딥러닝을 활용한 클라우드 엣지 수직/수평 협업

2020.12.10

GEdge Platform 코어 개발자
윤주상(jsyoun@deu.ac.kr)

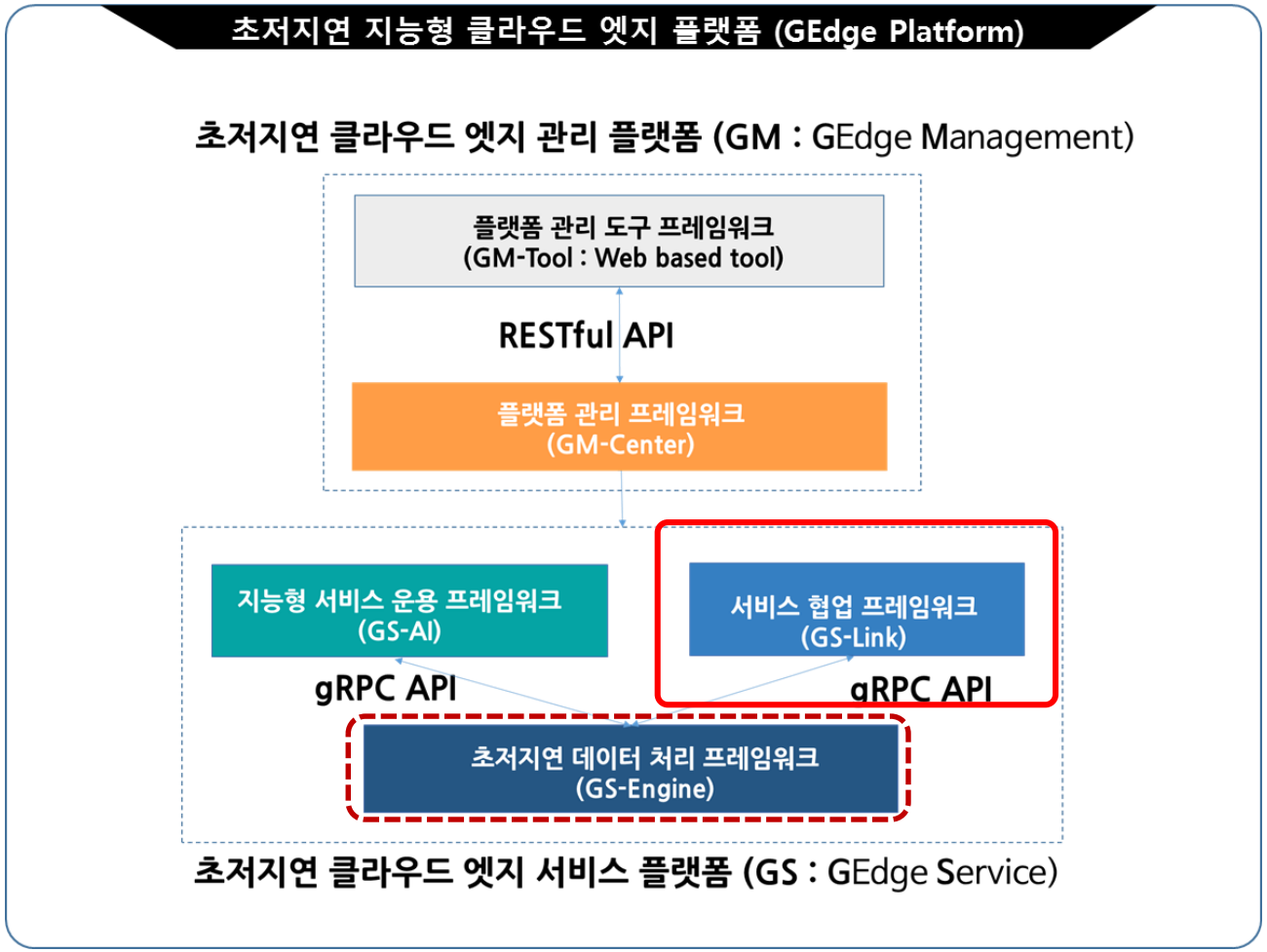
"The First talk of Edge Computing with Clouds"

