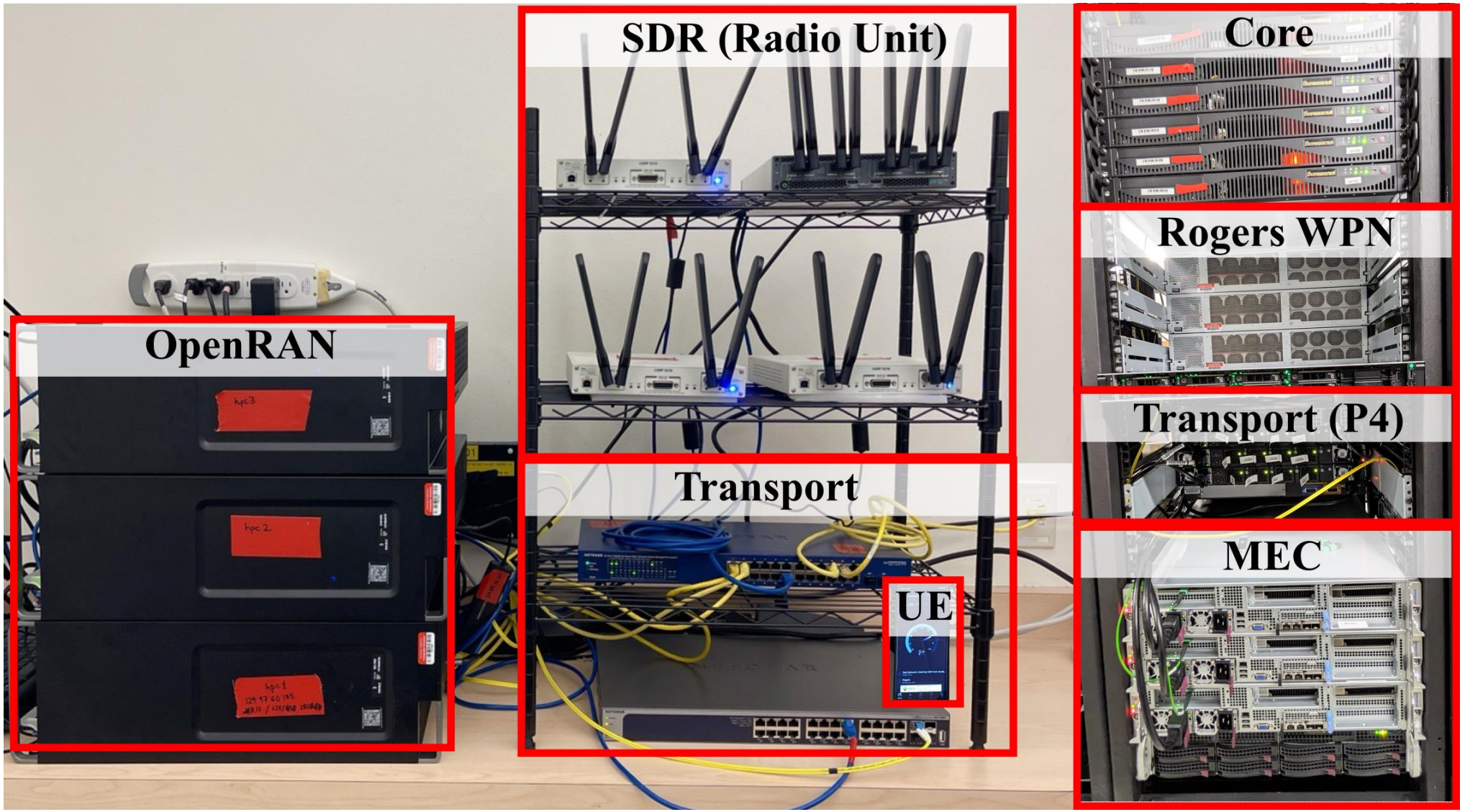
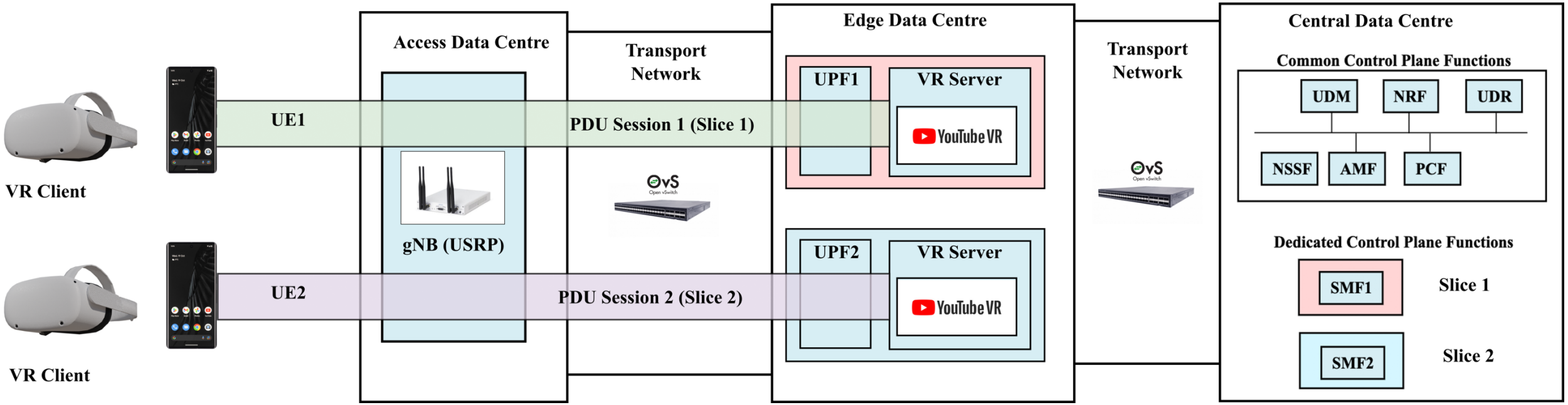
5G Network Slicing and Orchestration
Demo

