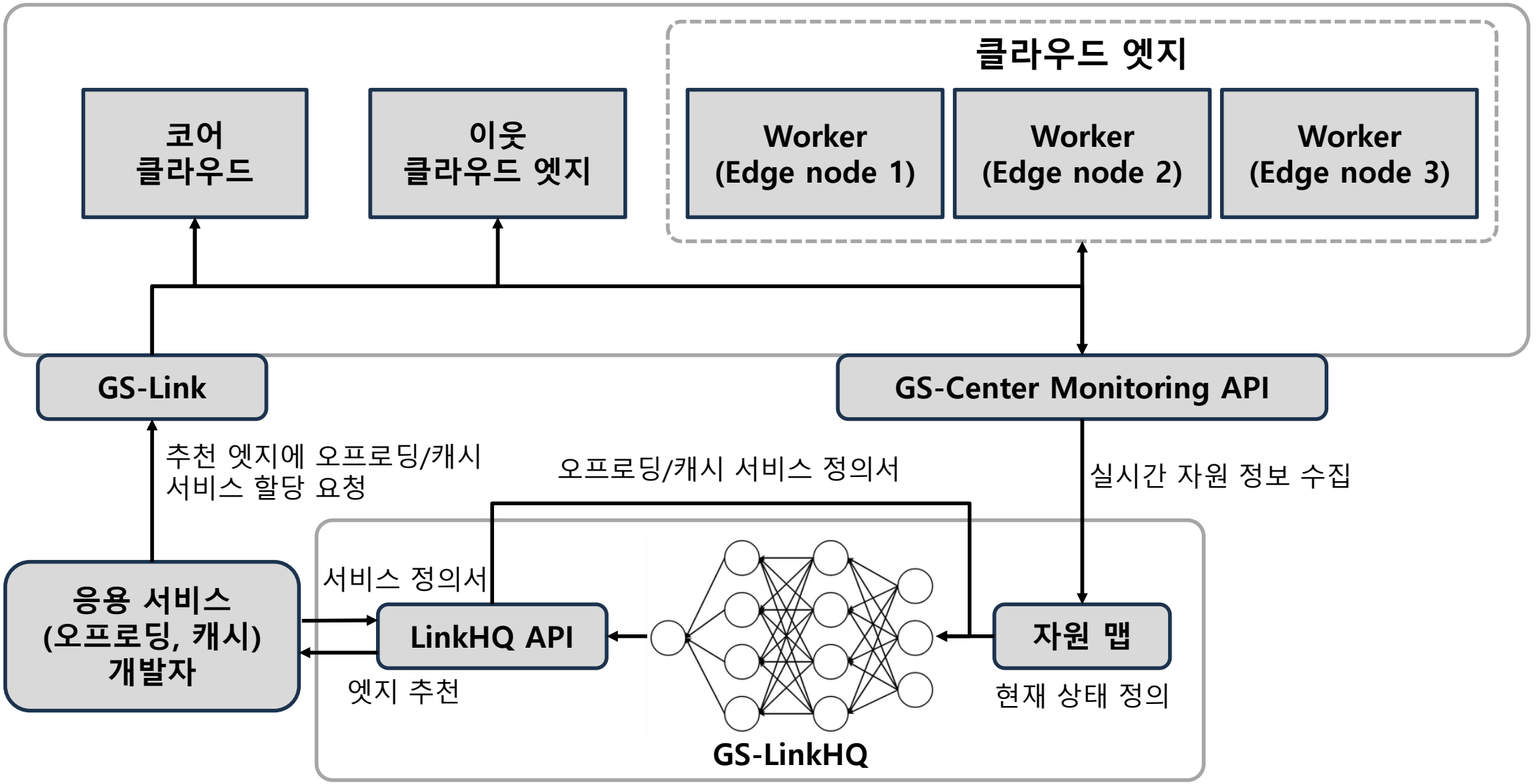


3 Deep Q-Learning (DQN)

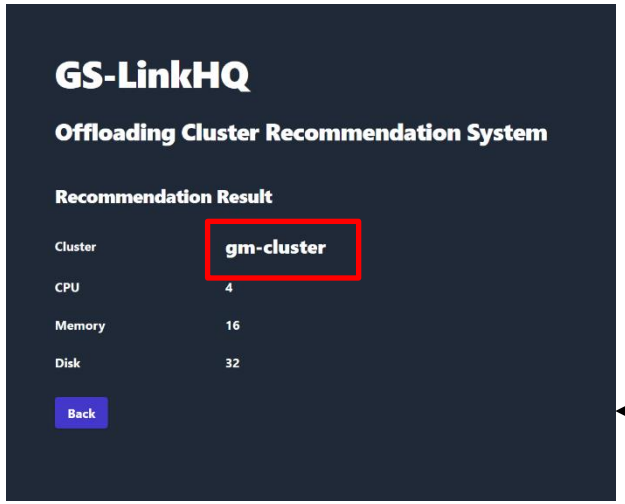


- **Q-Learning**
  - Lookup-table 기반 강화학습 기법
  - Q table에 에이전트의 경험과 가치를 저장
  - 예상 보상이 최대가 되는 행동을 반환하도록 학습
  - 환경이 복잡하면 효율이 떨어짐
- **Deep Q-Learning**
  - Q table 대신 신경망을 통해 Q-value를 근사
  - Replay memory를 통해 경험의 효율을 높임
  - 정책을 학습하는 네트워크와 액션을 추론하는 네트워크를 분리하여 학습 안정성 향상
  - 복잡한 환경에서도 안정적인 정책 학습 가능

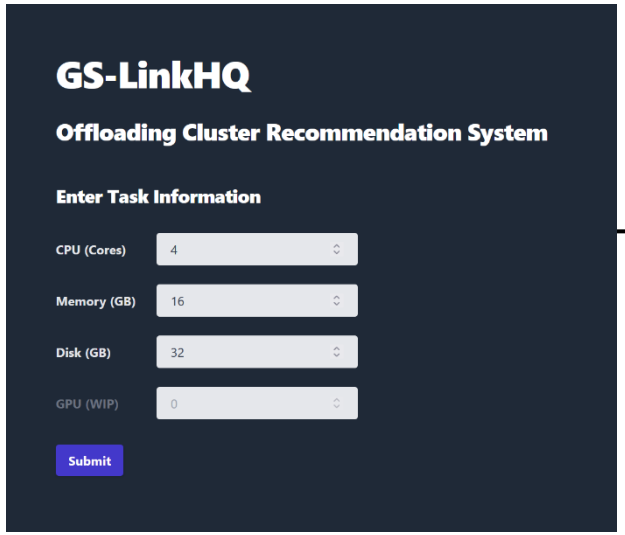
4 시스템 모델링


클라우드 엣지
코어
클라우드
이웃
클라우드 엣지
Worker
(Edge node 1)
Worker
(Edge node 2)
Worker
(Edge node 3)
GS-Link
GS-Center Monitoring API
추천 엣지에 오프로딩/캐시
서비스 할당 요청
오프로딩/캐시 서비스 정의서
실시간 자원 정보 수집
서비스 정의서
응용 서비스
(오프로딩, 캐시)
개발자
LinkHQ API
엣지 추천
자원 맵
현재 상태 정의
GS-LinkHQ