- GEdge Platform 커뮤니티 멤버들의 첫번째 이야기 -

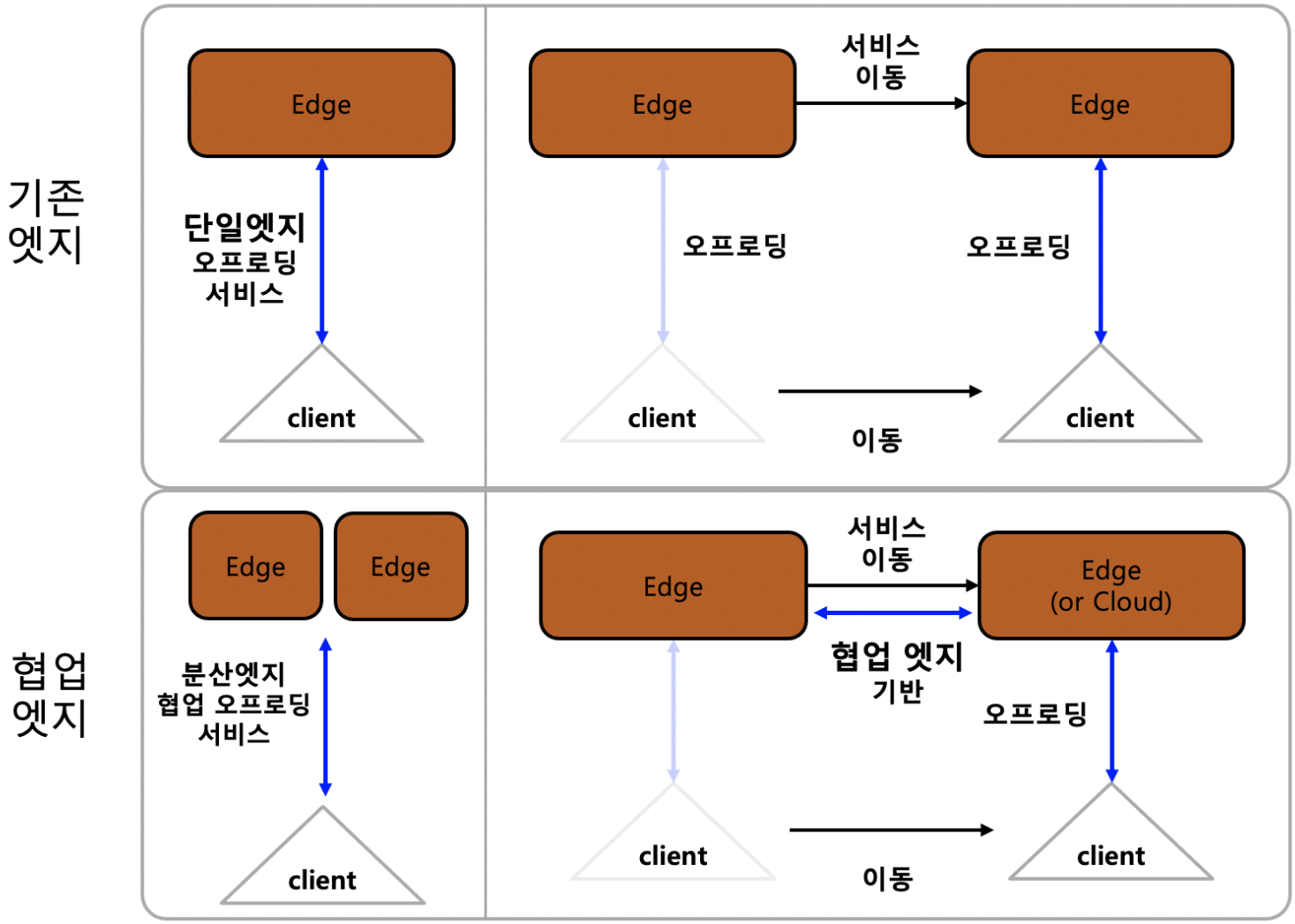
GEdge Platform Community 1st Conference

Contents

클라우드 엣지 수직/수평 협업 모델



1 기존 엣지 모델 vs. 협업 엣지 모델



2 GEdge 플랫폼 내 서비스 정의

- 일반적인 엣지 오프로딩 서비스
  - 디바이스가 가진 응용을 엣지에서 대신 실행하고 그 결과를 리턴 받는 서비스
  - 디바이스는 오프로딩 요청 시 실행 코드, 실행 시 필요한 데이터 등을 첨부하여 서비스를 요청

- 제안된 GEdge 오프로딩 서비스
  - 메인 응용 서비스 : 엣지 기반 인공지능 서비스 (추론 서비스)
  - GE 플랫폼에서는 3가지 응용 서비스 분류 모델을 지원
    - 오프로딩 요청 시 응용 별 별도의 오프로딩 서비스 제공
      - 3가지 협업 오프로딩 제공
    - 차별화된 서비스 모델 : 협업 오프로딩 (GEdge 오프로딩)
      - 엣지 간, 엣지-클라우드 간 컴퓨팅 자원을 공유하는 협업 오프로딩 서비스. 여기서 협업 오프로딩 서비스는 엣지 간 수평적 오프로딩 서비스, 엣지-클라우드 간 수직적 오프로딩 서비스로 정의

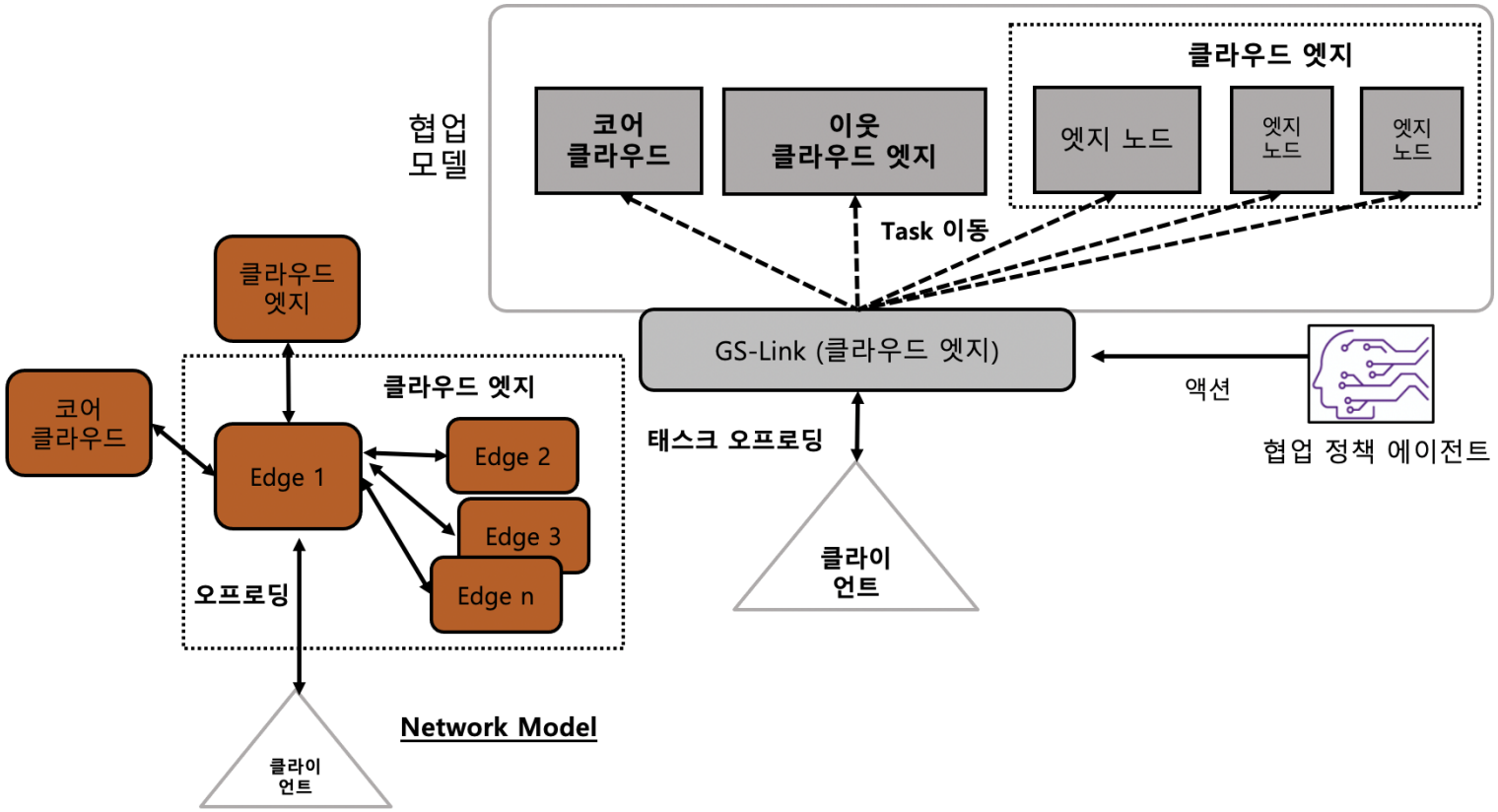
2 협업 오프로딩 in GEdge 플랫폼

- 기본 협업 오프로딩 (Collaborative Offloading Service(COS)-1)
  - Type 1 응용을 위한 협업 오프로딩 서비스
  - 일반적인 오프로딩 서비스
  - 정책 기반 응용을 위한 단일 엣지 할당 방법 적용
- 독립 태스크 용 협업 오프로딩 (COS-2)
  - Type 2 응용을 위한 협업 오프로딩 서비스
  - 정책 기반 태스크 별 단일 엣지 할당 방법 적용
- 종속 태스크 용 협업 오프로딩 (COS-3)
  - Type 3 응용을 위한 협업 오프로딩 서비스
  - 정책 기반 멀티 태스크(응용)을 고려한 그룹 엣지 할당 방법 적용

