

# Elastic Video Content Delivery Networks at the Edge

Tuan Tran, Christoph Neumann, Guillaume Bichot



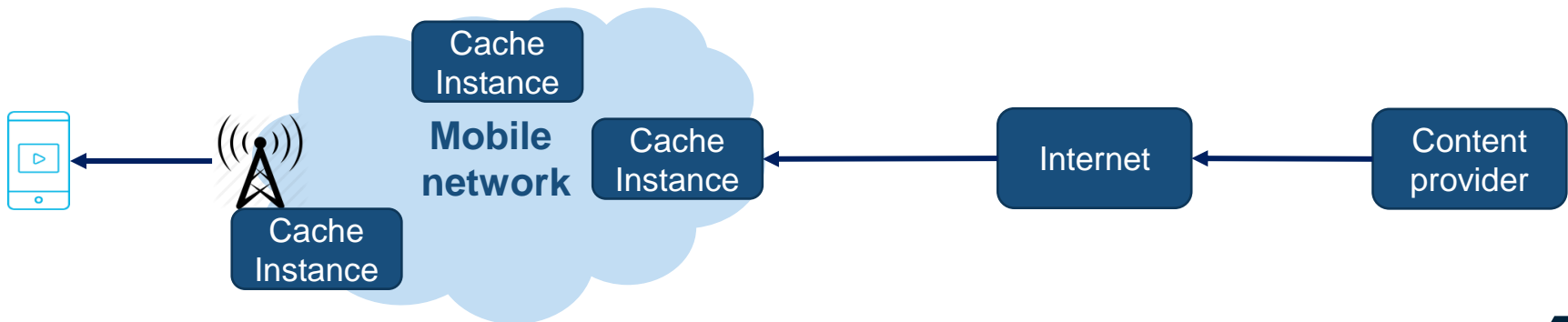
May 10, 2023

# Motivation

Delegating the streaming content to vCDN cache instances deployed at nearby edge locations improves QoE

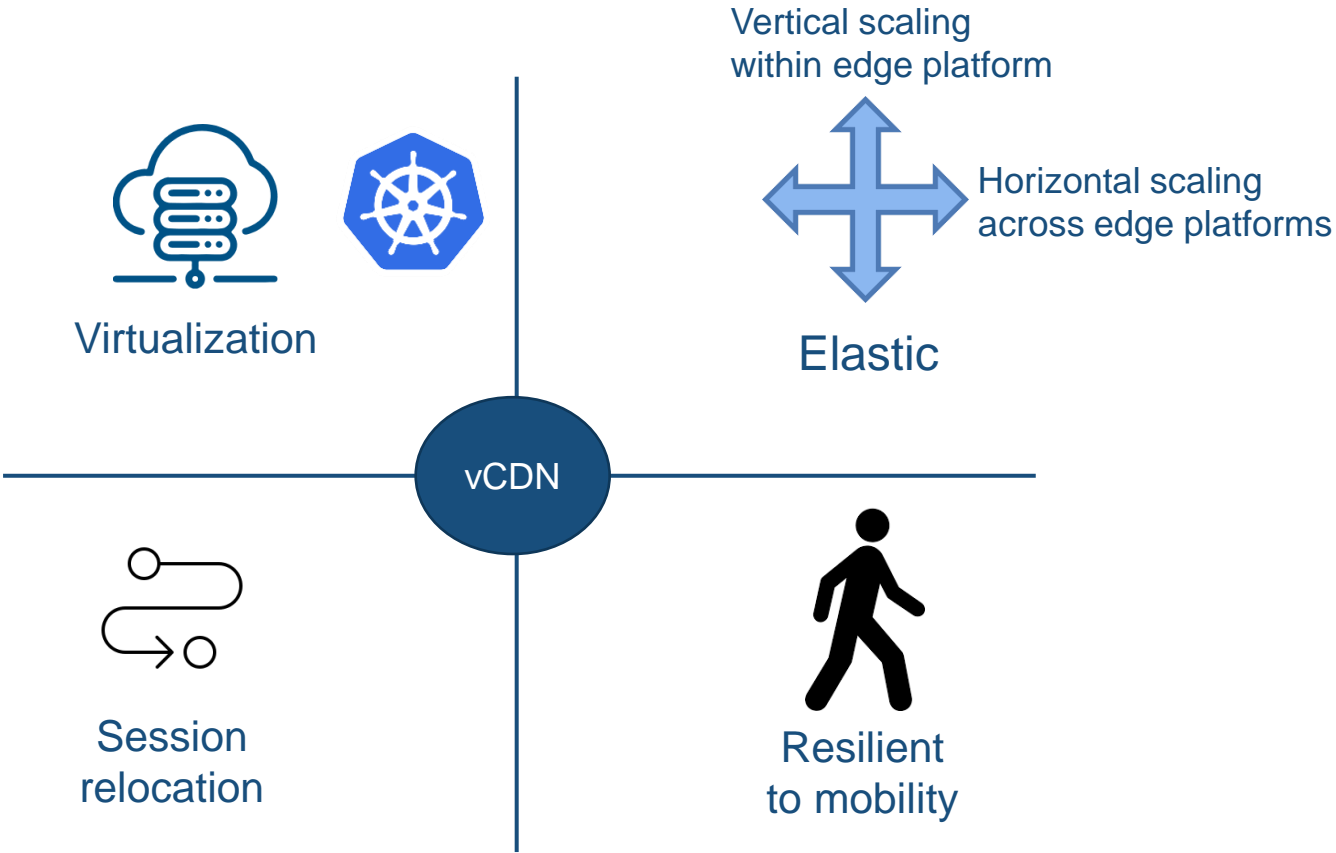
But, continuously running vCDN cache instances at all edge locations is a waste of resources

