NeoTec Use Case Discussion

Nabeel Cocker
Red Hat

Luay Jalil
Verizon

March 19th, 2025

Red Hat

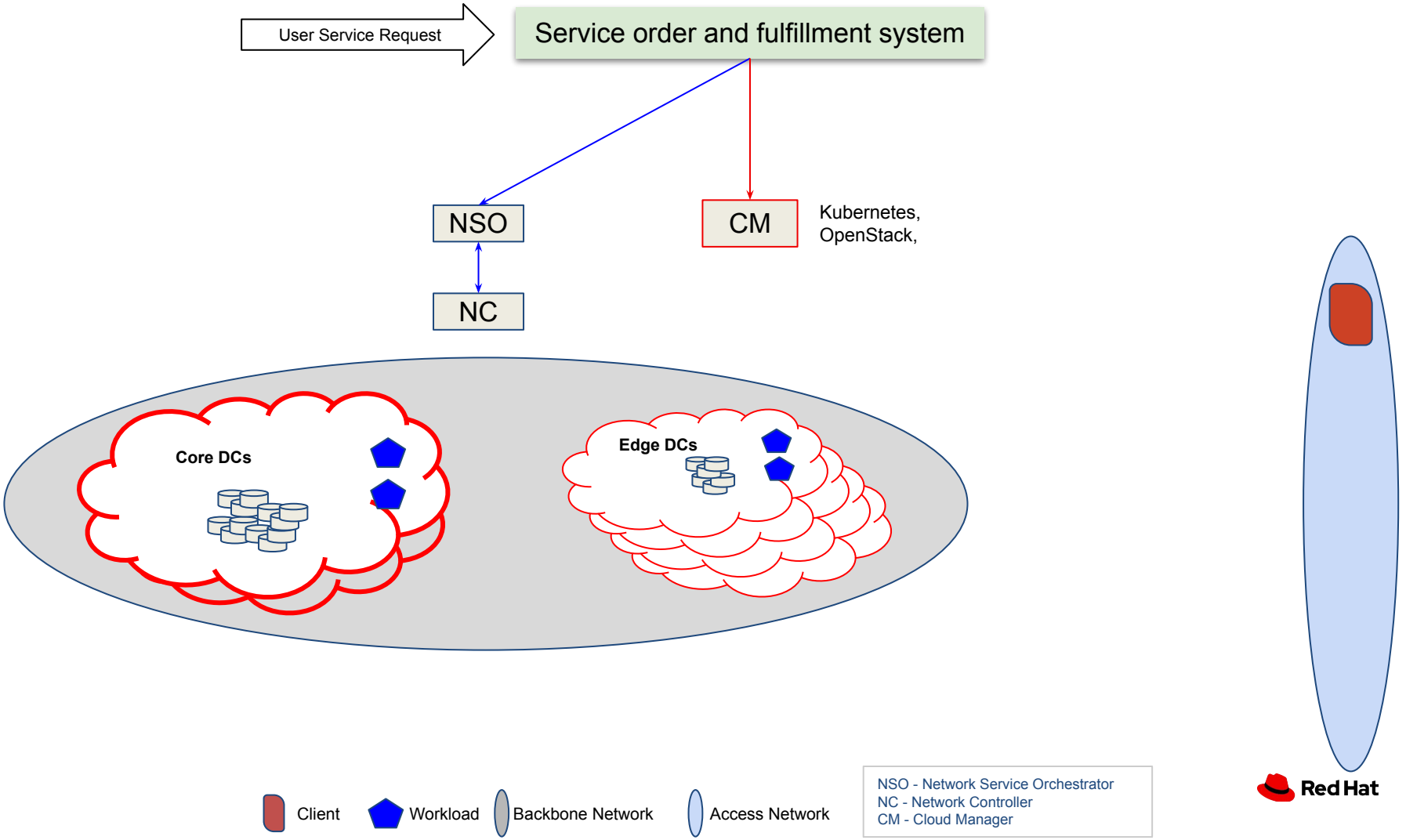
Edge computing

- Architecture that provides cloud computing capabilities at the edge of the network
- Placement of small footprint of compute resources closer to the end users or sources of data
- Main use case is reduced latency for delay sensitive services