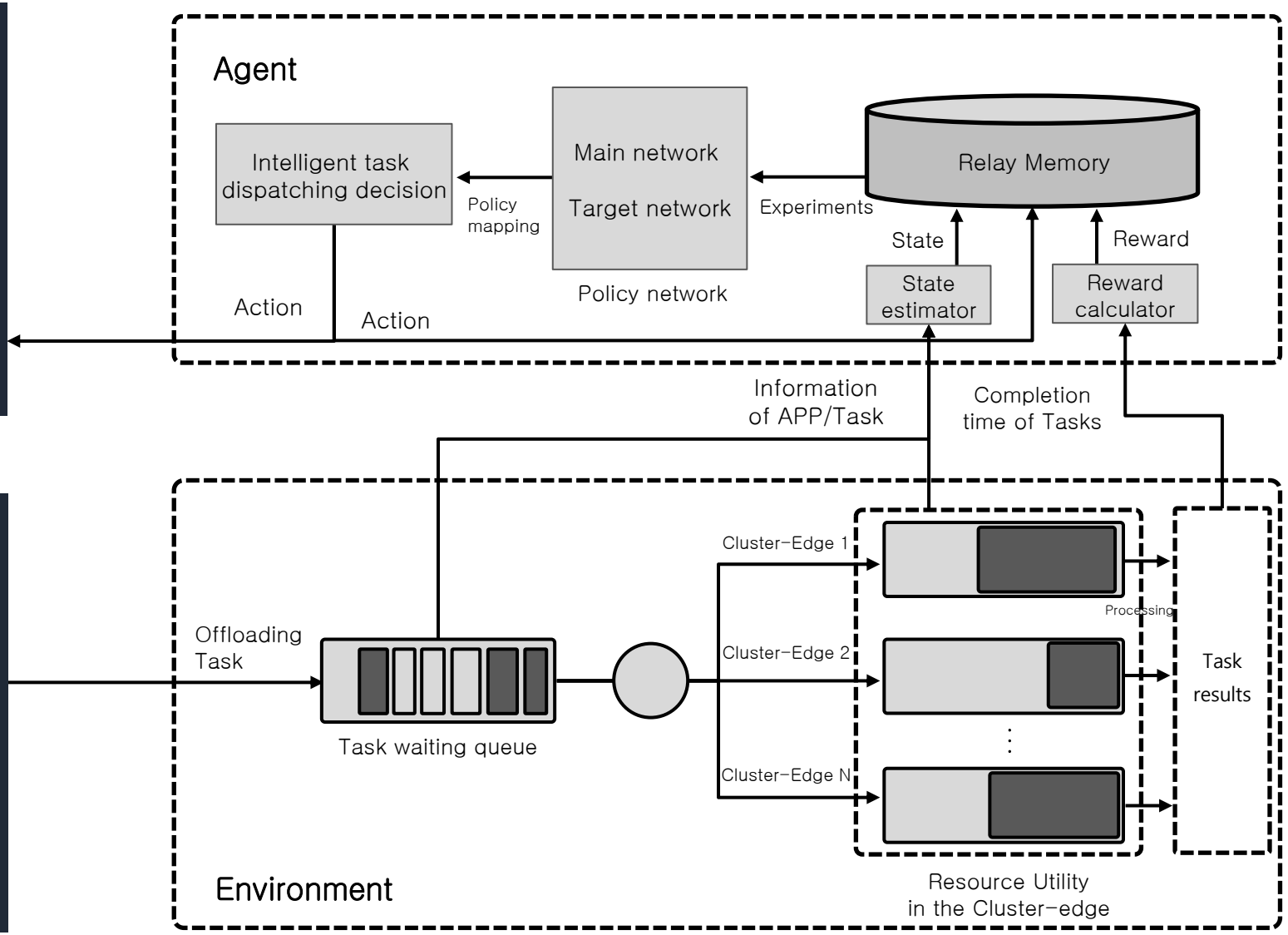
4 시스템 모델링



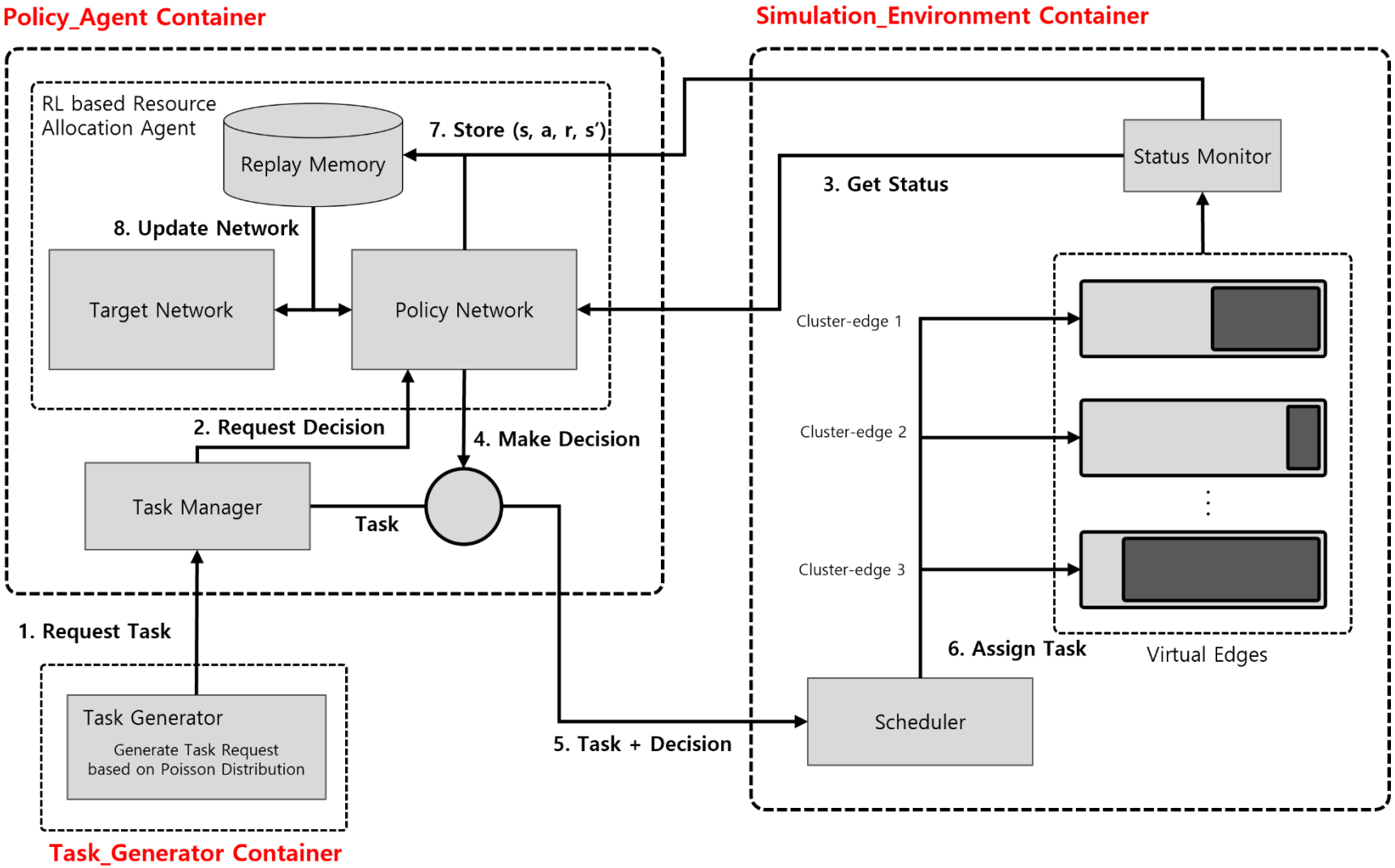
LinkHQ Result Page



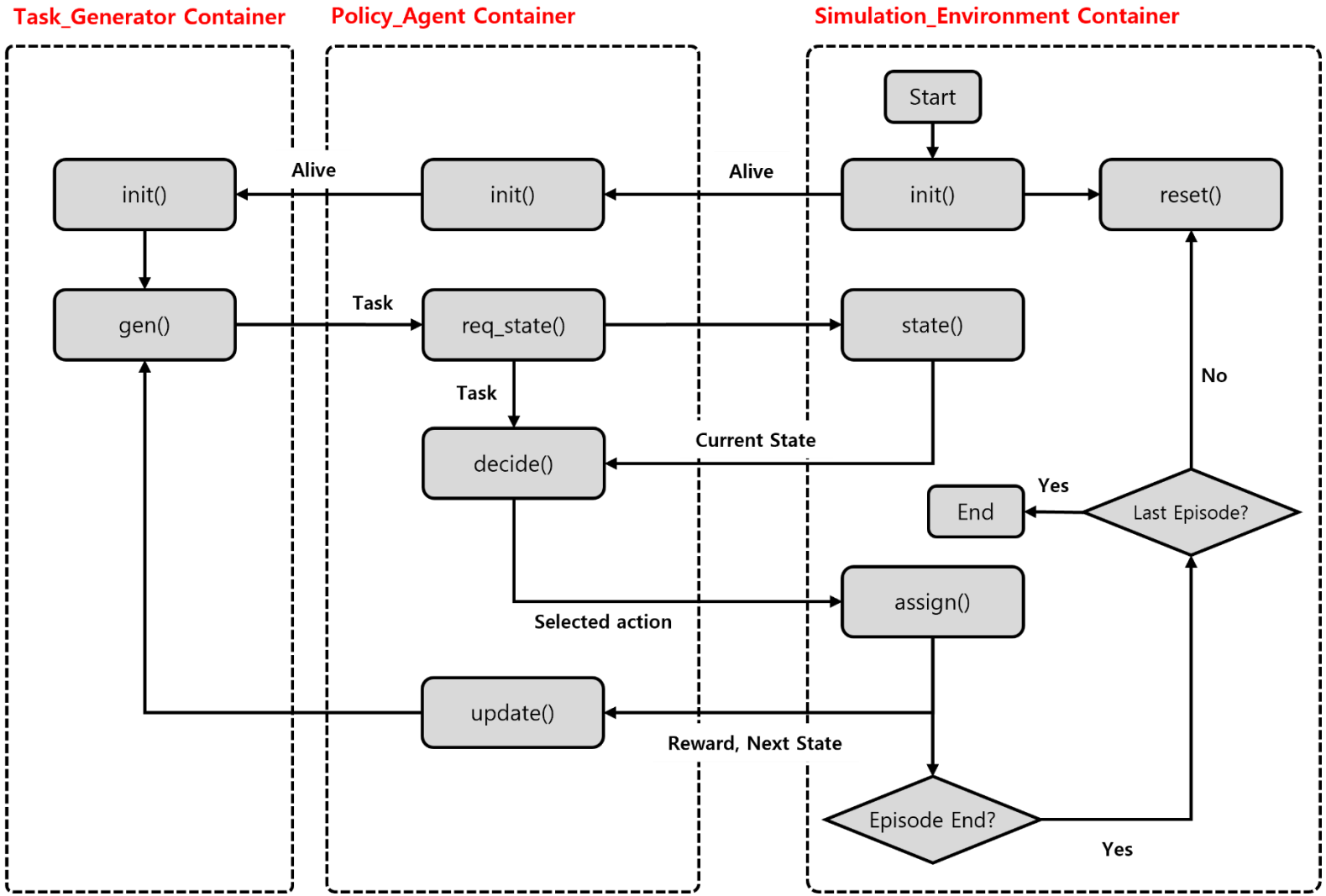
LinkHQ Request Page



5 학습 시뮬레이터 구축


Policy_Agent Container
Simulation_Environment Container
RL based Resource Allocation Agent
Replay Memory
7. Store (s, a, r, s')
Status Monitor
3. Get Status
8. Update Network
Target Network
Policy Network
Cluster-edge 1
Cluster-edge 2
Cluster-edge 3
2. Request Decision
4. Make Decision
Task Manager
Task
1. Request Task
6. Assign Task
Virtual Edges
Task Generator
Generate Task Request based on Poisson Distribution
5. Task + Decision
Scheduler
Task_Generator Container

6 정책 학습 과정


Task_Generator Container
Policy_Agent Container
Simulation_Environment Container
Start
init()
Alive
init()
Alive
init()
reset()
gen()
Task
req_state()
state()
Task
No
Current State
decide()
Yes
End
Last Episode?
assign()
Selected action
update()
Reward, Next State
Episode End?
Yes

II 지능형 오프로딩 정책 생성

1 오프로딩 시스템 수식화