Raouf Boutaba
David R. Cheriton School of Computer Science
University of Waterloo

Network Slicing Testbed at UW



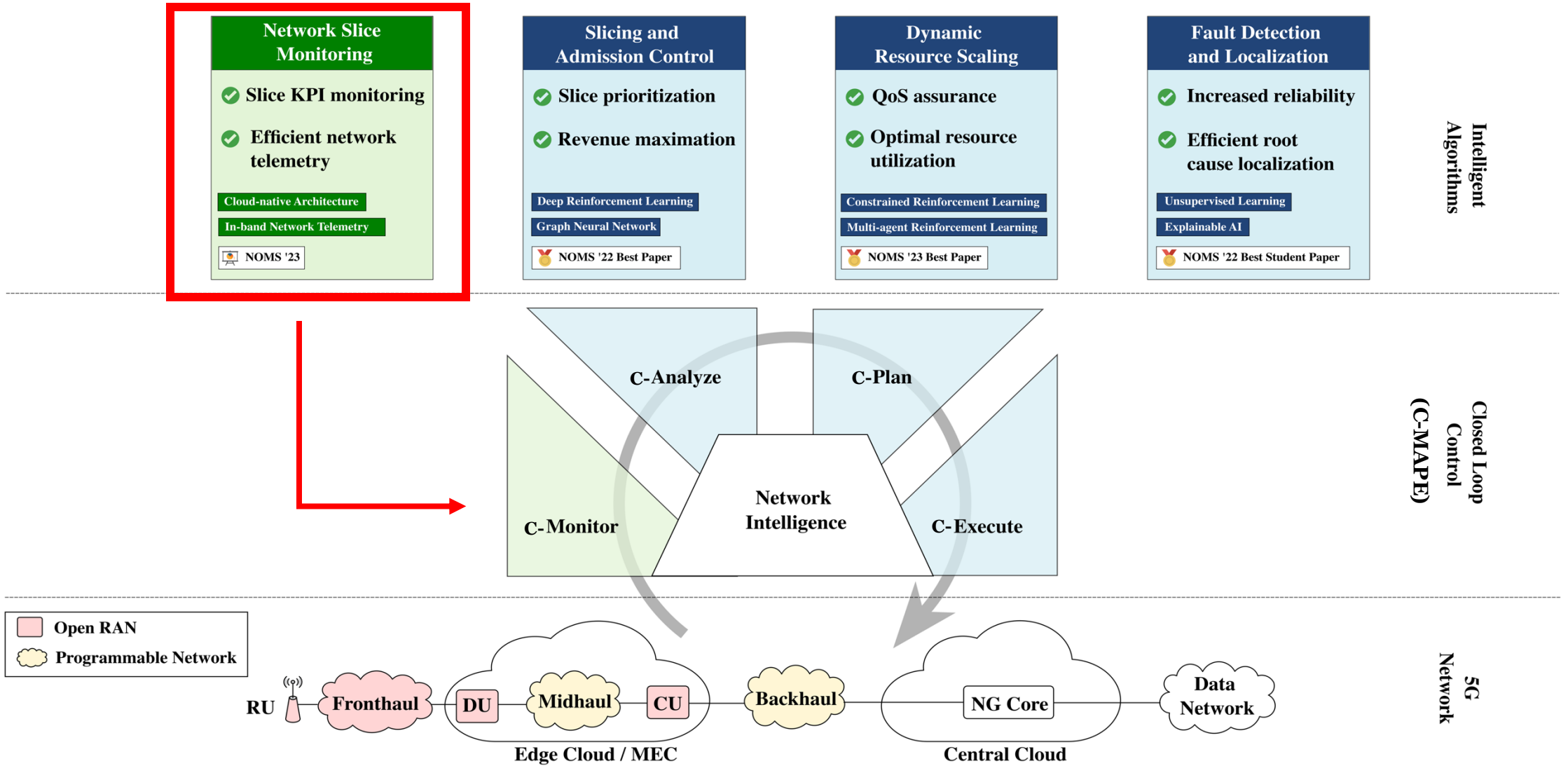
Use-case: VR Streaming