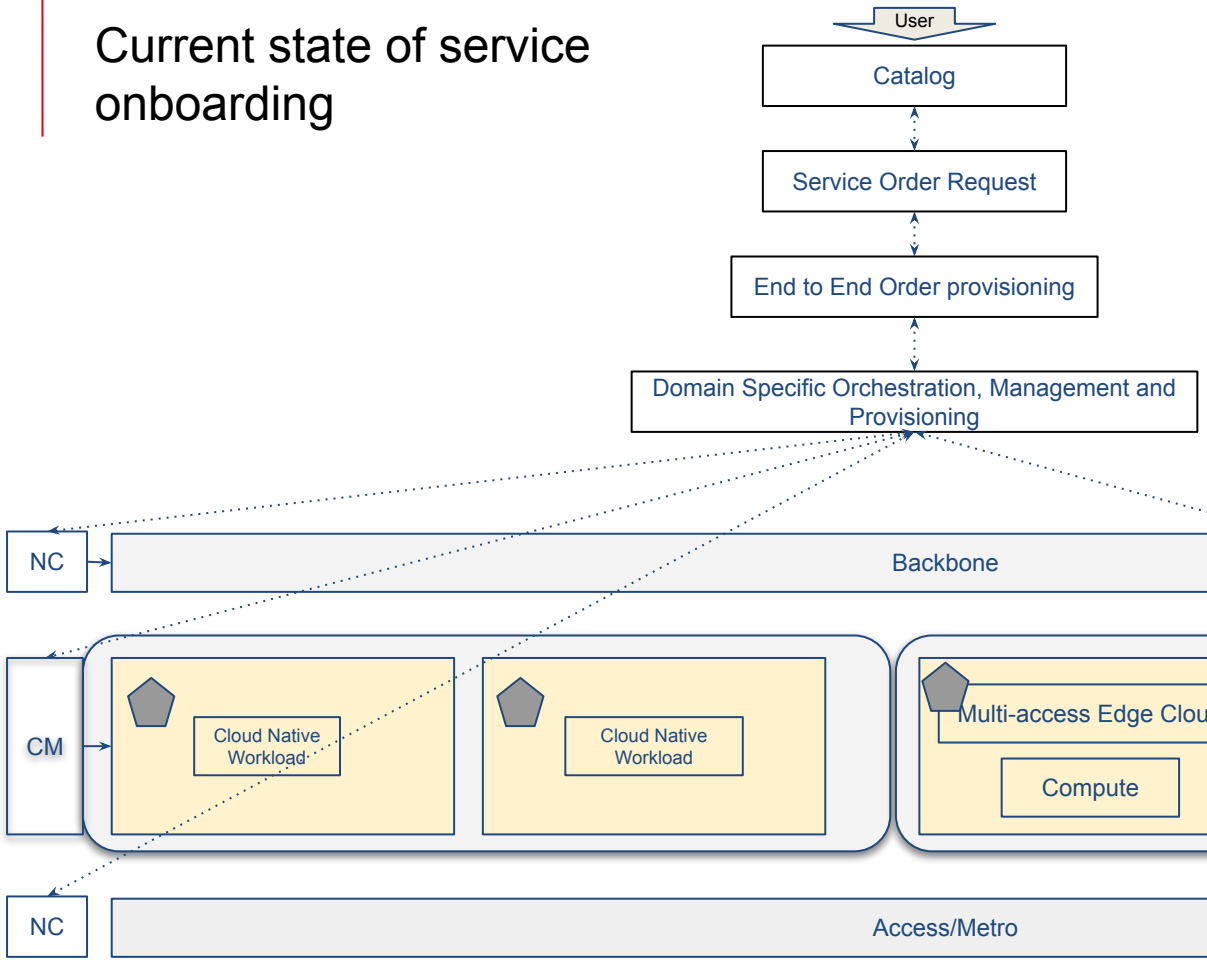
Example services

- AI
  - Online inference
  - Model instantiated close to the data
    - Facial/voice recognition at security checkpoints or building access (offices, malls etc)
      - Traffic patterns vary during the day and week
        - Office hours, weekends
    - Latency sensitive, interactive
    - Dynamic scaling based on traffic volume
- UPF
  - Scheduled and unscheduled traffic increase
  - Dynamic scaling at ports, docks, event venues
- Caching on-demand (content, scheduled/event based)