Notation	Definition
n	Number of edge servers
e_k	k-th edge server
e_k^{CPU}	Available CPU resource of e_k
e_k^{MEM}	Available memory resource of e_k
e_k^{GPU}	Available GPU resource of e_k
t_i	i-th requested task
t_i^e	Index of edge requested by Task t_i
t_i^{index}	Flag indicating whether t_i is requested to e_k (one hot encoded)
t_i^{CPU}	Required CPU of t_i
t_i^{MEM}	Required memory of t_i
t_i^{GPU}	Required GPU of t_i
t_i^T	Flag indicating whether t_i is time sensitive task
S_i	State set of system when task i requested
A_i	Decided action set of requested task i
α_i^{index}	Index of edge assigned Task t_i
α_i^k	Flag indicating whether t_i is assigned to e_k (one hot encoded)
R_i	Reward of A_i

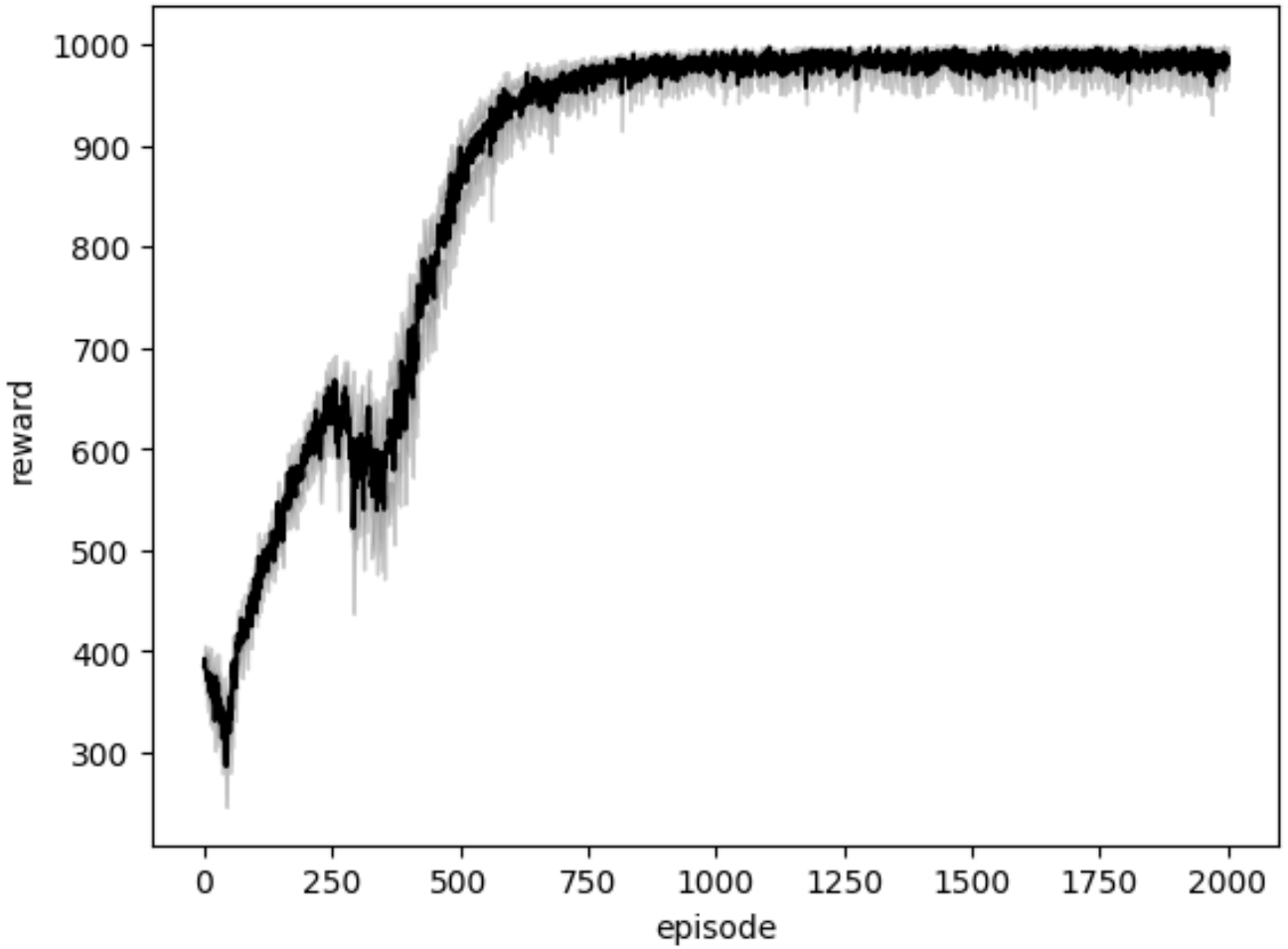
- State (S_i) : 태스크 정보 + 엣지 상태
- Action (A_i) : 오프로딩 결정 엣지
- Reward (R_i) : Action에 대한 평가 점수

$$S_i = \{t_i^0, t_i^1, \ldots, t_i^n, t_i^{CPU}, t_i^{MEM}, t_i^{GPU}, t_i^T, e_0^{CPU}, e_0^{MEM}, e_0^{GPU}, \ldots, e_n^{CPU}, e_n^{MEM}, e_n^{GPU}\}$$