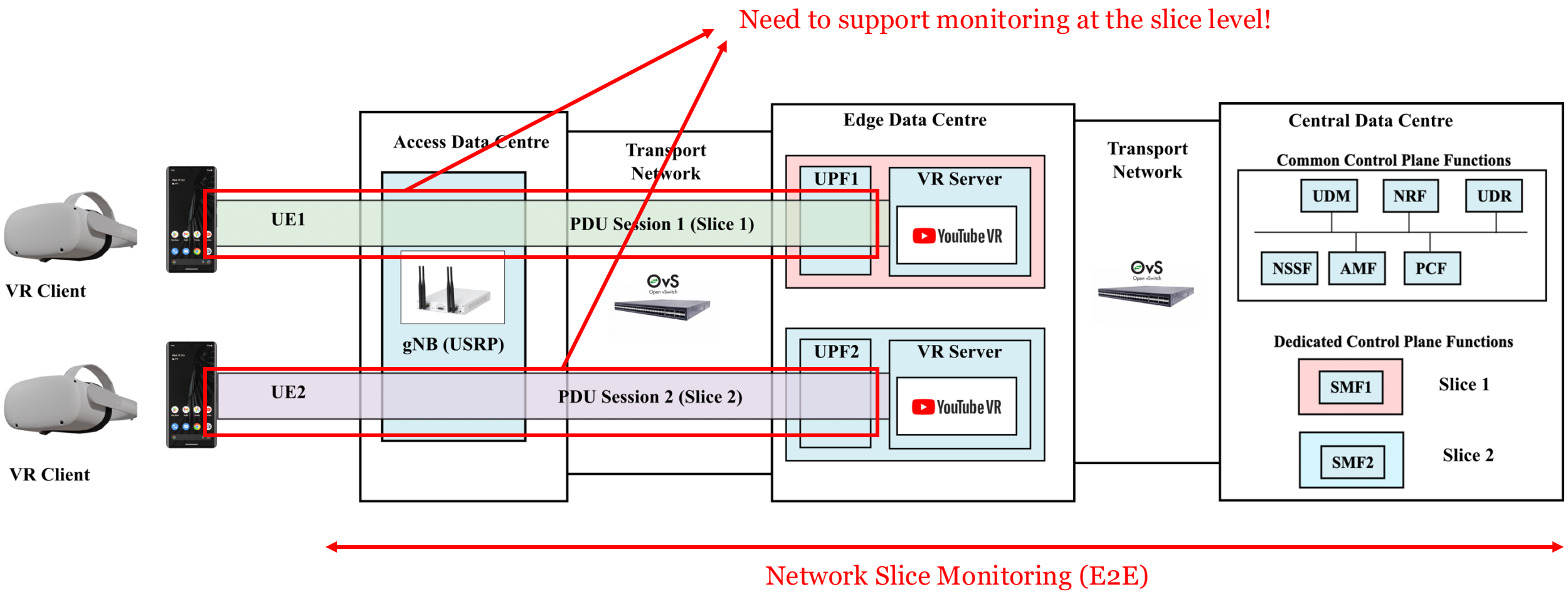
UNIVERSITY OF WATERLOO

NETWORK SLICE MONITORING

Network Slice Monitoring

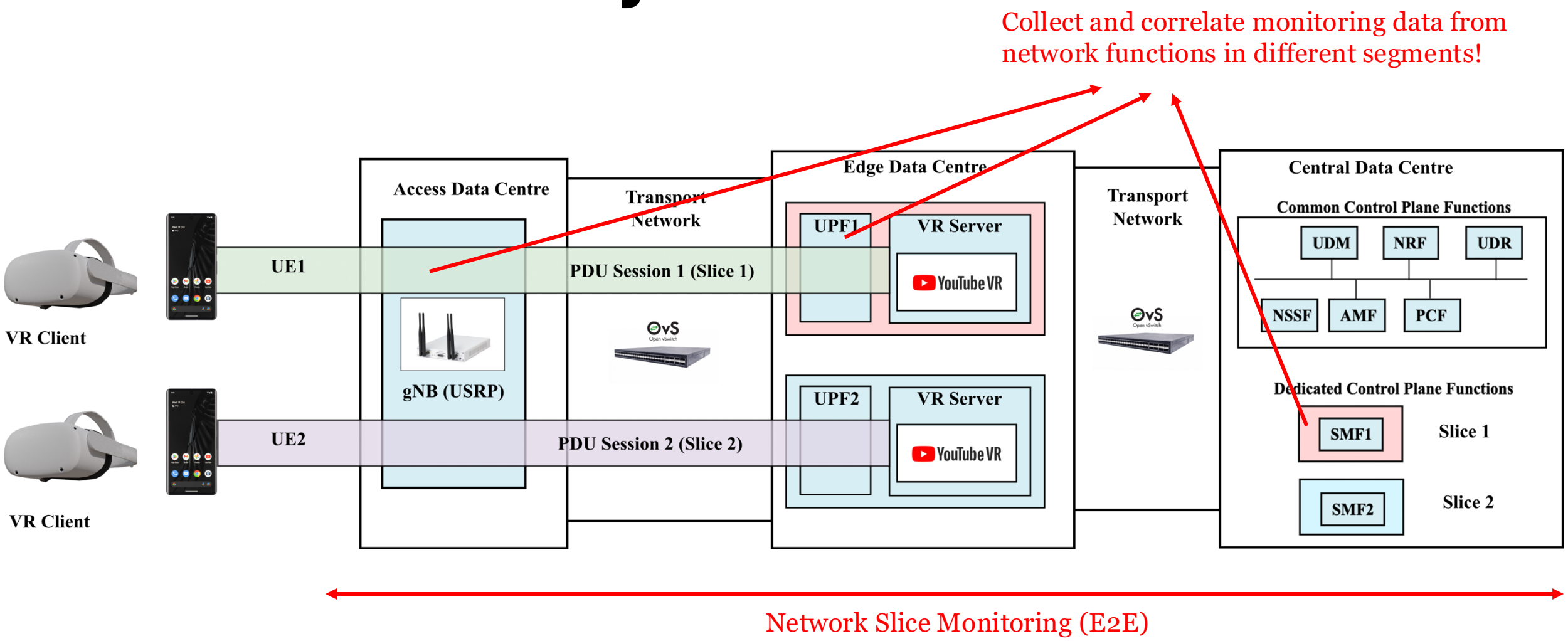


Network Slice Monitoring



