

# Compute First Networking (CFN) dyncast Scenarios and Requirements

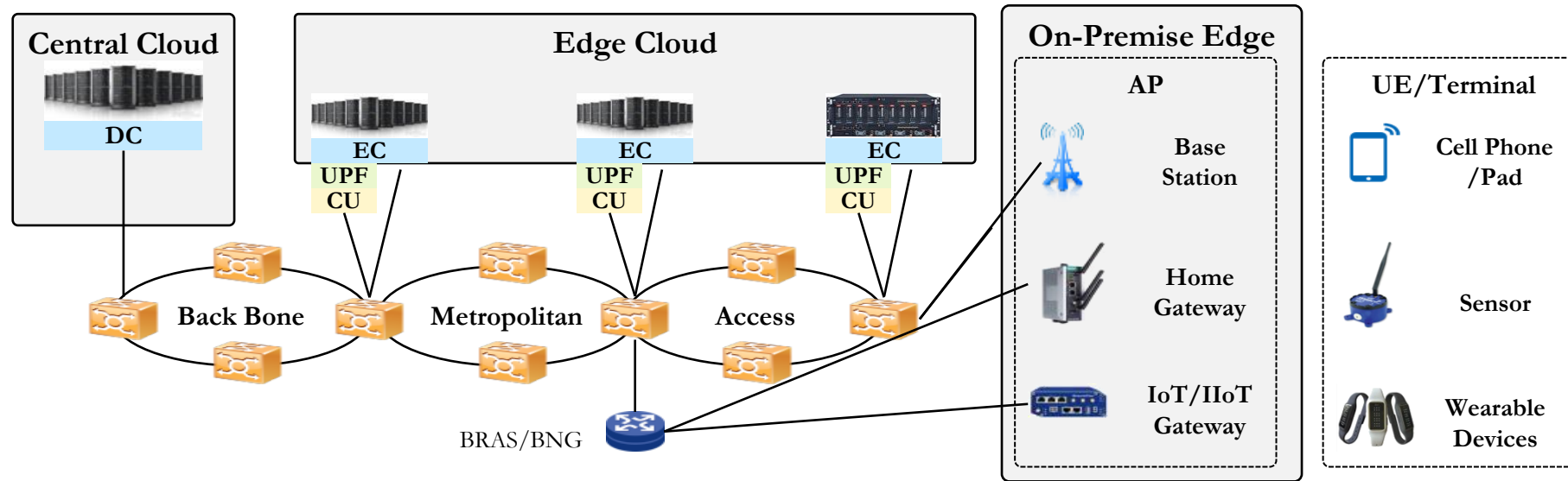
draft-geng-rtgwg-cfn-dyncast-ps-usecase-00