- 모든 협업 오프로딩 서비스는 수평적 협업 서비스 제공

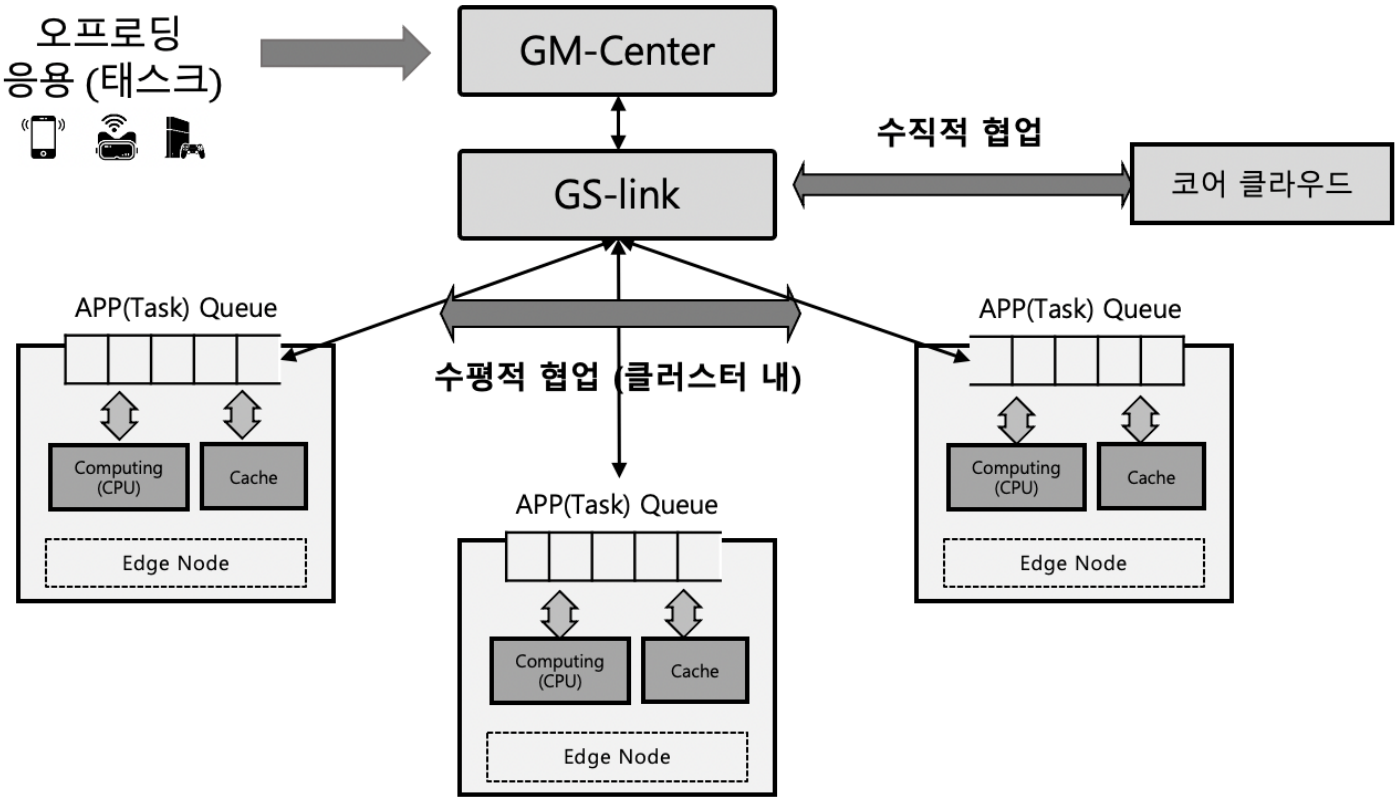
2 협업 응용(태스크) 오프로딩 모델

- 협업 오프로딩 모델 (추상적 모델)