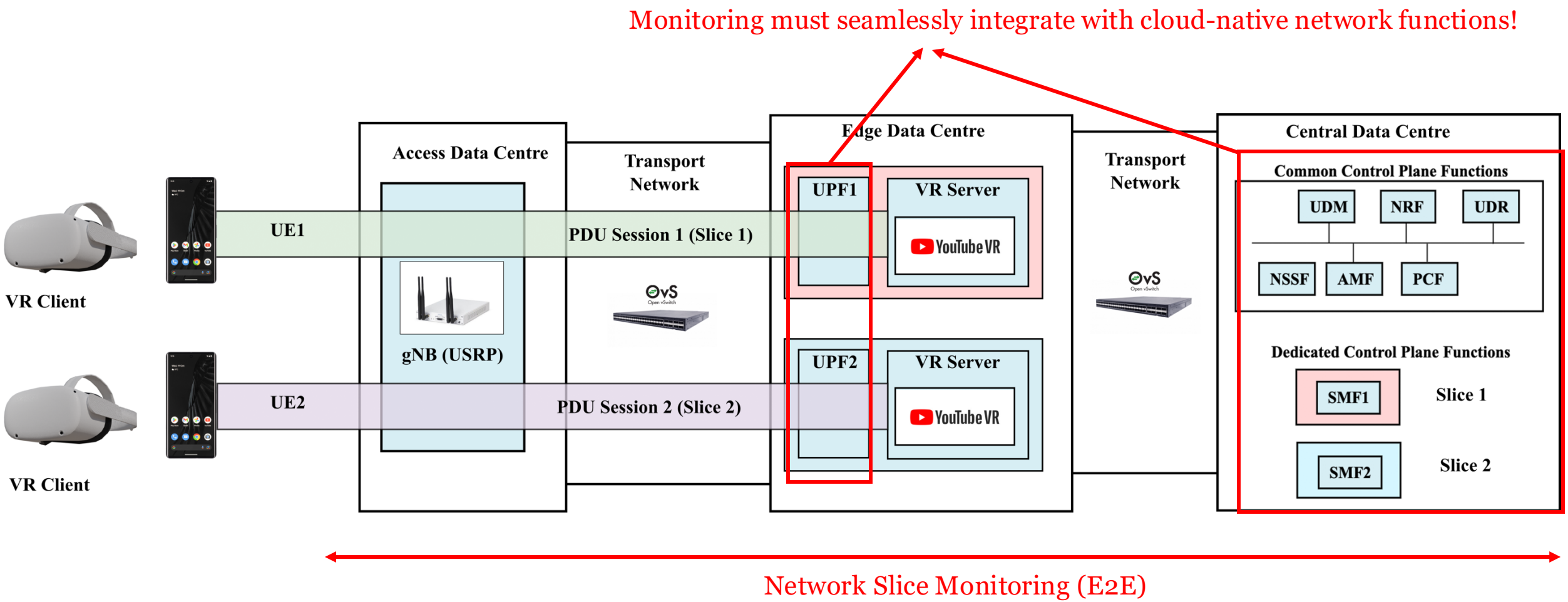
UNIVERSITY OF WATERLOO

Network Slice Monitoring



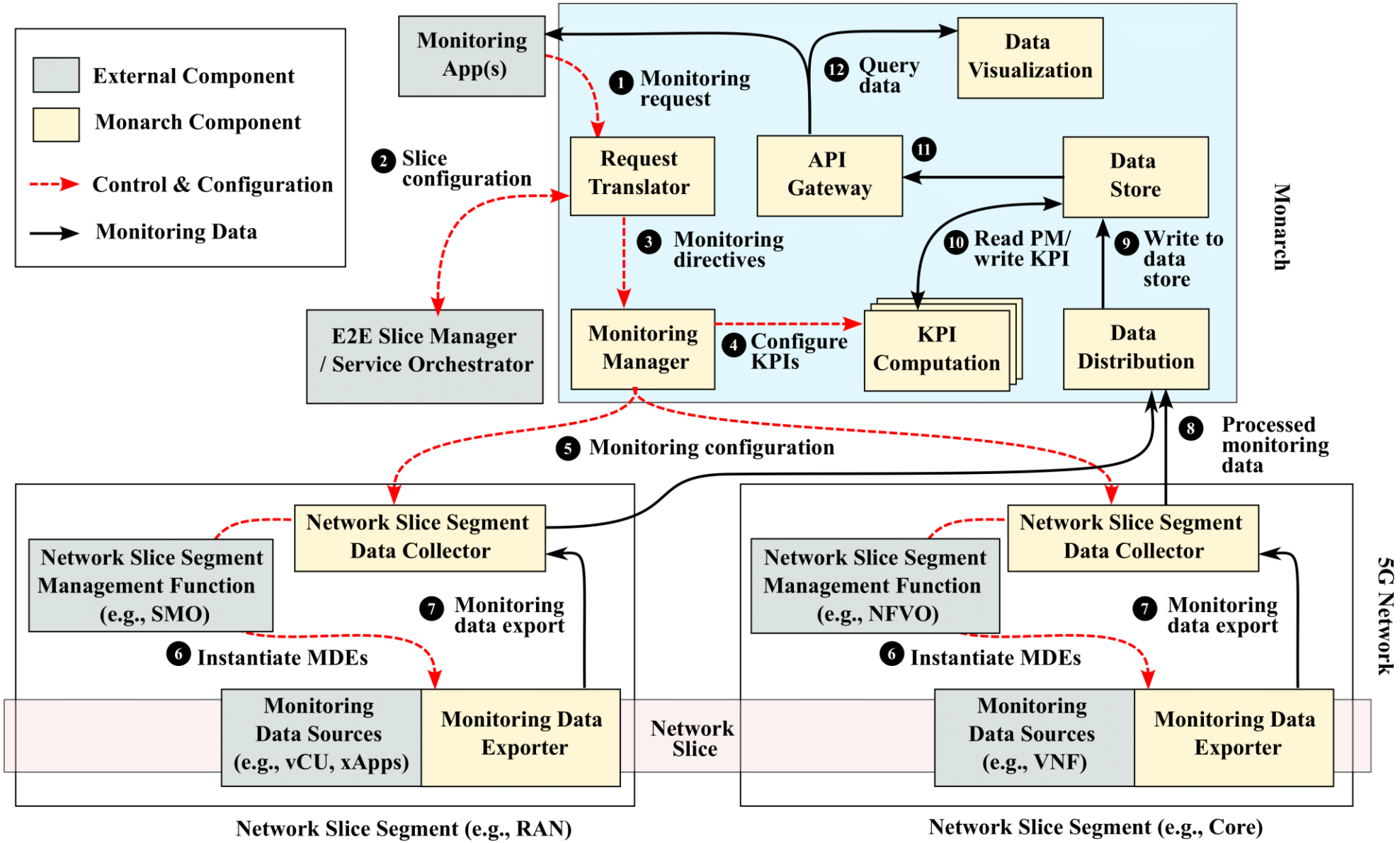
UNIVERSITY OF WATERLOO

Network Slice Monitoring

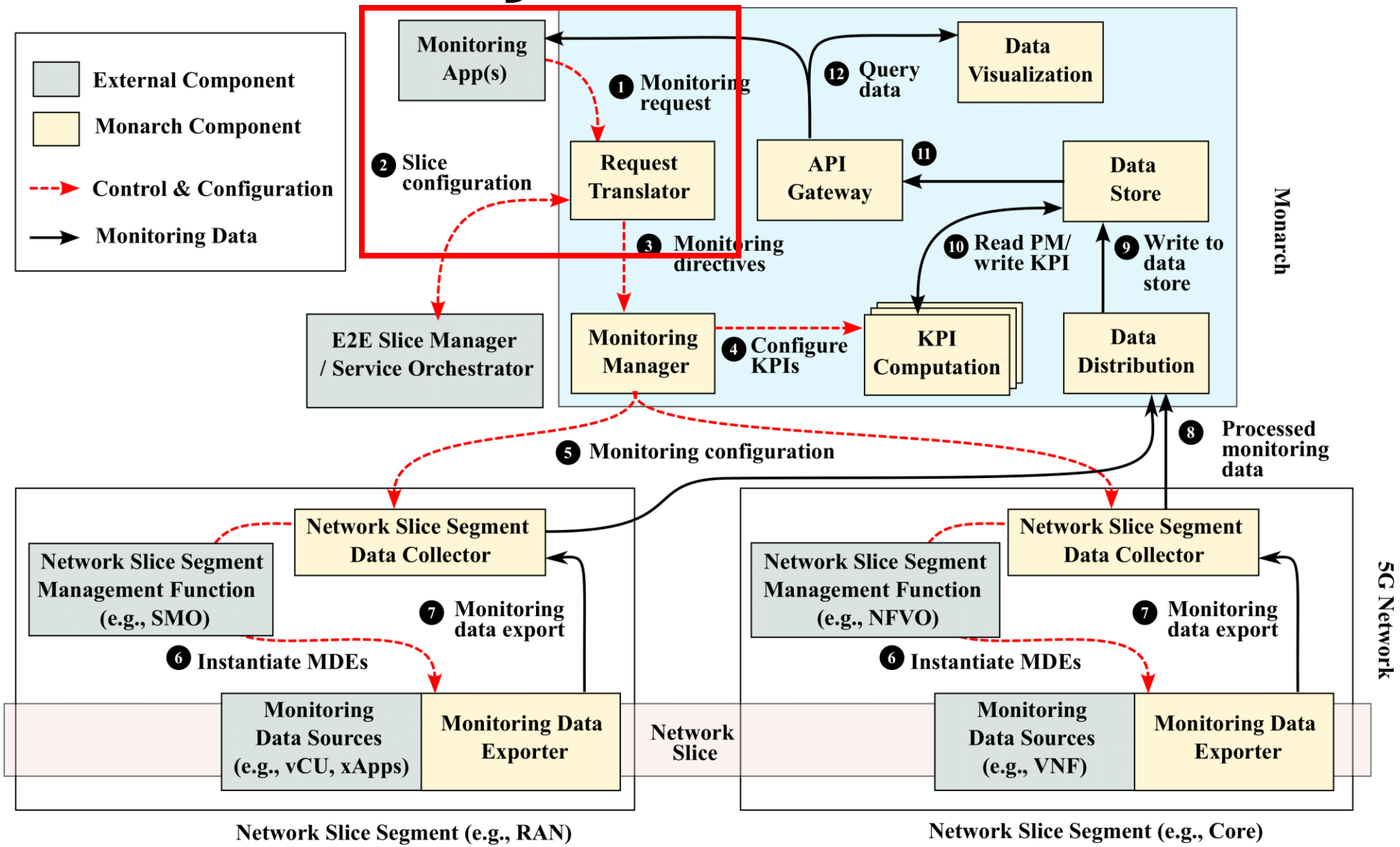