$$A_i = \{\alpha_i^0, \alpha_i^1, \alpha_i^2, \ldots, \alpha_i^n\}$$

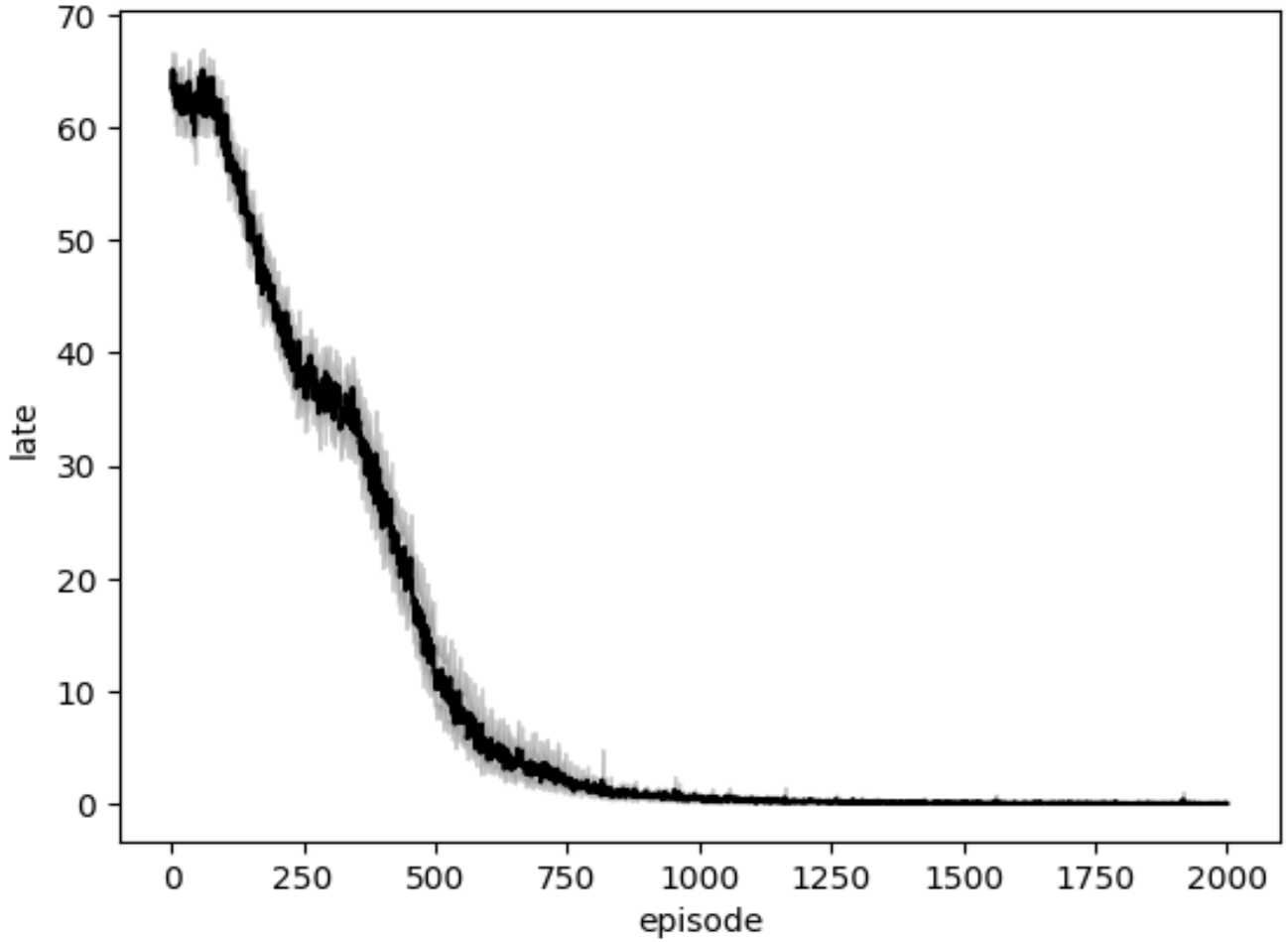
$$R_i = \begin{cases} 10 & \text{allocated and } t_i^T = 0 \\ 10 & \text{allocated and } t_i^T = 1 \text{ and } t_i^{index} = \alpha_i^{index} \\ 1 & \text{allocated and } t_i^T = 1 \text{ and } t_i^{index} \neq \alpha_i^{index} \\ -10 & \text{rejected} \end{cases}$$

2 학습 결과



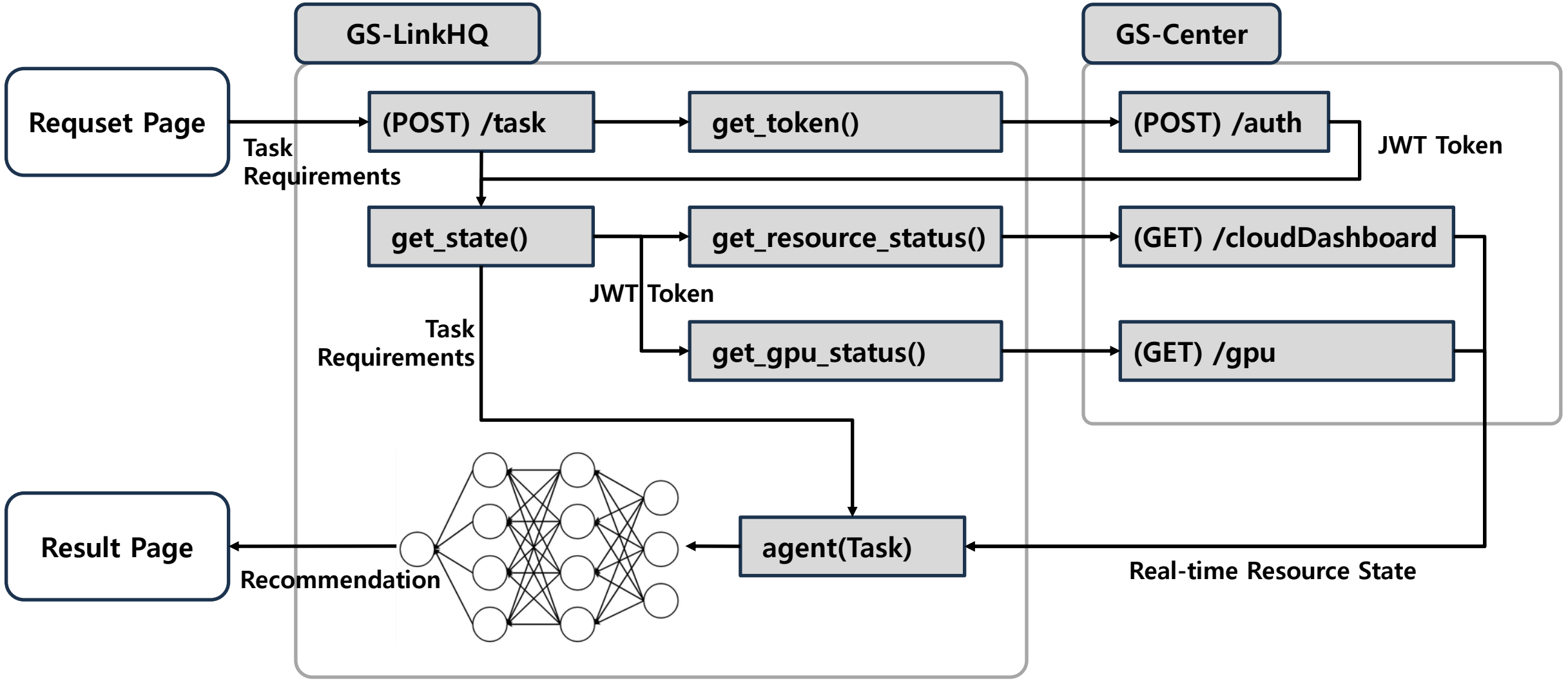
- **Reward**
  - 각 에피소드에서 reward 총 합
  - 800 에피소드 이후로 시뮬레이터 최대 reward =1000 으로 수렴