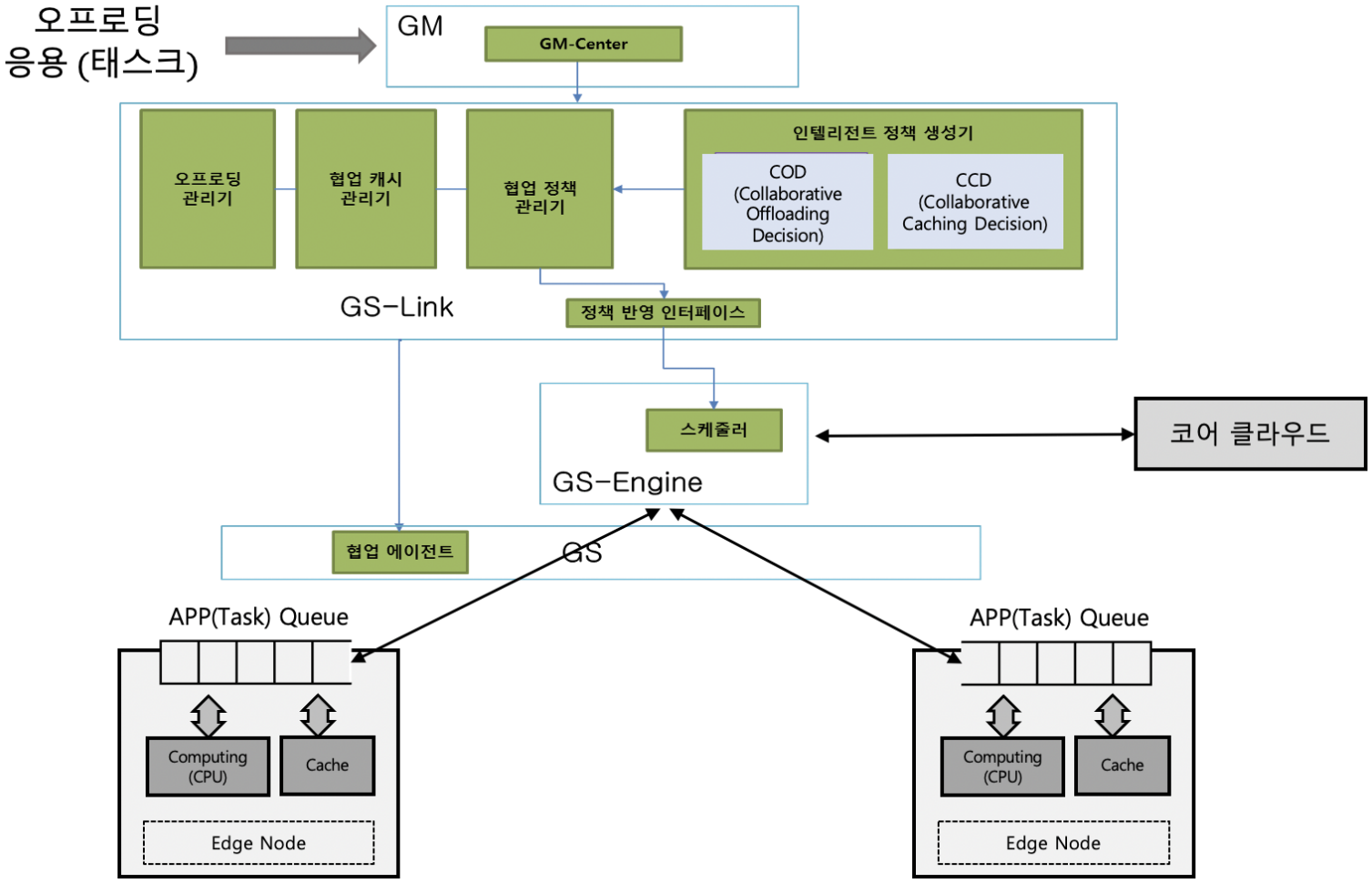


2 클라우드 엣지 서비스 협업 프레임워크

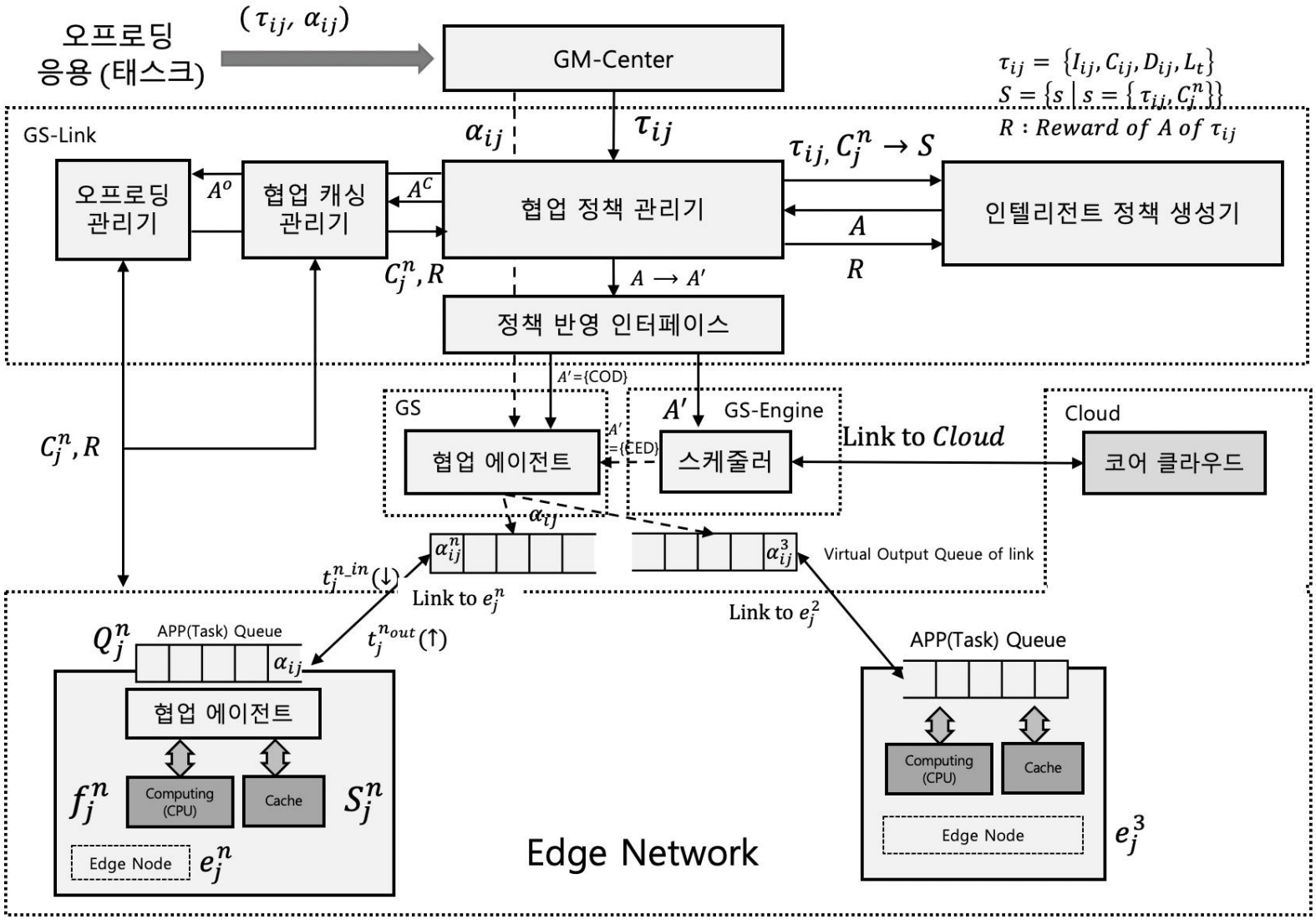
- 싱글클러스터 기반 서비스 협업 프레임워크



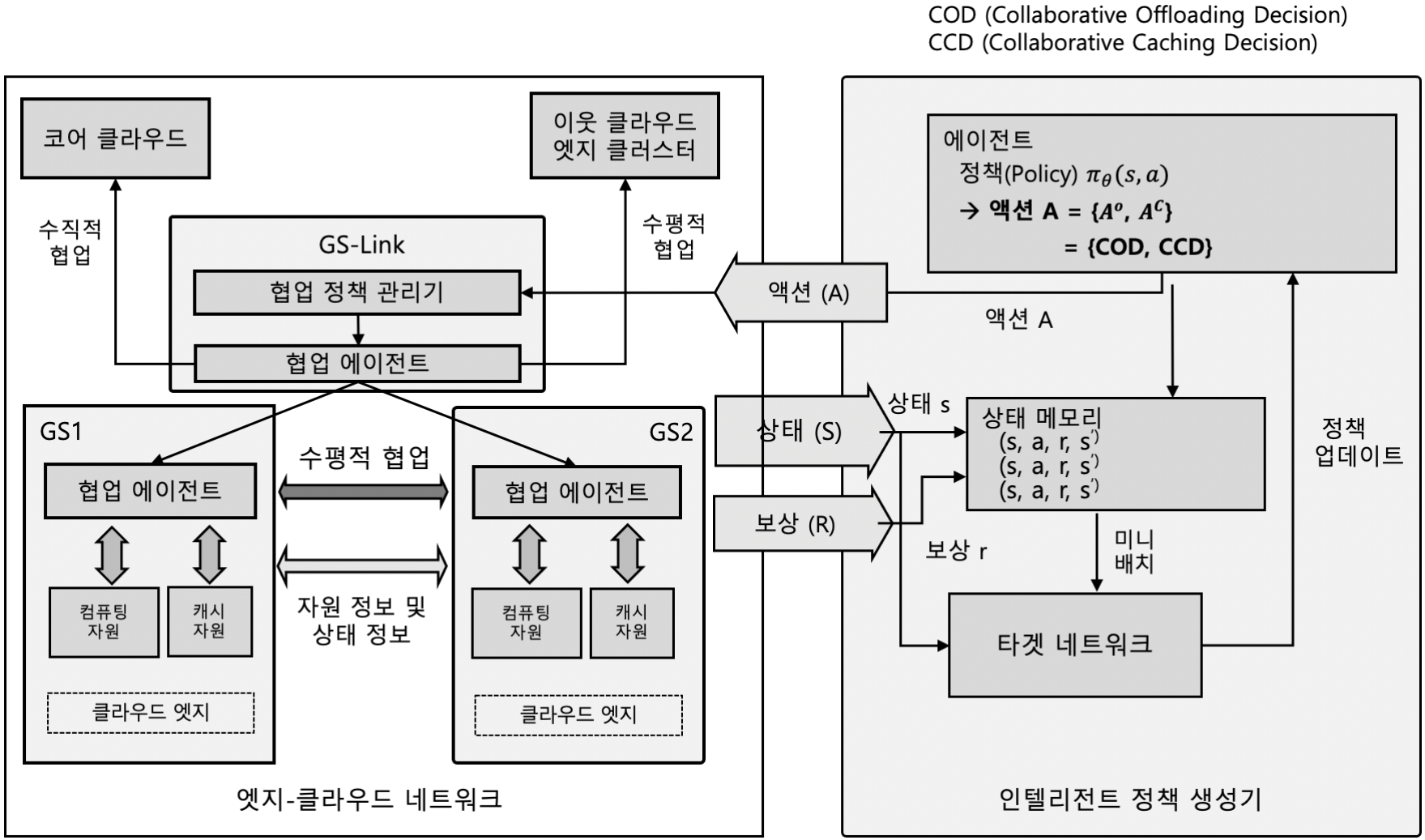
3 GS-Link 구조


오프로딩
응용 (태스크)
GM
GM-Center
오프로딩
관리기
협업 캐시
관리기
협업 정책
관리기
인텔리전트 정책 생성기
COD
(Collaborative
Offloading
Decision)
CCD
(Collaborative
Caching Decision)
GS-Link
정책 반영 인터페이스
스케줄러
GS-Engine
코어 클라우드
협업 에이전트
GS
APP(Task) Queue
Computing
(CPU)
Cache
Edge Node