Common theme is dynamic service instantiation and life cycle management


User Service Request
Service order and fulfillment system
NSO
CM
Kubernetes,
OpenStack,
NC
Core DCs
Edge DCs
Client
Workload
Backbone Network
Access Network
NSO - Network Service Orchestrator
NC - Network Controller
CM - Cloud Manager
Red Hat

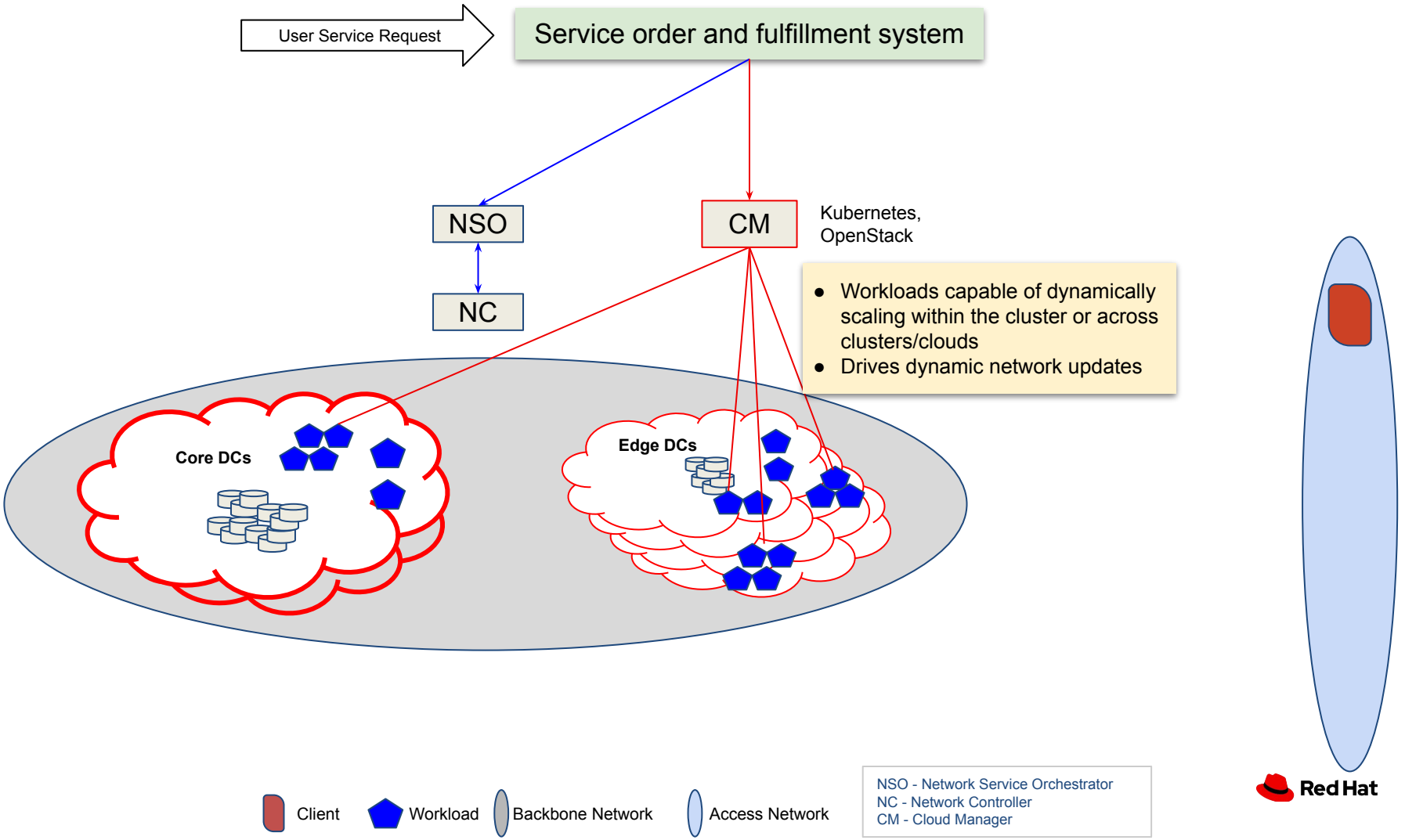
Current state of service onboarding

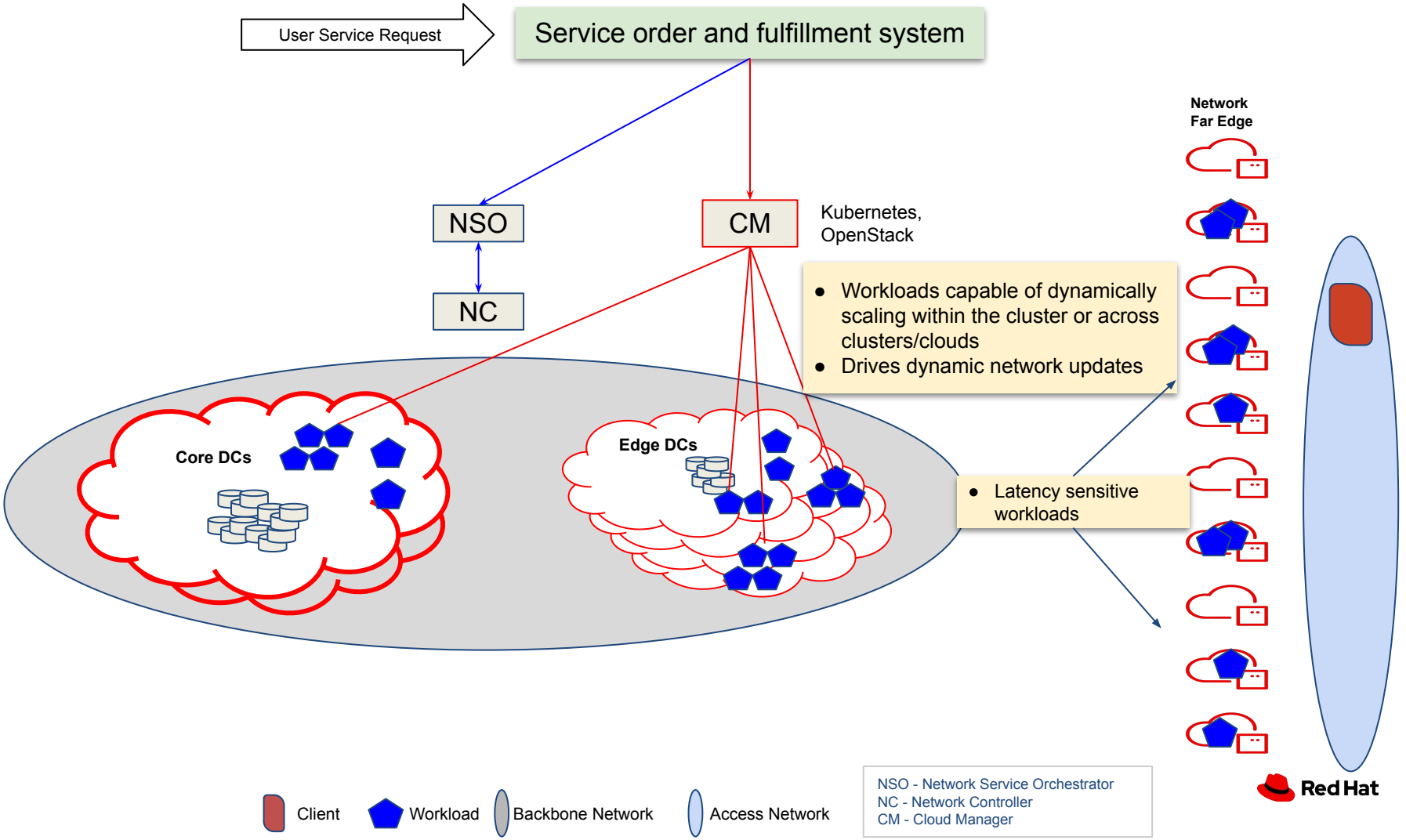


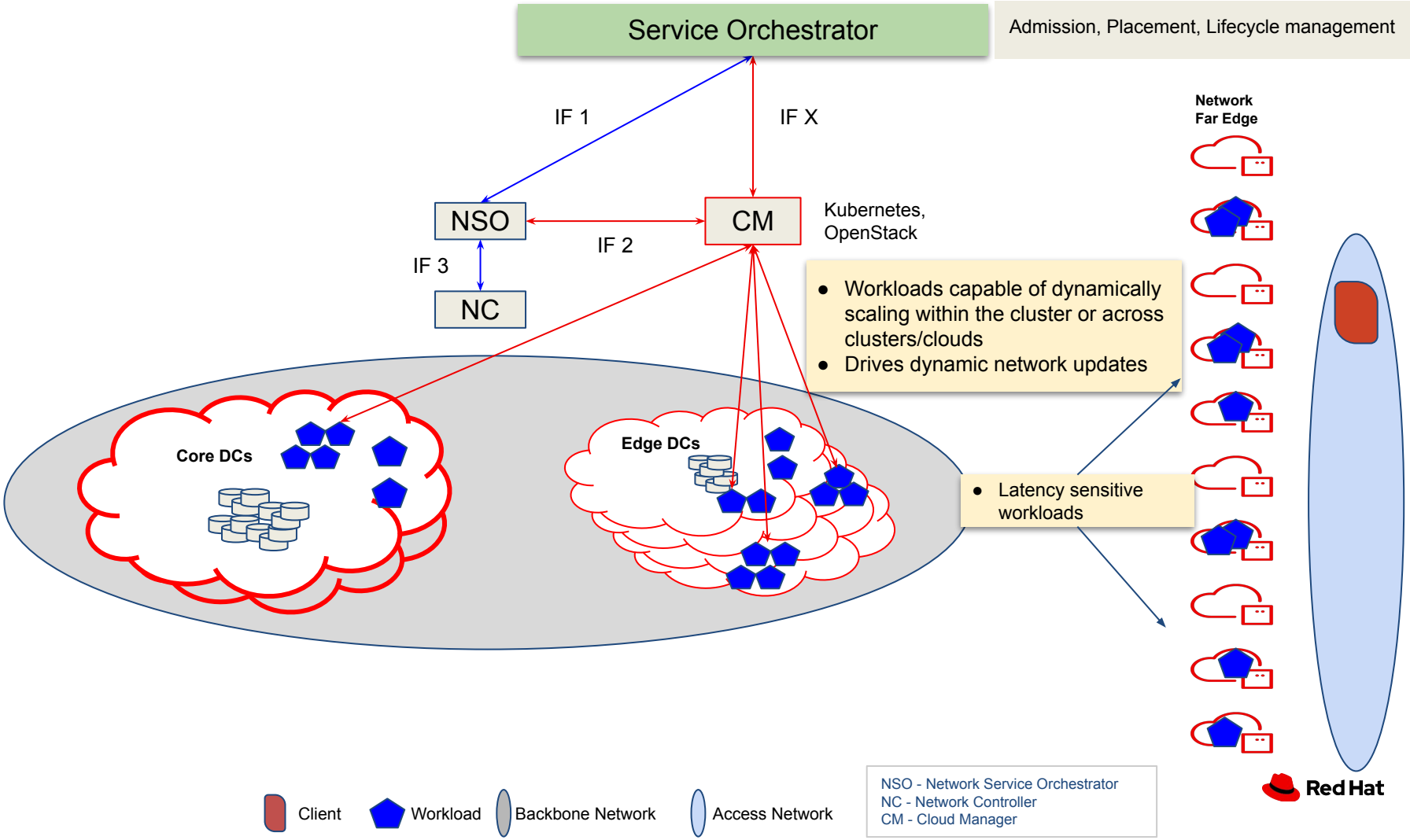
1. Service request placed by user via ordering/catalog system
2. Service instantiation location is based on input from user (zip code)
3. Order acceptance only after ensuring resource availability to meet SLA/SLO at specific location:
   a. Compute
   b. Network
   c. ***This involves querying various systems to determine if resources are available (adds delay to service instantiation) - Multiple vendors, systems***
4. Provisioning
   a. Network - via NSO
   b. Compute - via CM
5. Service/workload instantiation

NOTE:
1. Workloads/services were fairly static

Red Hat


User Service Request
Service order and fulfillment system
NSO
NC
CM
Kubernetes,
OpenStack
Workloads capable of dynamically scaling within the cluster or across clusters/clouds
Drives dynamic network updates
Core DCs
Edge DCs
Client
Workload
Backbone Network
Access Network
NSO - Network Service Orchestrator
NC - Network Controller
CM - Cloud Manager
Red Hat