3 학습 결과

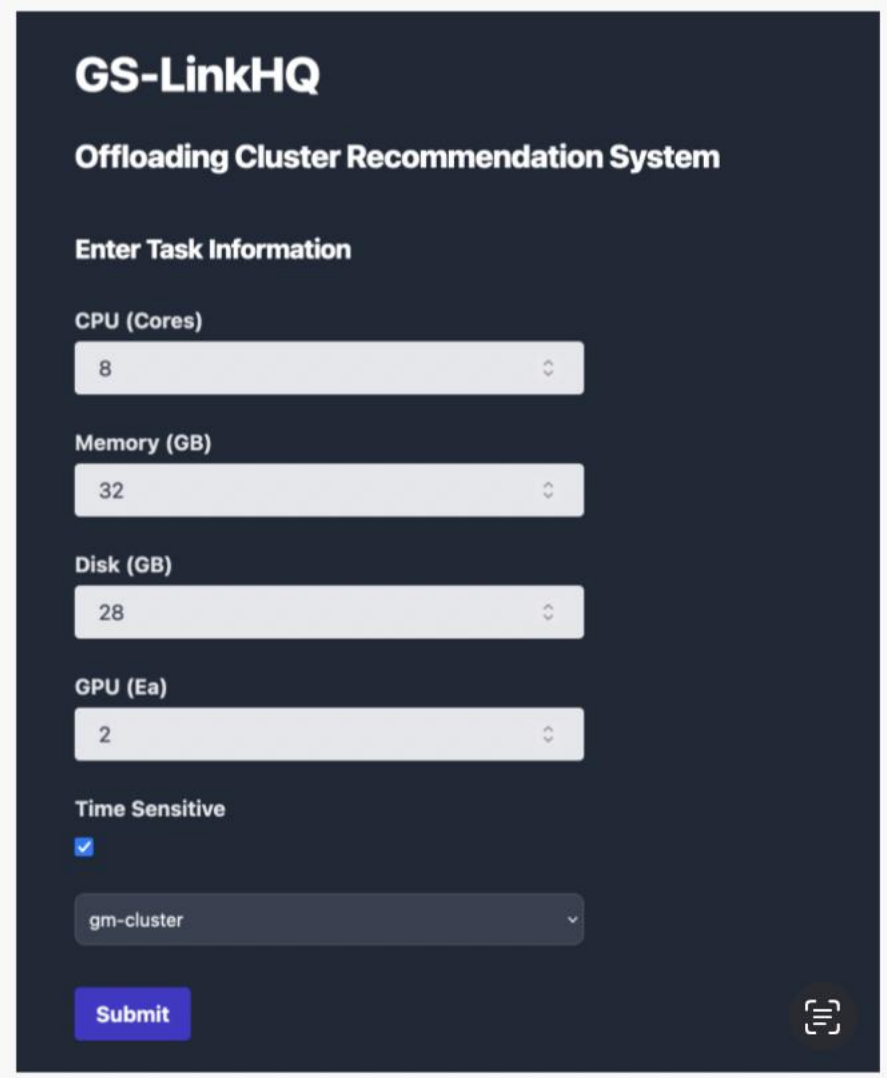


- Late Count
  - 각 에피소드에서 시간민감형 태스크가 멀리 떨어진 엣지에 할당된 횟수
  - 800 에피소드 이후로 0건으로 수렴

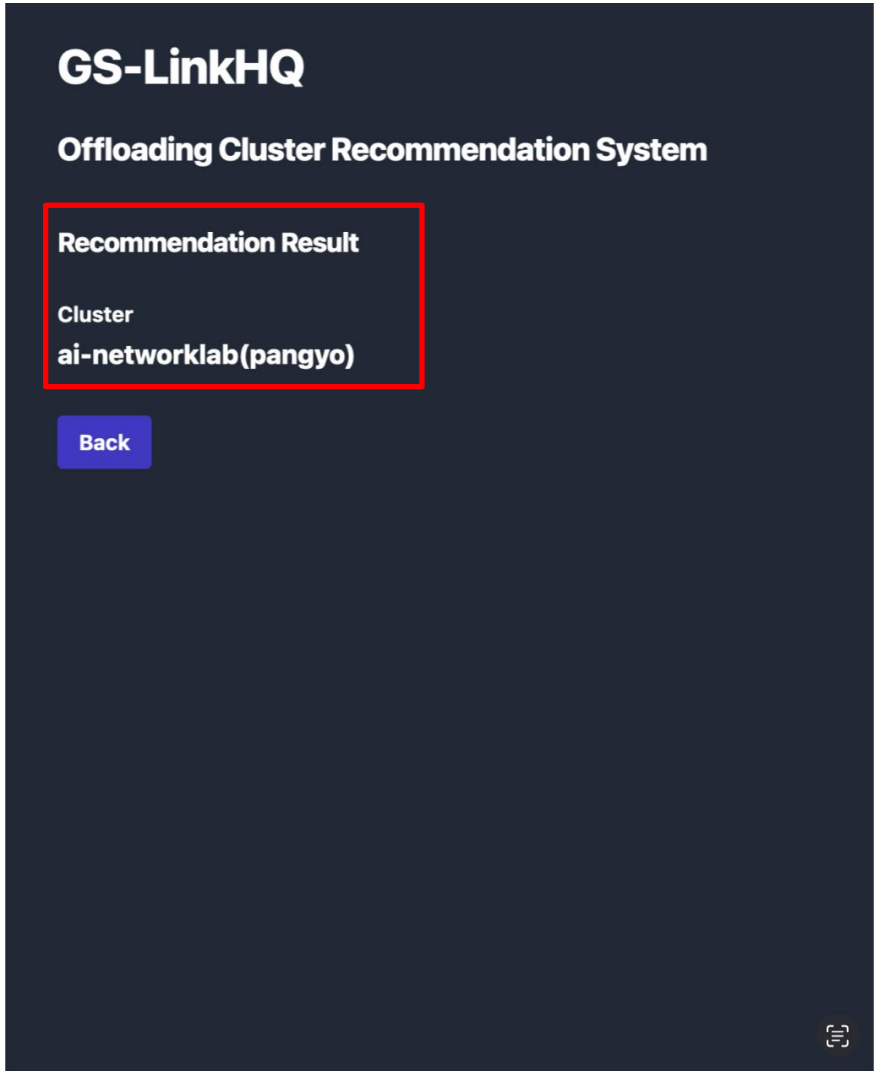
4 오프로딩 엣지 자원 정책 형상



5 오프로딩 엣지 추천 서비스



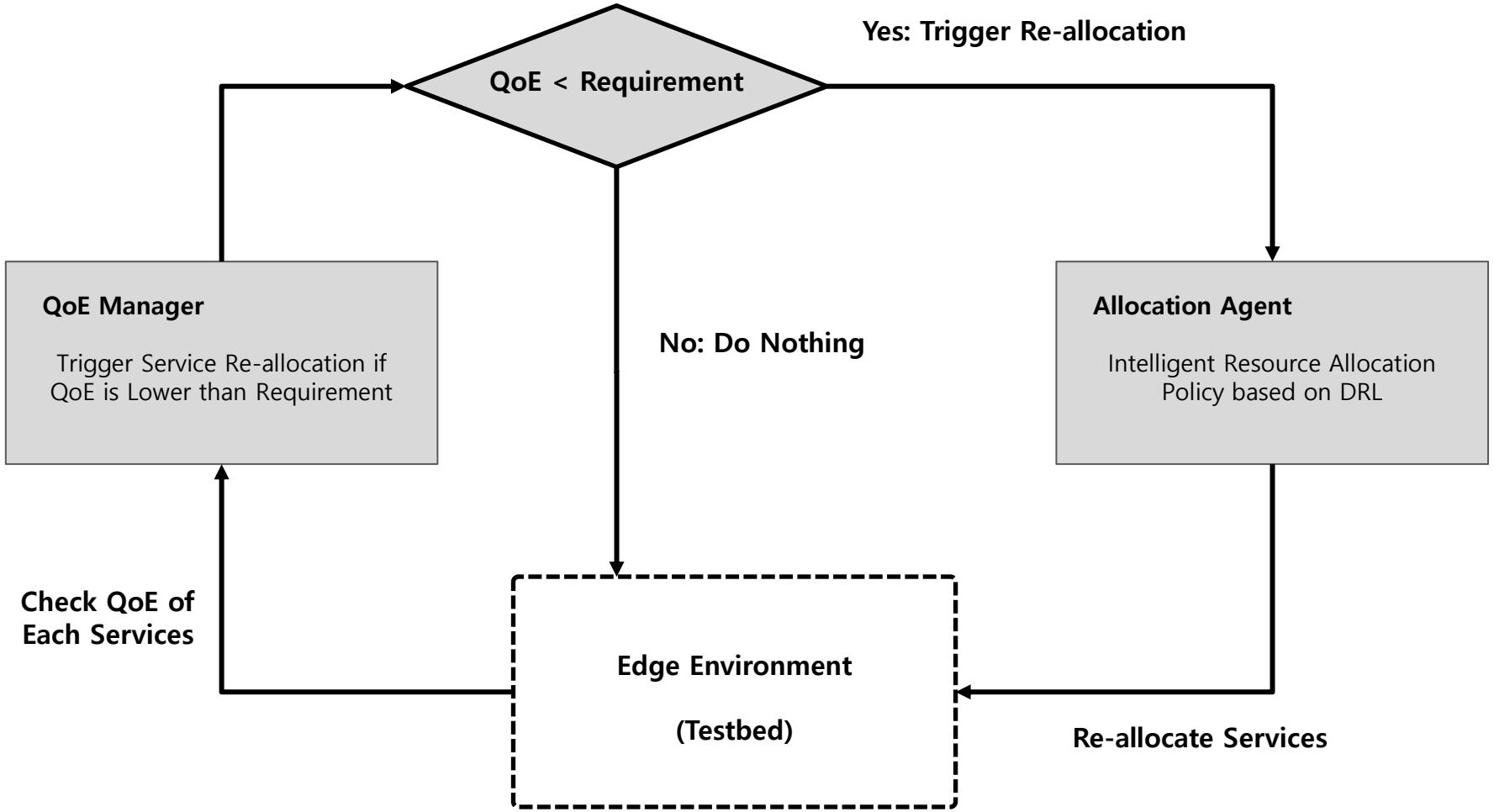
Recommendation Request Page



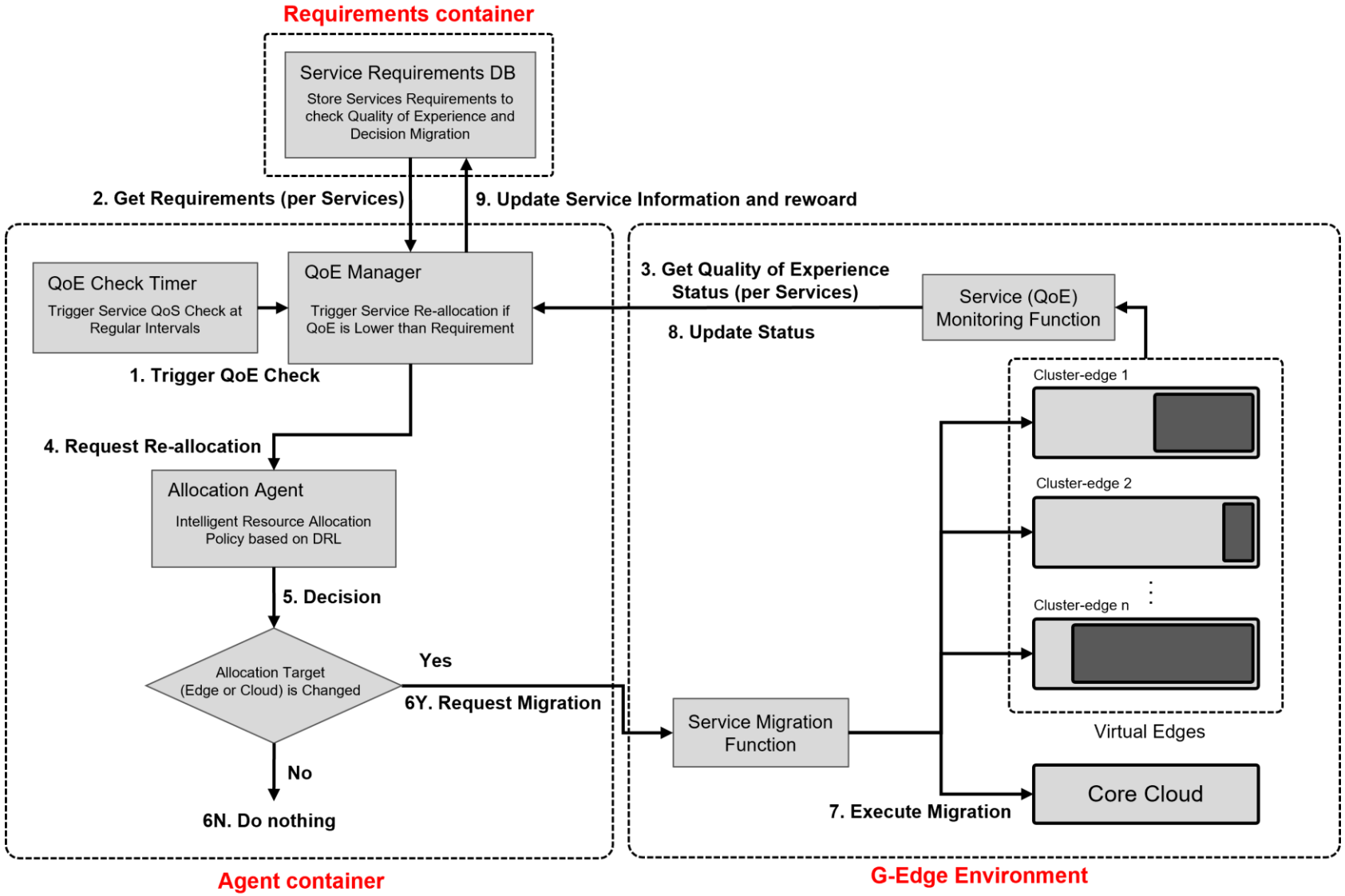
Recommendation Result Page

지능형 서비스 이동 정책

1 서비스 이동 정책


QoE < Requirement
Yes: Trigger Re-allocation
No: Do Nothing
QoE Manager
Trigger Service Re-allocation if QoE is Lower than Requirement
Allocation Agent
Intelligent Resource Allocation Policy based on DRL
Check QoE of Each Services
Edge Environment
(Testbed)
Re-allocate Services

2 서비스 이동 정책 생성기 작동 구조



3 서비스 이동 매니저

GEdge Platform

```
...
class QoEManager:
    def __init__(self, env, db, agent):
        self.db = db
        self.env = env
        self.agent = agent

    def qoe_check(self):
        requirements = self.db.get_requirements()
        qoe_status = self.env.get_qoe()

        for service in qoe_status:
            if service['requirement'] > requirements[service['task_id']]:

    def request_allocation(self, service):
        target = self.agent.allocation(service['task_id'])

        if target != service['target']:
            res = self.env.migration(service['task_id'], target)

            if res:
                self.update_db(service, target)
                self.agent.update_reward()

    def update_db(self, service, target):
        self.db.hest(service['task_id'], 'target', target)
...
```