APP(Task) Queue
Computing
(CPU)
Cache
Edge Node

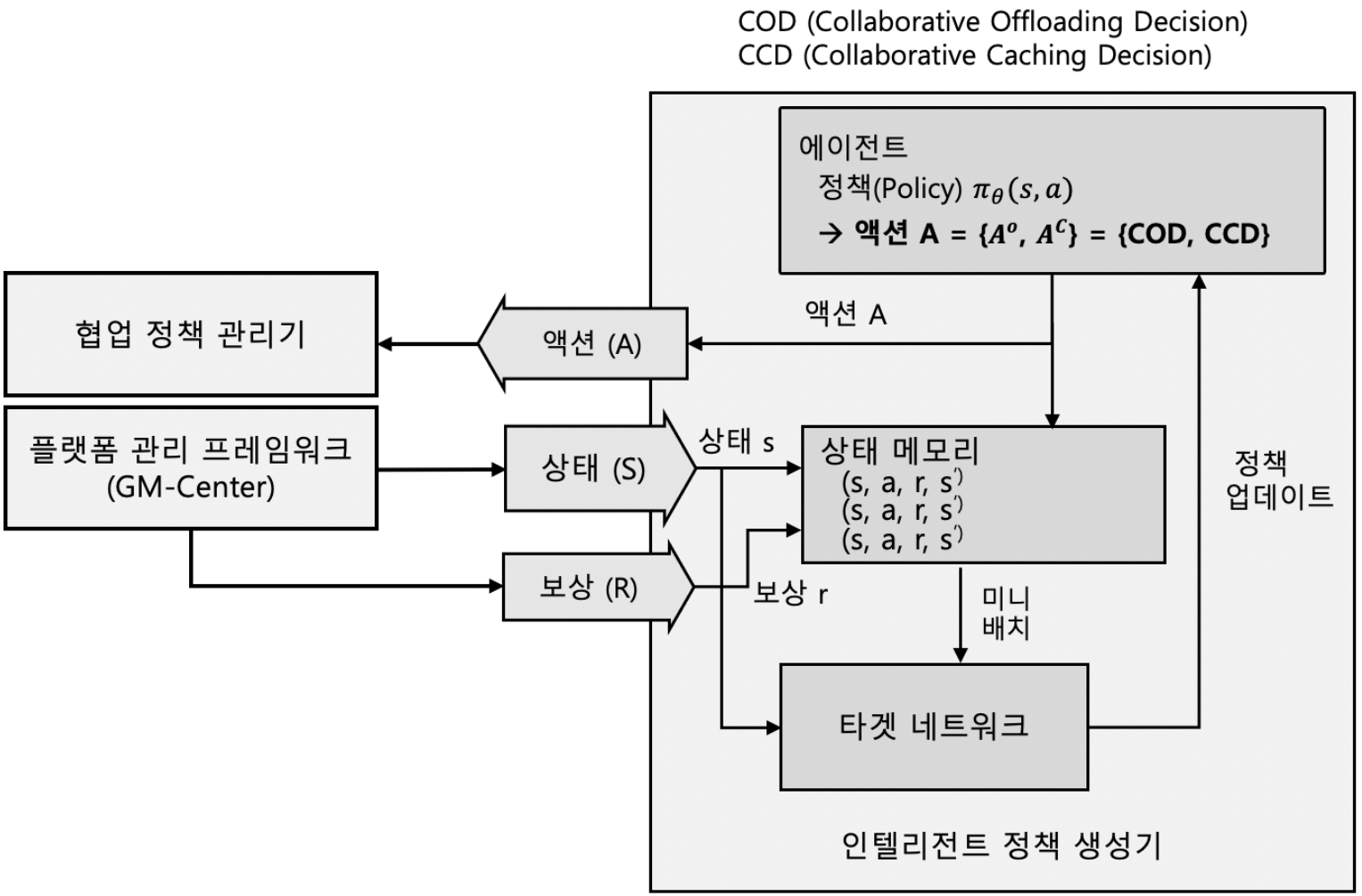
3 GS-Link 구조 내 메시지 정의



3 인텔리전트 정책 생성기



3 협업 서비스 정책 모델


COD (Collaborative Offloading Decision)
CCD (Collaborative Caching Decision)
에이전트
정책(Policy) $\pi_\theta(s, a)$
→ 액션 A = {A^o, A^C} = {COD, CCD}
협업 정책 관리기
액션 (A)
액션 A
플랫폼 관리 프레임워크
(GM-Center)
상태 (S)
상태 s
상태 메모리
(s, a, r, s')
(s, a, r, s')
(s, a, r, s')
정책
업데이트
보상 (R)
보상 r
미니
배치
타겟 네트워크
인텔리전트 정책 생성기