P. Liu, China Mobile

L. Geng, China Mobile

P. Willis, BT

# ICT Infrastructure Redefinition

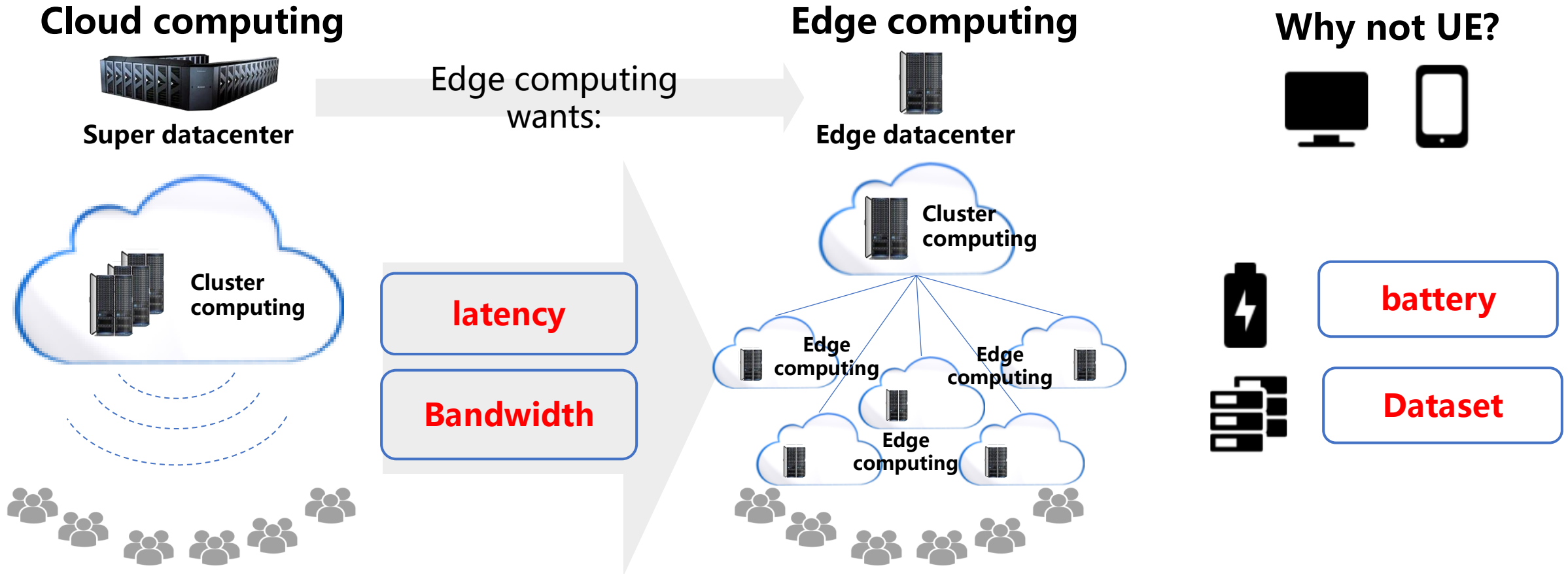


## Facts in China Mobile

- CDN nodes in every city (**330+**) and major county (**250+**), with **25000+** servers installed
  - These nodes can be upgrade to vCDN and then edge computing infrastructure*
  - More diverse computing resource need to be provided ;*
- More edge computing nodes will be setup in an on-demand manner
  - County aggregation **6000+**, Access aggregation **10,000+**, On-site **100,000+**

Service providers are offering the integrated computing and networking infrastructure.

# Why edge computing?



# General Challenges of Edge Computing

- Resource Limitation
  - *fewer servers – 10s of server per node.*
- Heterogeneous Hardware
  - *CPU, GPU, Memory, ASICs*
- Dynamic Load
  - *Available resources change quickly*
- Edge-cloud Coordination
  - *Edge does not solve all*
- High Cost
  - *On-site maintenance is expensive*
- Mission Critical
  - *Users are counting on you (i.e. 100% reliability of industry automation)!*

Many of this challenges are **NOT** solvable solely in “Computing Domain”.

*Nearest but not the best.*  
How could the “Network Domain” Help?