서비스 요구사항 만족여부 확인

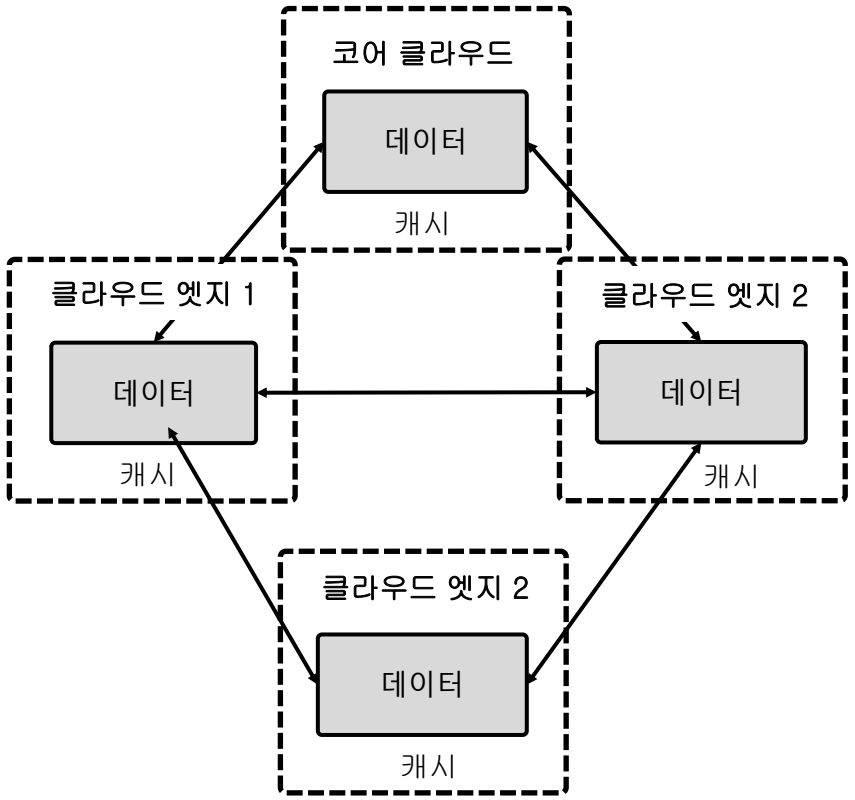
서비스 자원 재할당 요청

서비스 DB 및 리워드 업데이트

IV 지능형 캐싱 정책 생성

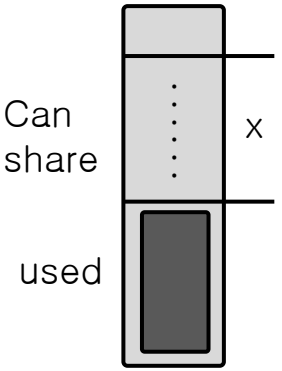
1 캐시 정책 시나리오

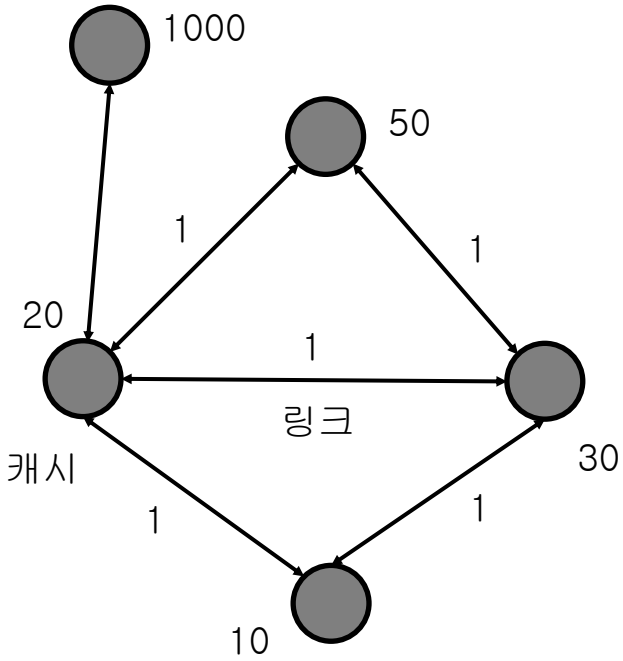
- 협업 캐시 물리적 네트워크



물리적 캐시 네트워크 모델

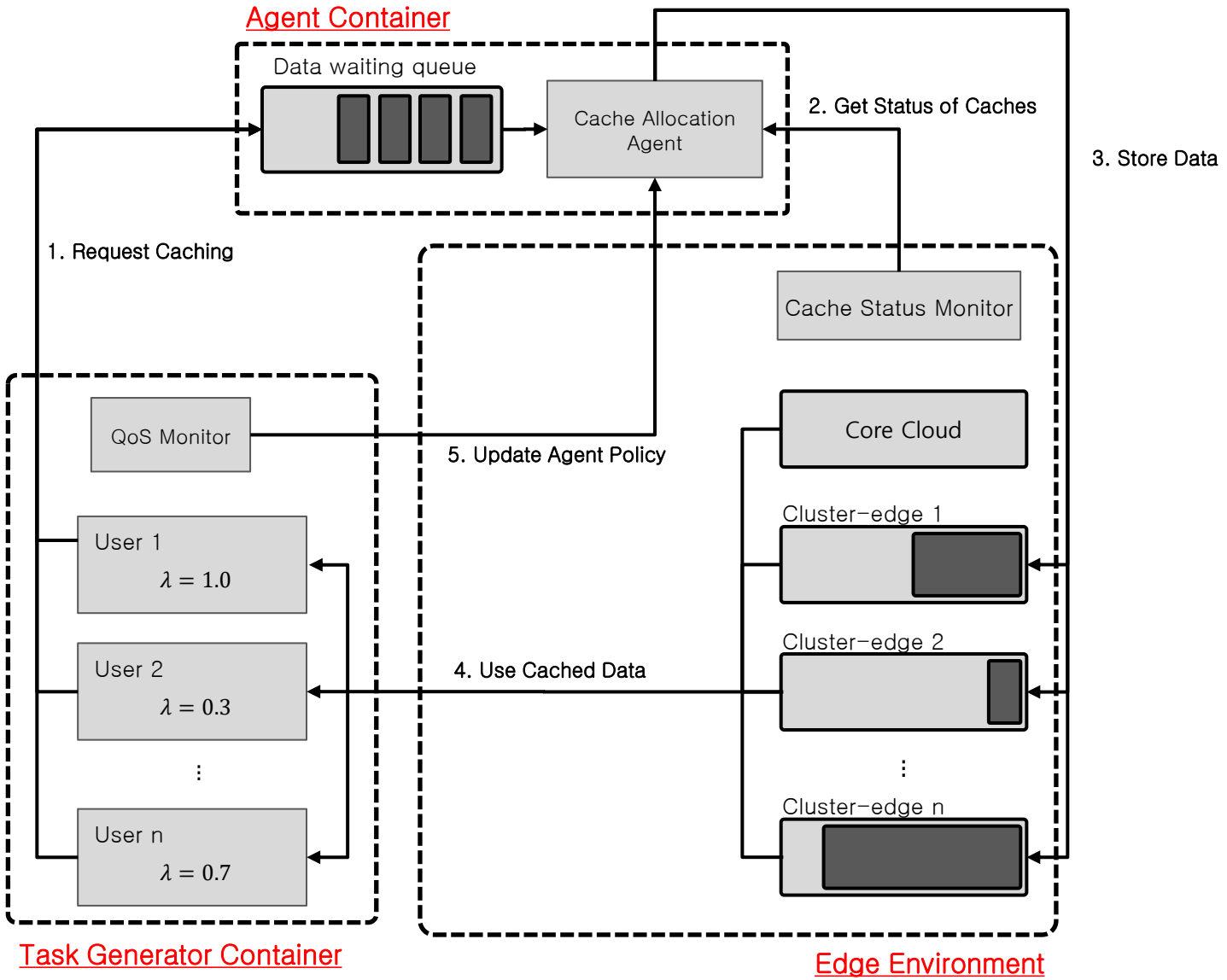
캐시 자원
정의 (x-level)





그래프 기반 캐시 네트워크

2 분산 캐싱 정책 생성기 구현 계획



3 분산 캐싱 정책 생성기 구현 계획

GEdge Platform

Cache Assign

Endpoint

```
http://<agent>/cache
```

Method

`post`

Contents

- `cache`
  - `size`: Size of cache (MB)
  - `deadline`: Deadline of cache(ms)
  - `tts`: Time to survive of service (min)
  - `rts`: Max request number of service

Request

Request agent to assign a cache

Body Format

```
{
    "cache": {
        "size": <size>,
        "deadline": <deadline>,
        "tts": <tts>,
        "rts": <rts>
    }
}
```