UNIVERSITY OF WATERLOO

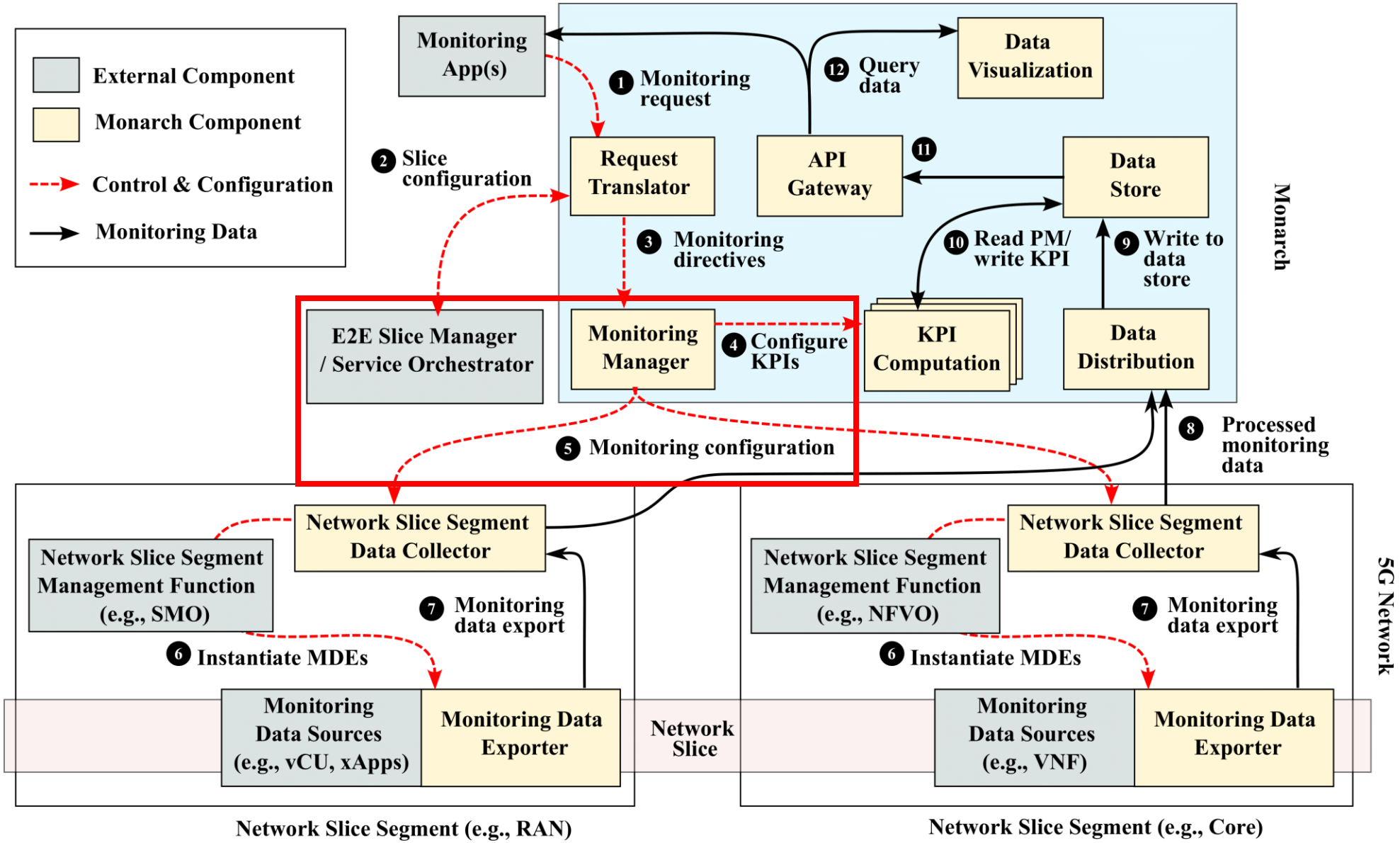
Network Slice Monitoring with Monarch



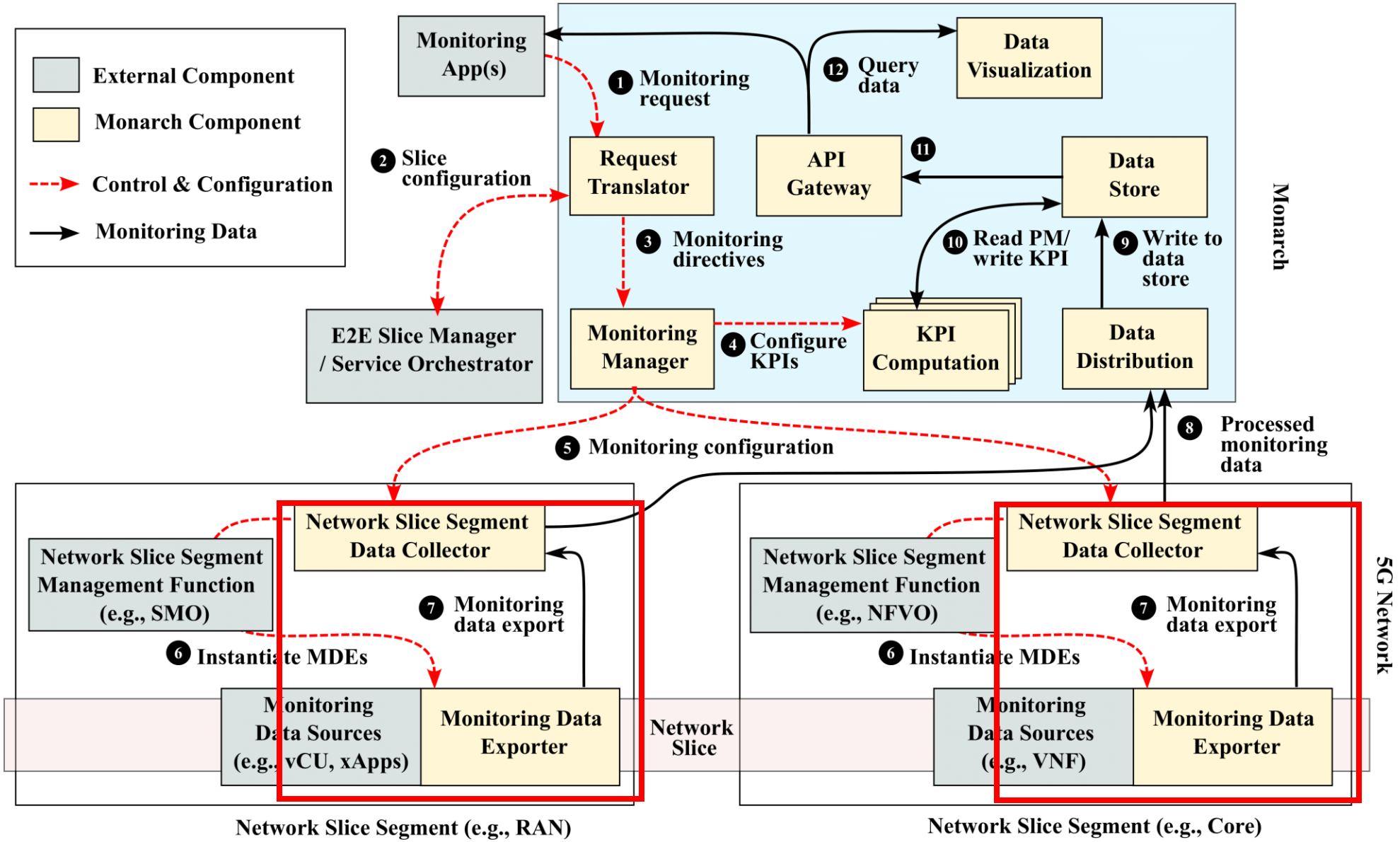
Network Slice Monitoring with Monarch