3 협업 오프로딩 지원

- 협업 오프로딩 모델 (Collaborative Offloading Model)
  - 전체 이관(Full Offloading) 또는 부분 이관(Partial Offloading)
- 협업 오프로딩 과정
  - 오프로딩 서비스 타입 결정 (전체 이관, 부분 이관 결정)

  - (전체 이관 경우, COS-1) → 오프로딩 태스크를 위한 최상의 엣지 결정
  - (부분 이관 경우, COS-2) → 태스크 분할 → 태스크 별 엣지 결정
  - (부분 이관 경우, COS-3) → 태스크 분할 → 태스크((마이크로)서비스)유무 & 성능 기반 엣지 결정 & 태스크 채인 설정

  - → 태스크 이동 → 태스크 실행 → 태스크 결과 리턴

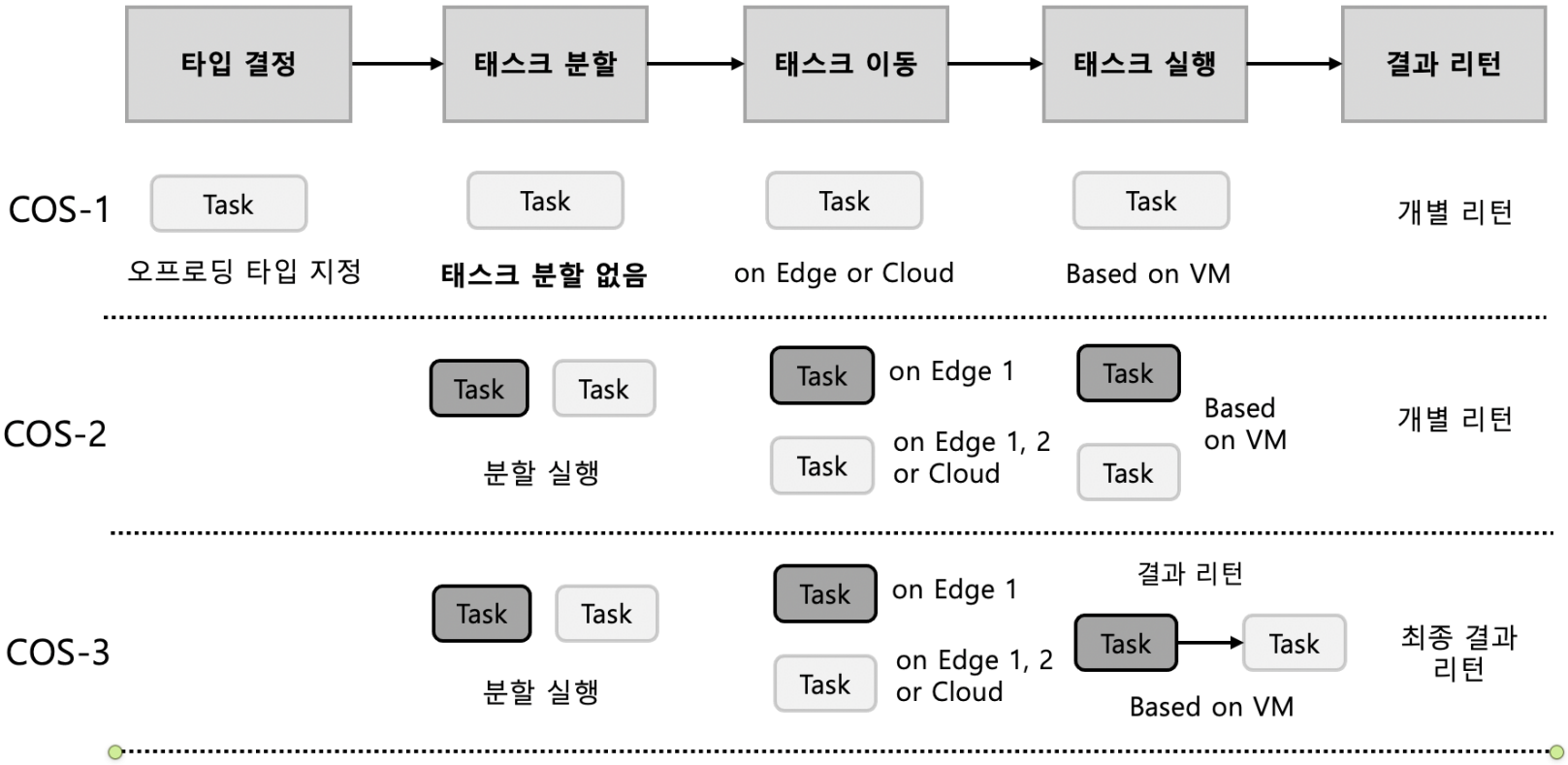
- 협업 오프로딩 과정

3 멀티클러스터 기반 수평적 협업

- 멀티클러스터 기반 수평적 협업 오프로딩 시나리오

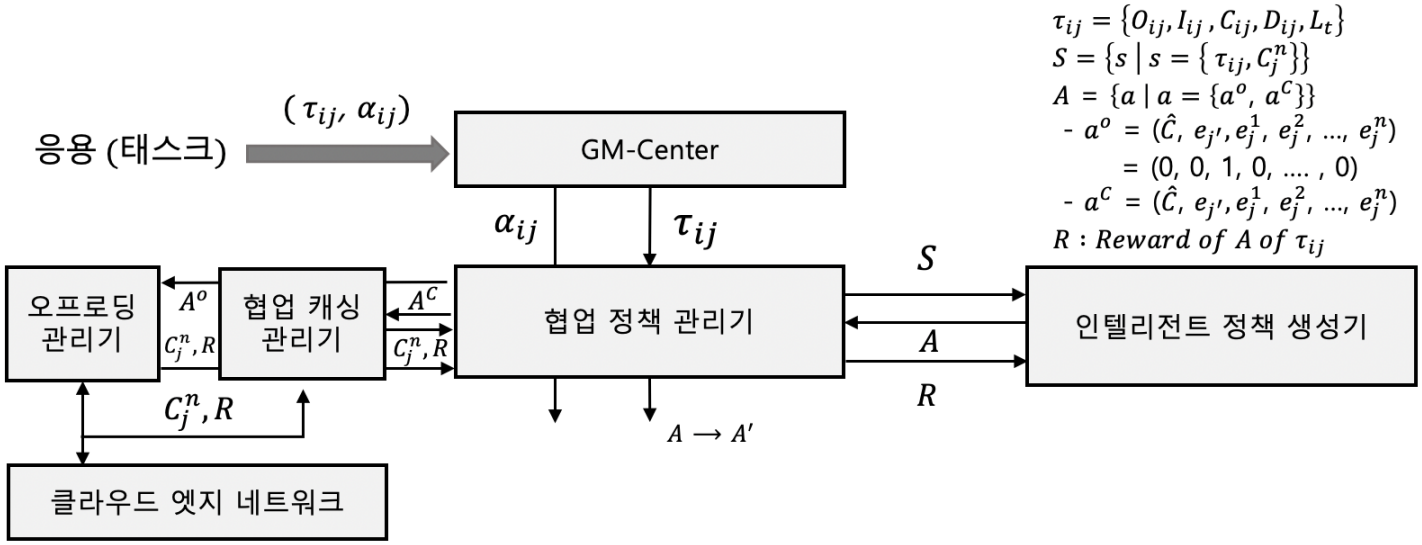
3 오프로딩 지원 방법

- **협업 오프로딩 과정**



3 오프로딩 정책 (π)

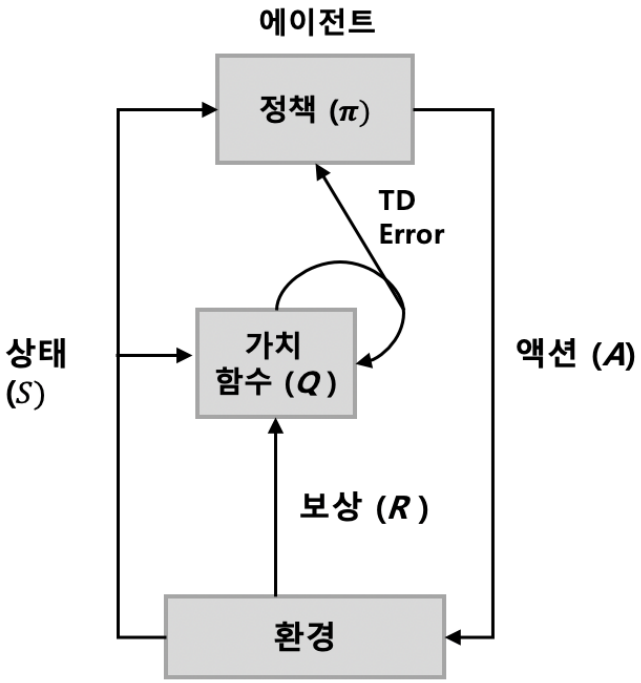
- **정책 생성 모델**



- **협업 서비스 정책**
  - <u>협업 오프로딩 정책</u>, <u>분산-협업 캐싱 정책</u>, <u>고속 서비스 이동 정책</u>
  - $\pi_n^O, n = \{1, 2, \ldots n\}$, $\pi_n^C, n = \{1, 2, \ldots n\}$, $\pi_n^M, n = \{1, 2, \ldots n\}$

3 딥러닝 기반 오프로딩 & 캐싱 정책

- 강화학습 정책 모델



$$S = \{s \mid s = \{\tau_{ij}, C_j^n\}\}$$

$$A = \{a \mid a = \{a^o, a^C\}\}$$