Sample

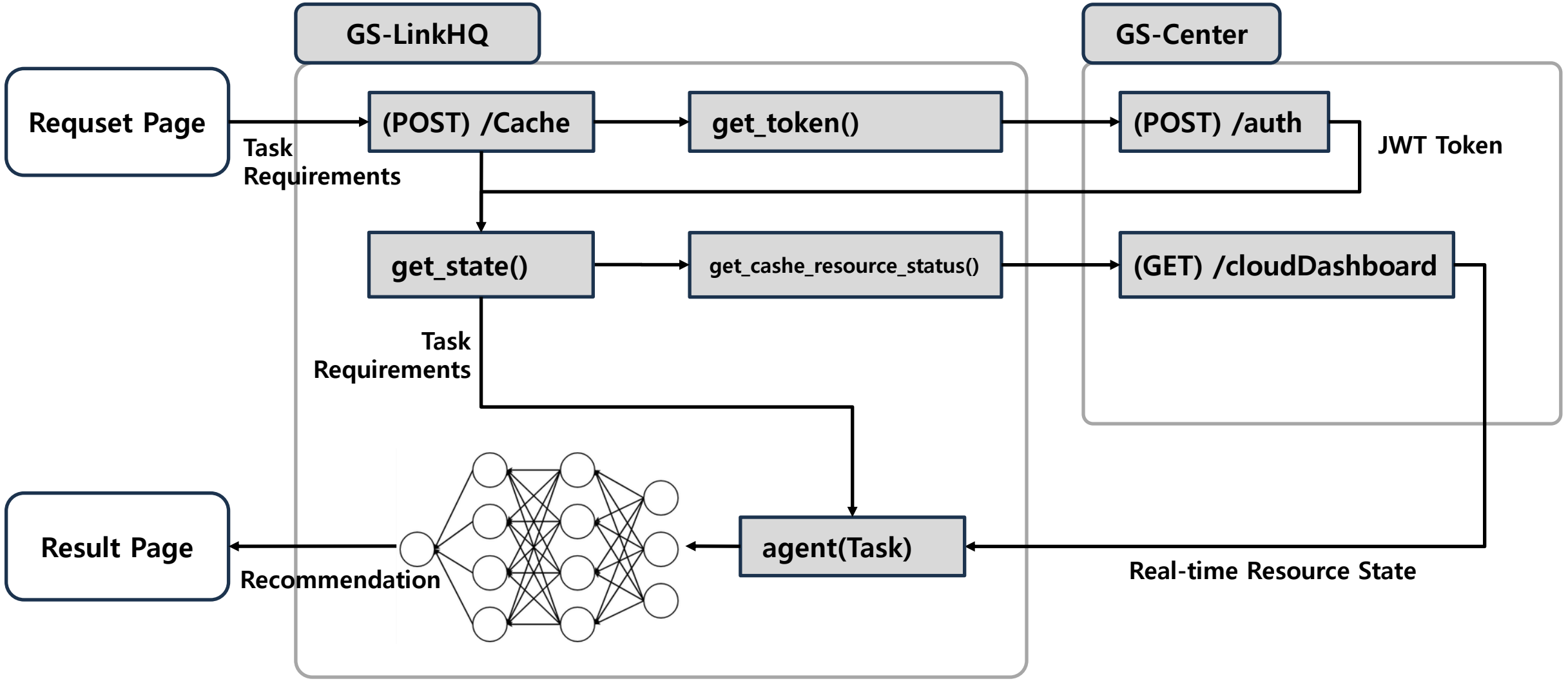
```
POST /cache HTTP/1.1
Host: <agent>
Content-Type: application/json
{
    "cache": {
        "size": 5,
        "deadline": 0.1,
        "tts": 1440,
        "rts": 100
    }
}
```

Response

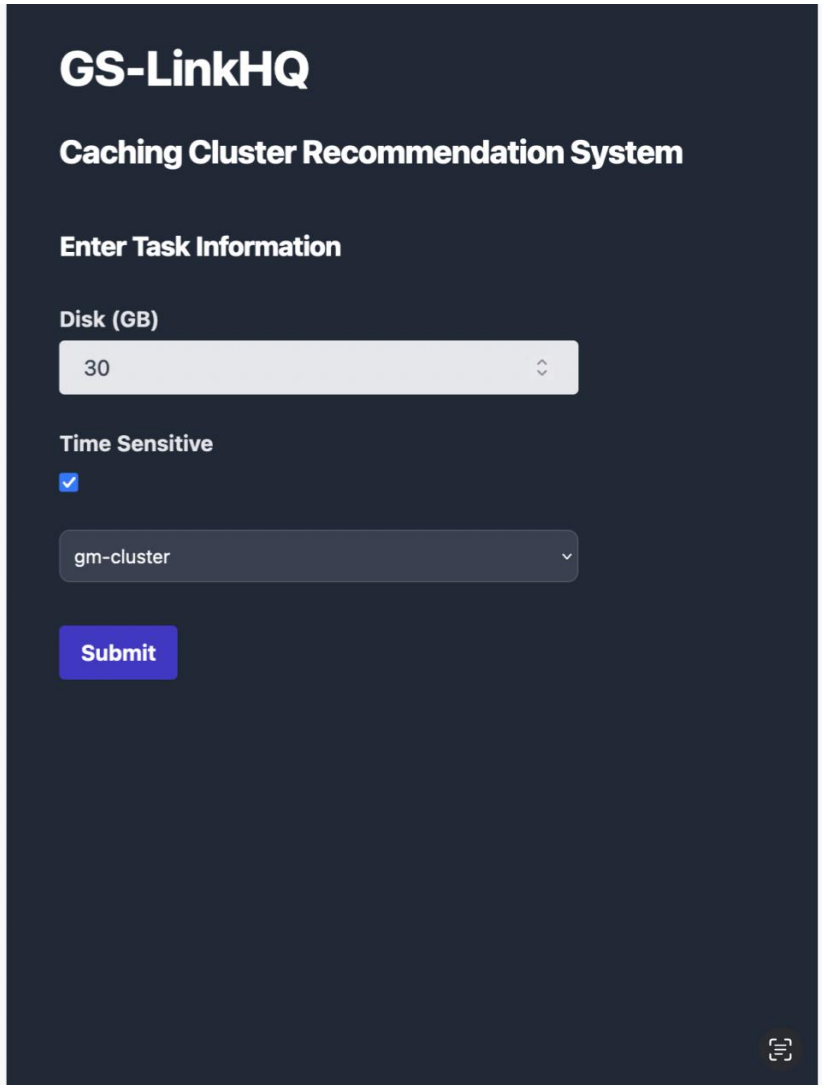
201

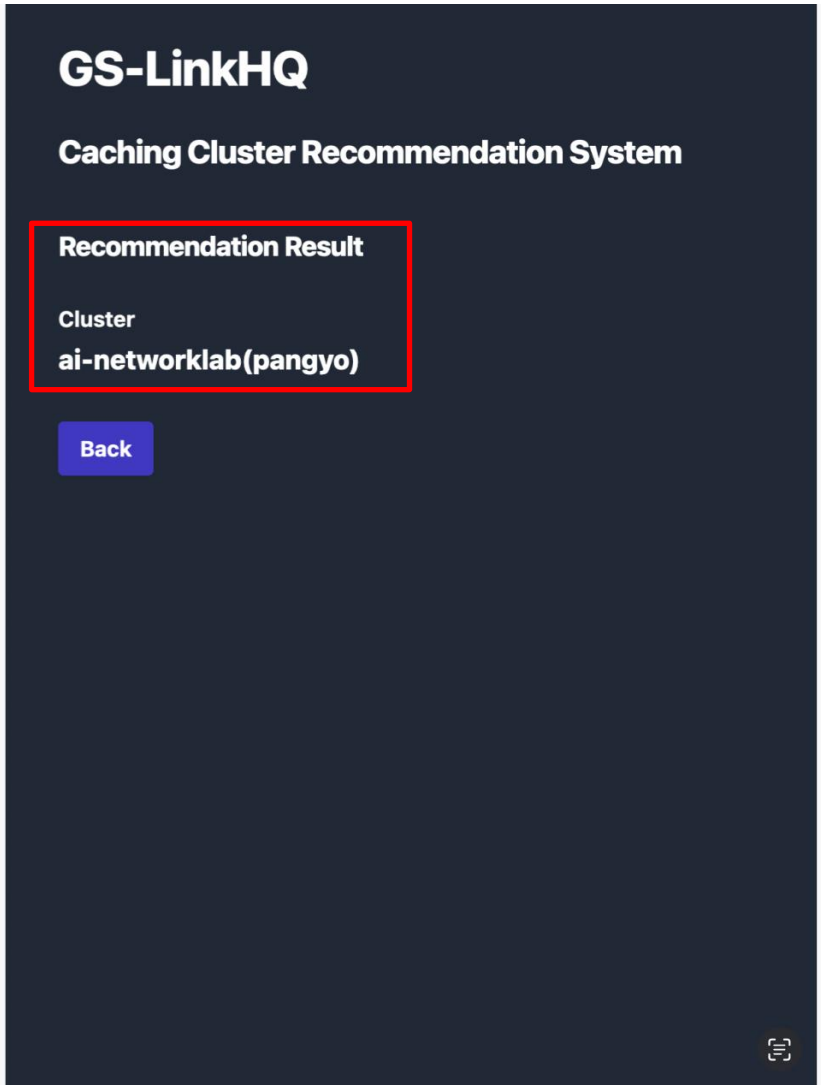
```
HTTP/1.1 201 Created
Location: /cache/<cache-id>
Content-Type: application/json
{
    "id": <cache-id>,
    "message": "Successfully created"
}
```

4 엣지 캐싱 자원 정책 서비스 형상



5 분산 캐시 정책 생성기 서비스





감사합니다.

http://gedge-platform.github.io

GS-Link 프레임워크 코어개발자(GS-LinkHQ)
윤주상 (joosang.youn@gmail.com)