User Service Request
Service order and fulfillment system
NSO
NC
CM
Kubernetes,
OpenStack
Network
Far Edge
Workloads capable of dynamically scaling within the cluster or across clusters/clouds
Drives dynamic network updates
Core DCs
Edge DCs
Latency sensitive workloads
Client
Workload
Backbone Network
Access Network
NSO - Network Service Orchestrator
NC - Network Controller
CM - Cloud Manager
Red Hat

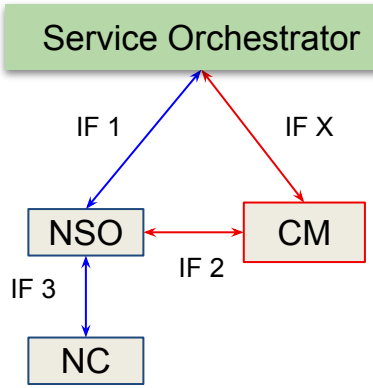

Service Orchestrator
Admission, Placement, Lifecycle management
IF 1
IF X
NSO
CM
Kubernetes,
OpenStack
IF 2
IF 3
NC
Network
Far Edge
Workloads capable of dynamically scaling within the cluster or across clusters/clouds
Drives dynamic network updates
Core DCs
Edge DCs
Latency sensitive workloads
Client
Workload
Backbone Network
Access Network
NSO - Network Service Orchestrator
NC - Network Controller
CM - Cloud Manager
Red Hat

Telco cloud manager and network service orchestrator

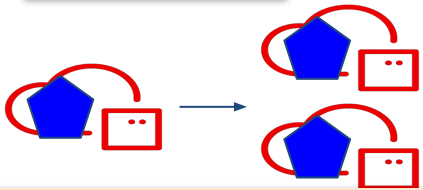
- Cloud manager
  - Life cycle management of workload/service
    - Scheduling and placement of service/workload on specific cloud/cluster and/or node
    - Pool of resources that are allocatable for workloads
    - Placement and scheduling based on workload resource requirements
- Workload can scale upto limit/quota (example trigger: CPU, queue occupancy)
  - Cloud:
    - Service scaled up e.g., replicas/deployment/stateful set scaled up OR service instantiated on a different cluster/cloud
      - NOTE: currently the placement of the workload does not take into account network path metrics but rather compute metrics only
  - Network:
    - Scale up bandwidth, instantiate additional path with same network requirements, updates to DNS, network load balancer, FW etc

IF 2, IF X

Scale up on the same cluster
>>> **IF 2 increase in path bandwidth**
>>> **IF X resource update to service orchestrator**



IF 1, IF 2, IF X



Scale up on a different cluster
>>> **IF 2/IF 1 Add path, network load balancer, DNS, FW etc**
>>> **IF X instantiate service, resource updates**

Service Orchestrator to CM and NSO

- Orchestrator determines placement based on:
  - Service requirements, available cloud capabilities, etc
    - Can the workload/service be instantiated at the required SLA/SLO without impacting existing services?
      - ***Requires current network resource availability information (based on metrics/telemetry or API)***
        - ***Cloud vs node***
      - ***Requires current compute resource availability information (based on metrics/telemetry or API)***
        - ***Cloud vs node especially when it comes to NUMA awareness***
      - ***Note: this is location dependant information***
    - Business logic:
      - Pricing
      - priority vs other workload/customer workloads for preemption
      - If there is a mix of GPUs, prioritize using GPUs with a higher performance delivered per unit of power
      - Green energy
  - If admissible
    - Trigger workflow:
      - NSO to provision path/network service
      - CM to reserve compute resources, instantiate service, LCM of application
- Orchestrator enters the lifecycle management state for the service

IF1

IF X

Red Hat

What is the ask?

- Definition of network resources that can be exposed to the CM to facilitate more optimized workload/service placement/scheduling
  - How are these exposed/consumed (API?)
- Definition of cloud resources to be exposed to the network orchestrator

- The above requires working through specific use cases to get a better understanding on the specifics
  - Complete life cycle of a service

- Support the initiative and will actively participate