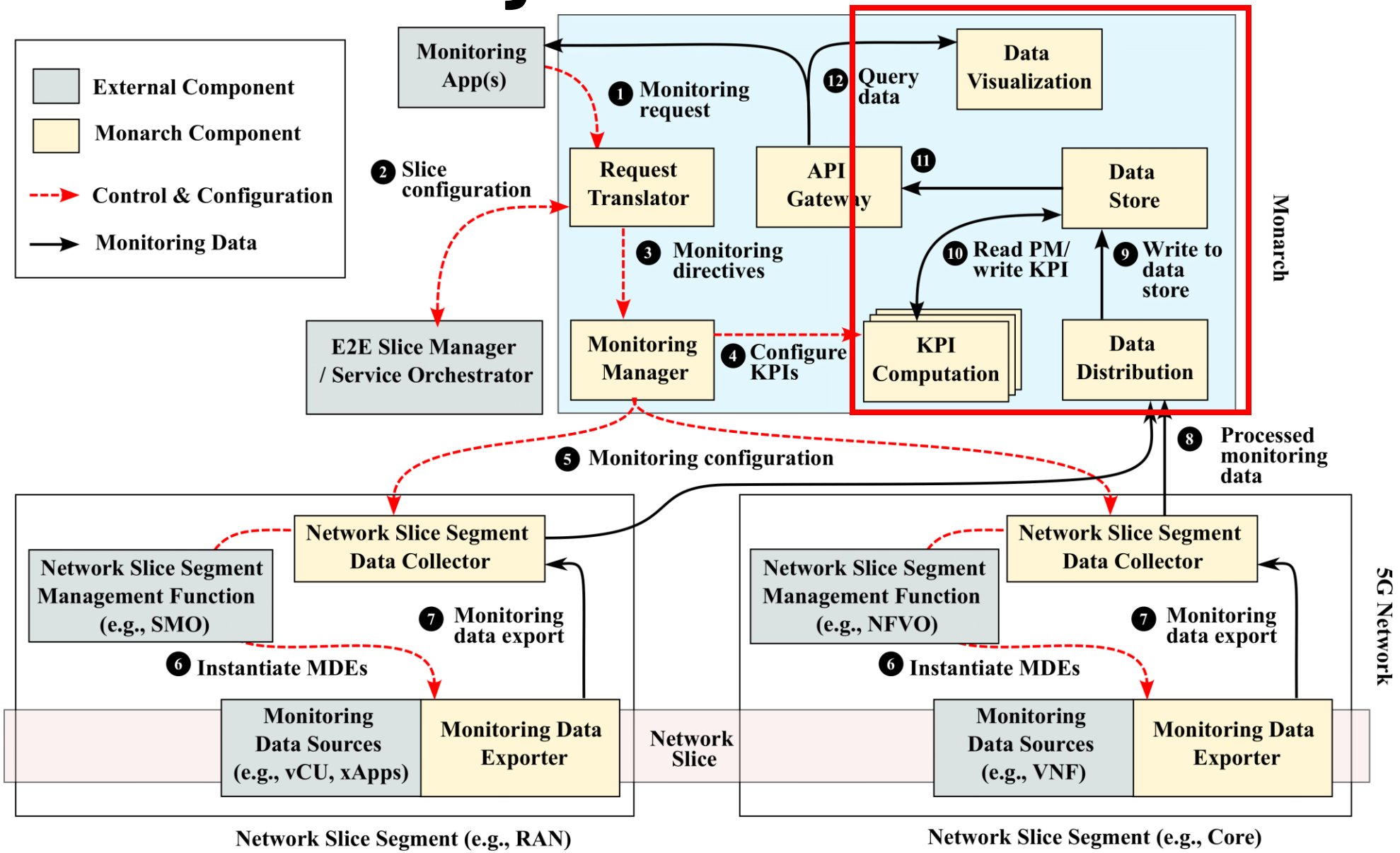
Network Slice Monitoring with Monarch



Network Slice Monitoring with Monarch



Network Slice Monitoring with Monarch



USE-CASE: VR STREAMING