# Requirements

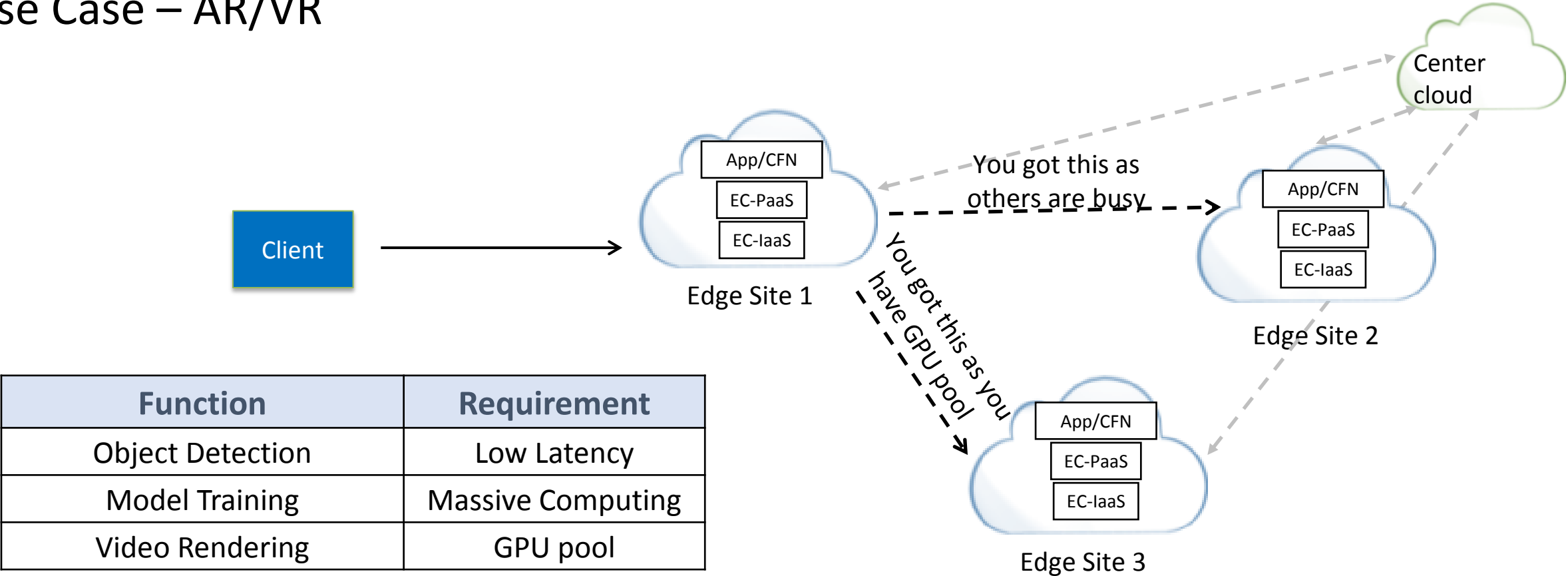
- **Providing Functional Equivalency**

- Same level of user experience no matter where you are and which edge sites you are connected

- **Providing Service Dynamics**

- Traffics are diverted/steered to preferred edge sites according to infrastructure status and user SLA requirements

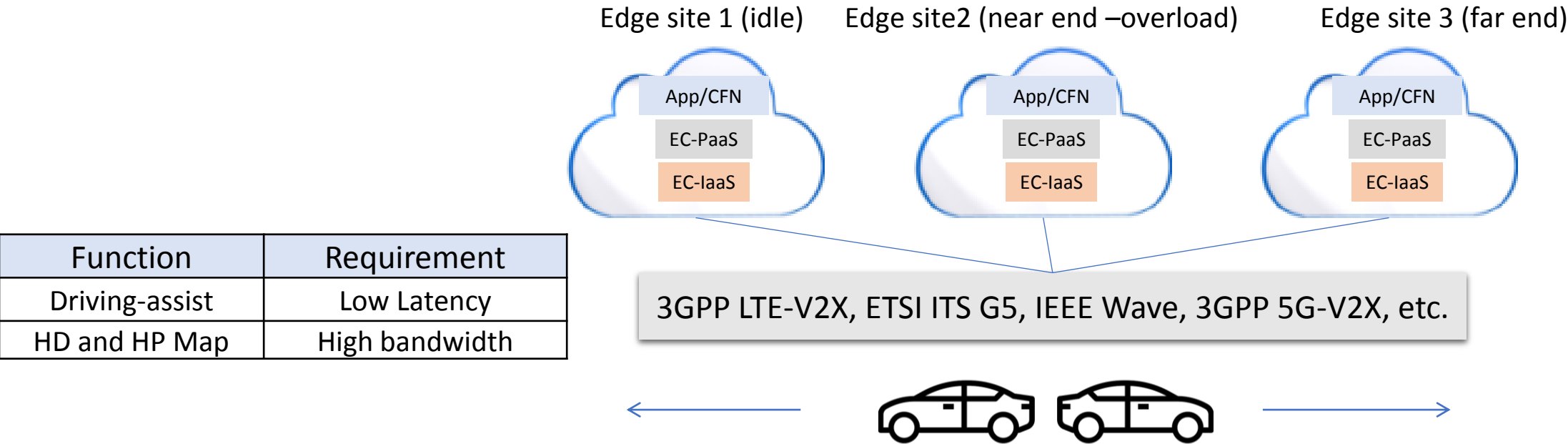
# Use Case – AR/VR



## Applying CFN-dyncast in AR/VR use cases

- Training in center cloud, whilst detection in edge DC
- Rendering tasks need to be diverted to GPU infrastructure
- Traffic/compute offloading for tide effect (Theatre/Sport stadium cases)