- $a^o = (\hat{C}, e_{j'}, e_j^1, e_j^2, \ldots, e_j^n)$
  $= (1, 0, 1, 0, \ldots, 0)$: *COS-2*
- $a^C = (\hat{C}, e_{j'}, e_j^1, e_j^2, \ldots, e_j^n)$

$$Q(s_t, a_t) = R(s_t, a_t) + \gamma Q(s_{t+1}, a_{t+1})$$

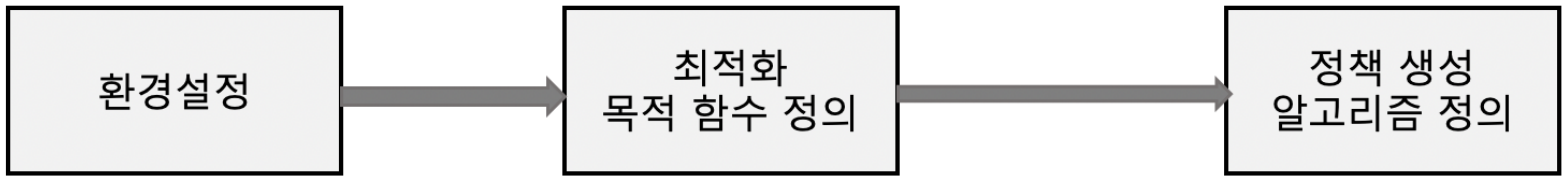
$R_t = \sum_{k=0}^{T} \alpha^k r_{t+k+1}$ (future reward at time t)

*$\alpha \in (0, 1]$ is a discount factor

r_t *is reward of action* a_t *in* s_t

3 딥러닝 기반 정책 생성

- **액션 생성**



- *MIN* (t_i: *TST(Task Service Latency))*
  * *TST = Task return time to node*
  - *Task request time to edge*

- $S = \{s \mid s = \{\tau_{ij}, C_j^n\}\}$
- $A = \{a \mid a = \{a^o, a^C\}\}$
- $Q(s_\tau, a_\tau)$
- $R = \sum_{k=0}^{T} \alpha^k r_{\tau+k+1}$
  $(a_t \rightarrow r_t)$

3 최적화 목적 함수 정의 (COS-1)

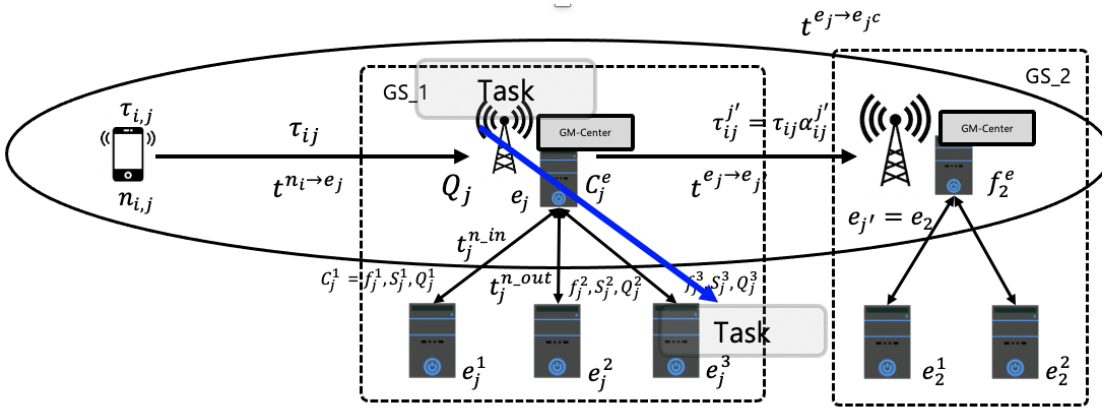
- 최적화 목적 함수
  - 모든 태스크의 평균 예상 서비스 지연

$$Min\ (T_{all} = \frac{\sum_{i=1}^{N} t_i}{N})\ \text{(N: the number of all tasks)}$$