DYNAMIC RESOURCE SCALING

Dynamic Resource Scaling (1/2)

DRS: Main Idea

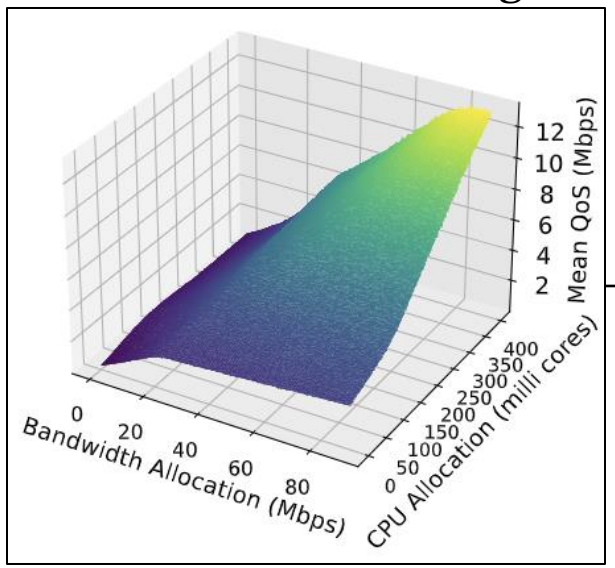
Scaling resources allocated to slices based on predicted traffic