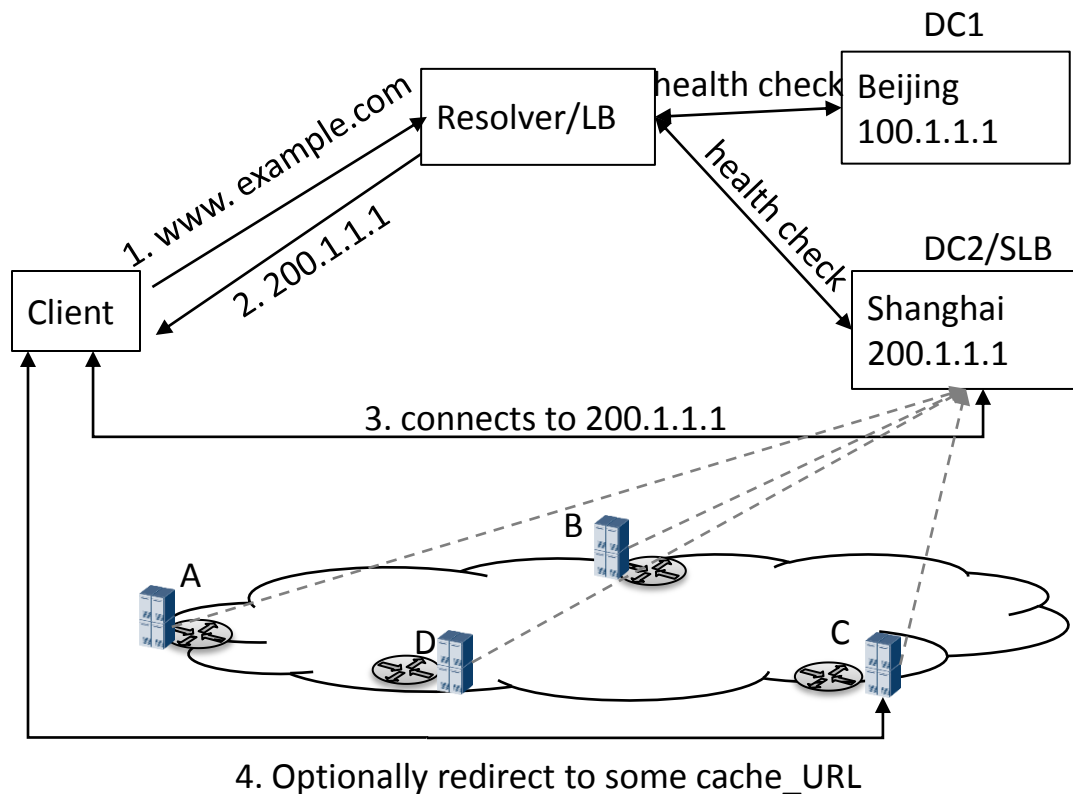
# Use Case – Connected Car



## Applying CFN-dyncast in Connected Car Use Cases

- Mission critical traffic is diverted to the closest sites
- Non-real-time traffic diverted to the cloud (Entertainment, Traffic status etc. )
- Protection and fast service requirement in the case of edge site failure

# Current Practices, considerations and gaps - efficiency and latency



- Use geographical location, pick closest
  - Edges are not so far apart. Locations do not matter most.
- Health check in an infrequent base ( $>1s$ ), switch when fail-over
  - Limited computing resources on edge, change rapidly ( $<1s$ )
- Random or round robin pick, network cost is not a concern or updated infrequently just to keepalive
  - Edges are not deployed in equal cost way, network status is considered at a later stage not at the same time
- Centralized determination, good for content retrieval.
  - Not be as good as for computation which has more dynamic nature and larger number
- Early binding: clients query first and then steer traffic.
  - Edge computing flow can be short. Early binding has high overhead.
- Caching at the client.
  - Stale info could be used.
- Others:
  - Network based solution uses least network cost, computing load is not considered
  - Traditional anycast bases on single request/reply packet, no flow affinity



# Proposed CFN-dyncast Features to solve the gaps

## **1. Anycast based service addressing methodology**

- Anycast makes sure data packet potentially can reach any of the edges
- Mapping of a unique service identifier to specific unicast address

## **2. Flow Affinity**

- Service continuity needs to be handled

## **3. Computing Aware Routing**

- Forwarding nodes is aware of the computing status
- Methods for notification and dimensions of computing resource measurement needs to be studied

# In Summary

- Service providers are offering the integrated computing and networking infrastructure
- Problem: How to optimally route service demands based on computing and network metrics to the best edge?
- Existing IETF protocol specification work does not sufficiently solve the identified problem at the network level
  - **Exposing up-to-date computing resources to the network layer**
  - **Computing and network metrics collection, representation, distribution and how to use them for edge determination**

# Thank you!