  - 태스크(응용) 서비스 타임

$$\text{- COS-1: } t_{ij} = t^{n_i \to e_j} + Q_j + t_j^{n\_in} + Q_j^n + \frac{C_{ij}}{f_j^n} + t_j^{n\_out} + t^{e_j \to n_i}$$

$$* \ t^{n_i \to e_j} = \frac{D_{ij}}{R^{n_i \to e_j}},\ t_j^{n\_in} = \frac{D_{ij}}{R_j^{n\_in}},\ t_j^{n\_out} = \frac{D_{ij}^r}{R_j^{n\_in}},\ t^{e_j \to n_i} = \frac{D_{ij}^r}{R^{e_j \to n_i}}$$

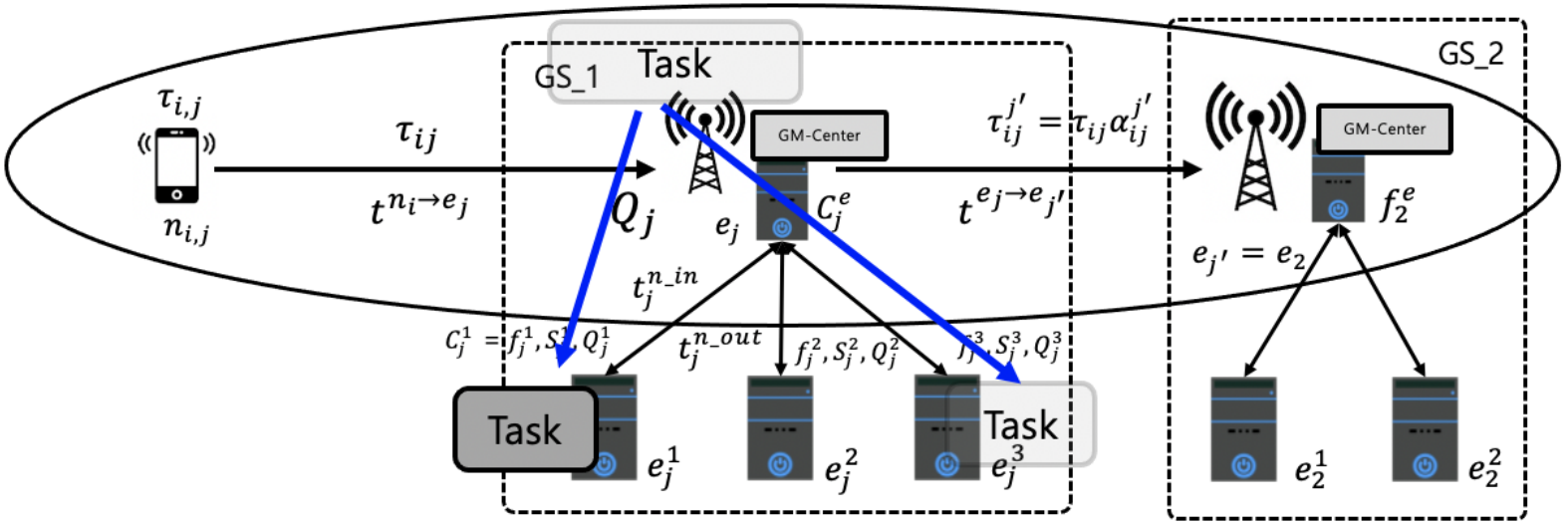


3 최적화 목적 함수 정의 (COS-2)

- **서비스 타입 2의 태스크(응용) 서비스 타임 :**
  - the offloading ratio of the task data

- COS-2: $t_{ij} = t^{n_i \to e_j} + Q_j + \max(\alpha_{ij}^1 (t_j^{1\_in} + D_{ij} \times \frac{C_{ij}}{f_j^1} + t_j^{1\_n\_out}), \alpha_{ij}^2 (t_j^{3\_in} + D_{ij} \times \frac{C_{ij}}{f_j^1} + t_j^{3\_n\_out})) + t^{e_j \to n_i}$

  * $\alpha_{ij}^1 + \alpha_{ij}^2 = 1$



3 보상 정의

- 보상
  - 최적화 목적 함수와 관련 있도록 디자인해야함
  - 모든 태스크의 평균 예상 서비스 지연을 최소화 할 수 있도록 보상 지불 방법을 정의하는 게 중요함

$$- Min\ (T_{all} = \frac{\sum_{j=1}^{N} \sum_{i=1}^{N} t_{ij}}{N}\ )\ \text{(N: the number of all tasks)}$$

$$- R_{\tau} = \sum_{k=0}^{T} \alpha^{k}\ r_{\tau+k+1}\ \text{(future reward at time t)}$$

  - 태스크 수행이 일정하다면

$$- r_{\tau} = \begin{cases} \pi t = avg\ (t_{ij}) - t_{ij},\ t_i < avg\ (t_{ij}) \\ -\gamma \pi t = t_{ij} - avg\ (t_{ij}),\ t_i > avg\ (t_{ij}) \\ \eta, \qquad\qquad if\ task\ is\ failed \end{cases}$$

$*\ \pi$ is the price of edge,
η is negative(penalty) value,
γ is penalty value $(0 < \gamma < 1)$

3 액션 생성을 위한 강화 학습 알고리즘

- **Double DQN**
  - **Long-term cumulative discount reward value $Q(s_t, a_t)$**
    - $Q(s_i, a_t) = E_\pi[\sum_{\tau=0}^{\infty} r_\tau(S_\tau, A_\tau)]$
  - **Optimization objective function**
    - $Q_\pi(s, a) = MAX_{A_\tau} E_\pi[\sum_{\tau=0}^{\infty} r_\tau(S_\tau, A_\tau)])$

3 서비스 협업 기술의 21년도 개발 계획

- 인텔리전트 정책 생성기 개발
  - 강화 학습 기반 정책 생성기 개발 완료
  - 최적 알고리즘 선정

- 분산캐쉬 사용을 위한 정책 생성 알고리즘 개발
- 고속 응용 이동 정책 생성 알고리즘 개발 및 정책 생성 학습 모델 개발
  - 엣지 공유캐시를 통한 고속 응용 이동 알고리즘 개발
    - 공유캐시 내 고속 응용 이동을 위한 딥러닝(어텐션+LSTM 알고리즘) 기반 응용 배치 선정 및 정책 알고리즘 개발
  - 협업 엣지 네트워크 모델 기반 고속 응용 이동 정책 생성 알고리즘 및 정책 생성을 위한 학습 모델 개발

감사합니다.

http://gedge-platform.github.io

GEdge Platform 코어 개발자
윤주상(jsyoun@deu.ac.kr)