DRS: Objective

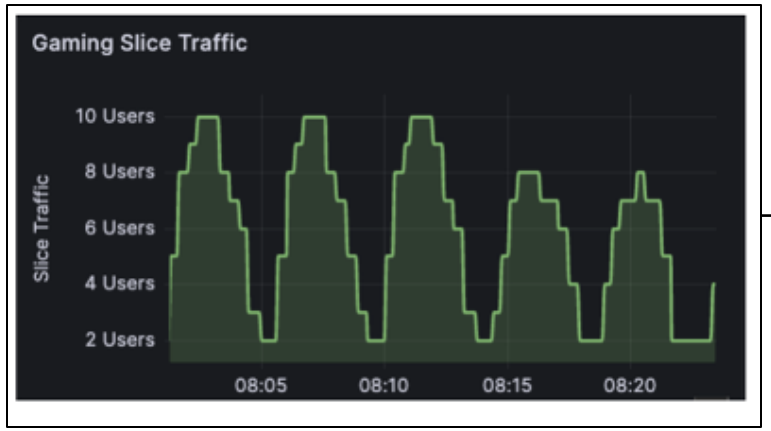
- Minimize resource allocation
- Maintain minimum QoS threshold

Dynamic Resource Scaling (2/2)

Network Modelling



Slice Traffic Forecast



Resource Scaling Algorithm

Algorithm 1 Resource Allocation Algorithm

Input: Traffic $\boldsymbol{x}^s_{\tau_i}$, Network Model $f^s_{QoS}(\boldsymbol{x}^s_{\tau_i}, \boldsymbol{r}^s_{\tau_i})$, QoS threshold q^s_{thresh}, QoS degradation threshold β^s_{thresh}, $\tau_{1,max}, \tau_{2,max}, \alpha_1, \alpha_2, \alpha_3, \epsilon_1, \epsilon_2$

Output: Optimal resource allocation vector $\boldsymbol{r}^s_{\tau_i}$

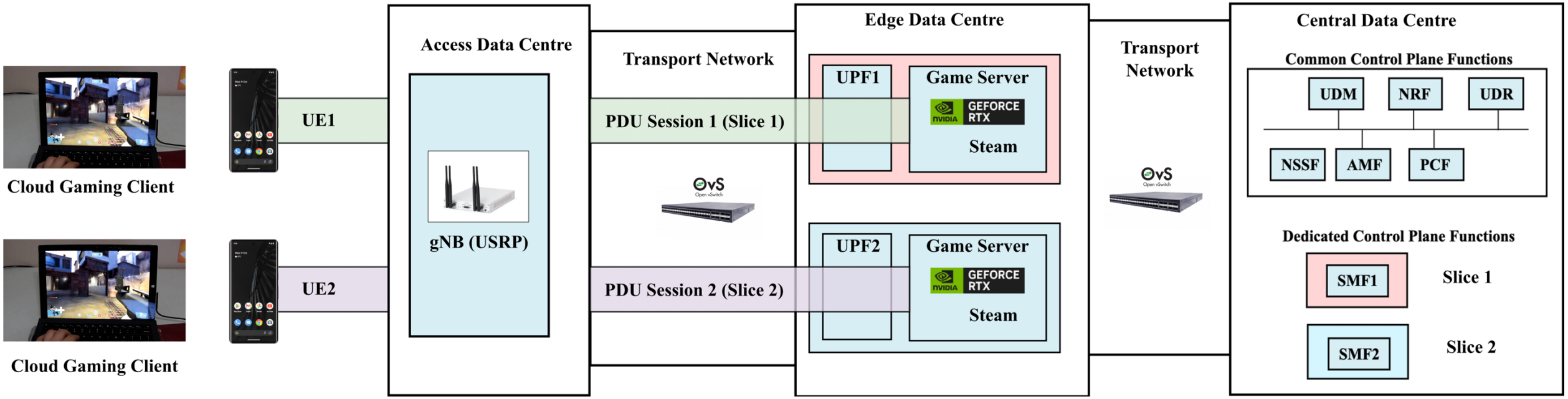
Initialize $\boldsymbol{\lambda}, \boldsymbol{\mu}$, LB $= 0$, UB $= \infty$, $\tau_1 = 0, \tau_2 = 0$
while $\frac{\text{UB}-\text{LB}}{\text{UB}} > \epsilon_1$ **or** $\tau_1 < \tau_{1,max}$ **do**
  $\mathbf{r} \leftarrow \texttt{Gridsearch}(\boldsymbol{x}^s_{\tau_i}, f_{QoS}(\boldsymbol{x}^s_{\tau_i}, \mathbf{r}))$
  while $|\nabla_r \hat{\mathcal{L}}| > \epsilon_2$ **or** $\tau_2 < \tau_{2,max}$ **do**
    $\mathbf{r} \leftarrow [\mathbf{r} - \alpha_1 \nabla_r \hat{\mathcal{L}}]^+$
    $\tau_2 \leftarrow \tau_2 + 1$
  end while
  $\lambda_s \leftarrow [\lambda_s + \alpha_2(\beta^s - \beta^s_{thresh})]^+, \forall s$
  $\mu_k \leftarrow [\mu_k + \alpha_3(\sum_{s \in S} r^{s,k} - R^k)]^+, \forall k$
  LB $= \max(\text{LB}, \mathcal{L}(\mathbf{r}, \boldsymbol{\mu}, \boldsymbol{\lambda}))$
  UB $= \min(\text{UB}, \sum_{s \in S} \eta^\intercal \mathbf{r}^s)$
  $\tau_1 \leftarrow \tau_1 + 1$
end while
return r

Resource Allocation →

UNIVERSITY OF WATERLOO

USE-CASE: CLOUD GAMING

Use-case: Cloud Gaming



UNIVERSITY OF WATERLOO