→ The vCDN must continuously adjust its virtual infrastructure in order to **optimize the trade-off between the cost and the QoE provided to end-users**

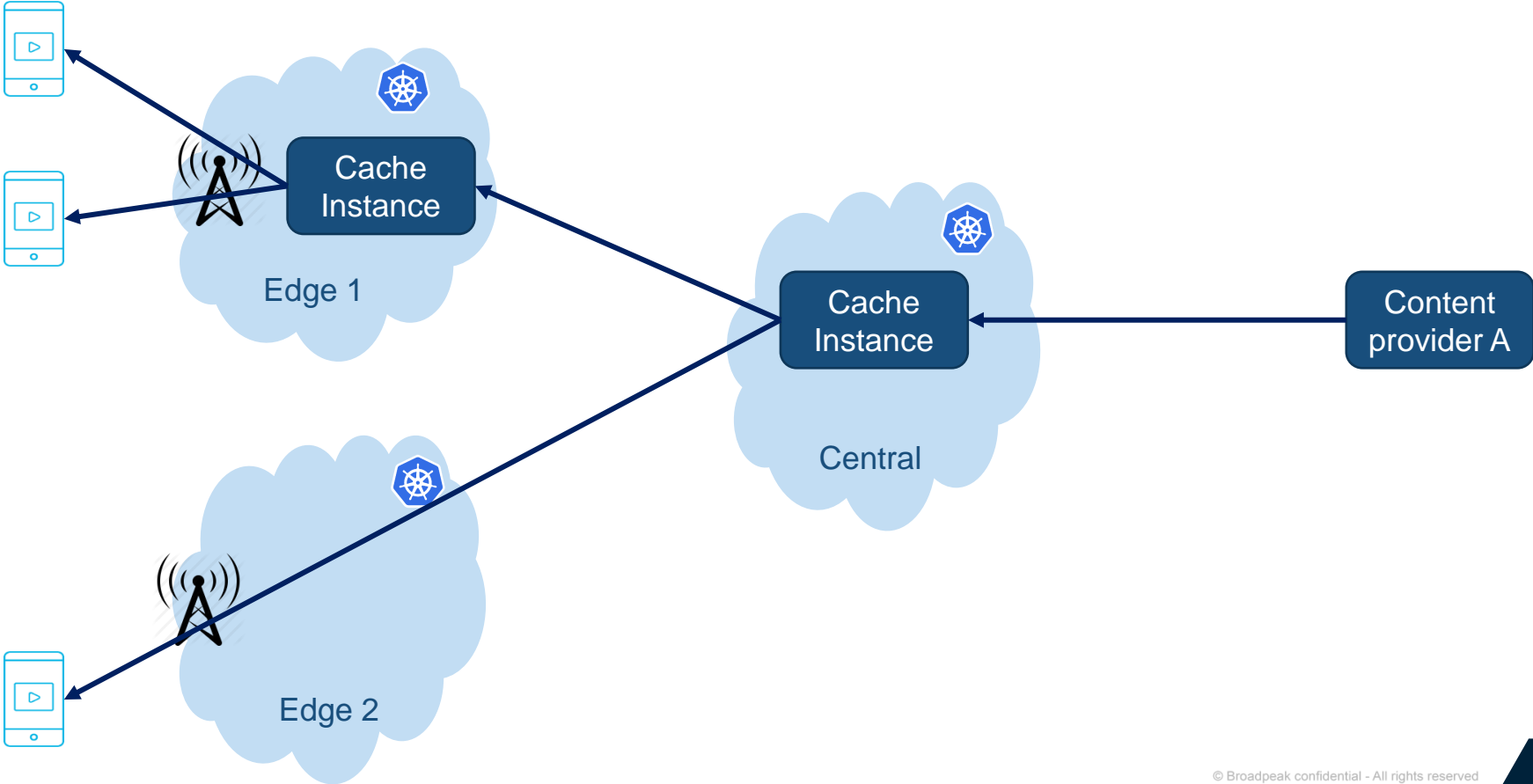


“Edge” means anywhere within the mobile network

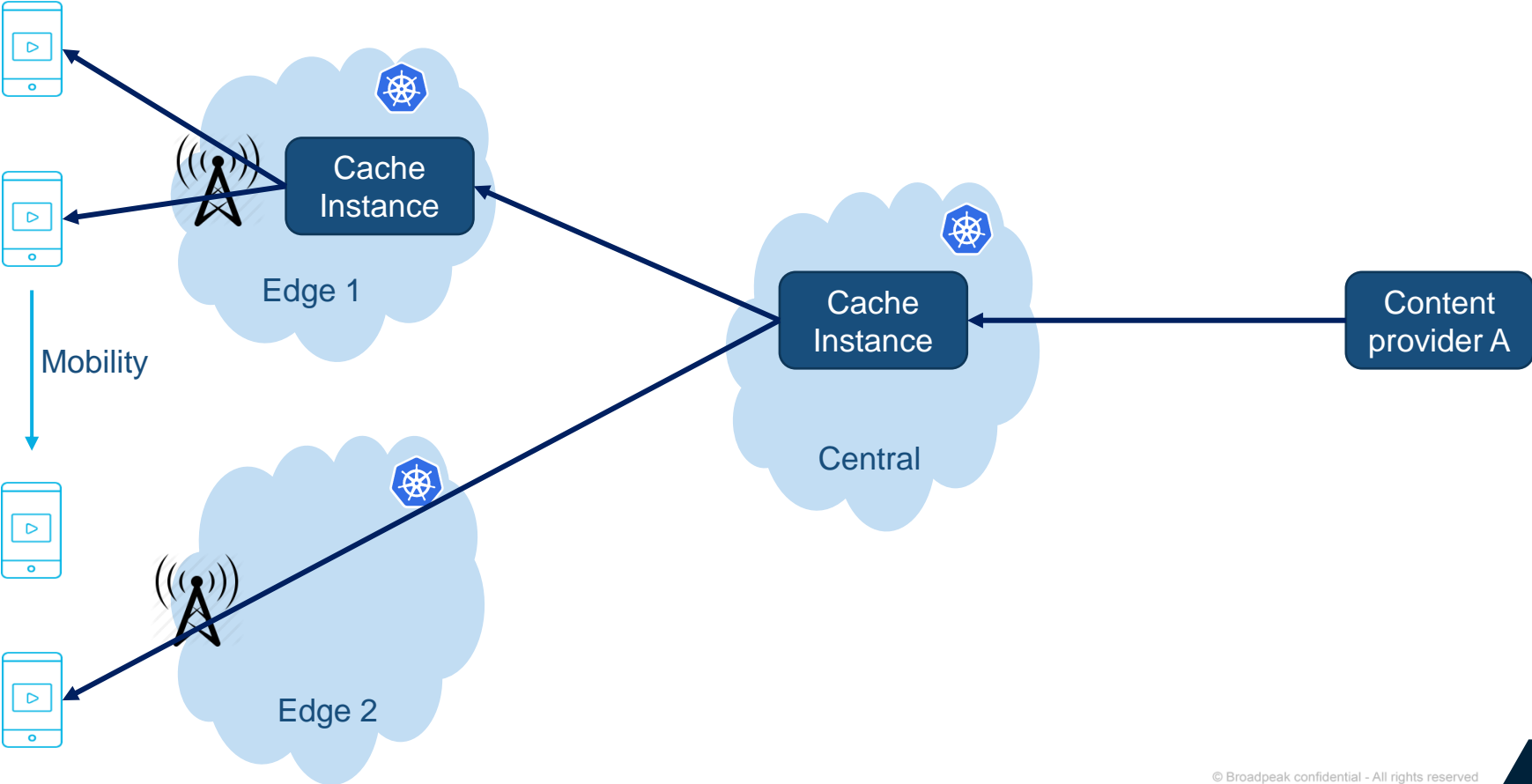
# vCDN system requirements



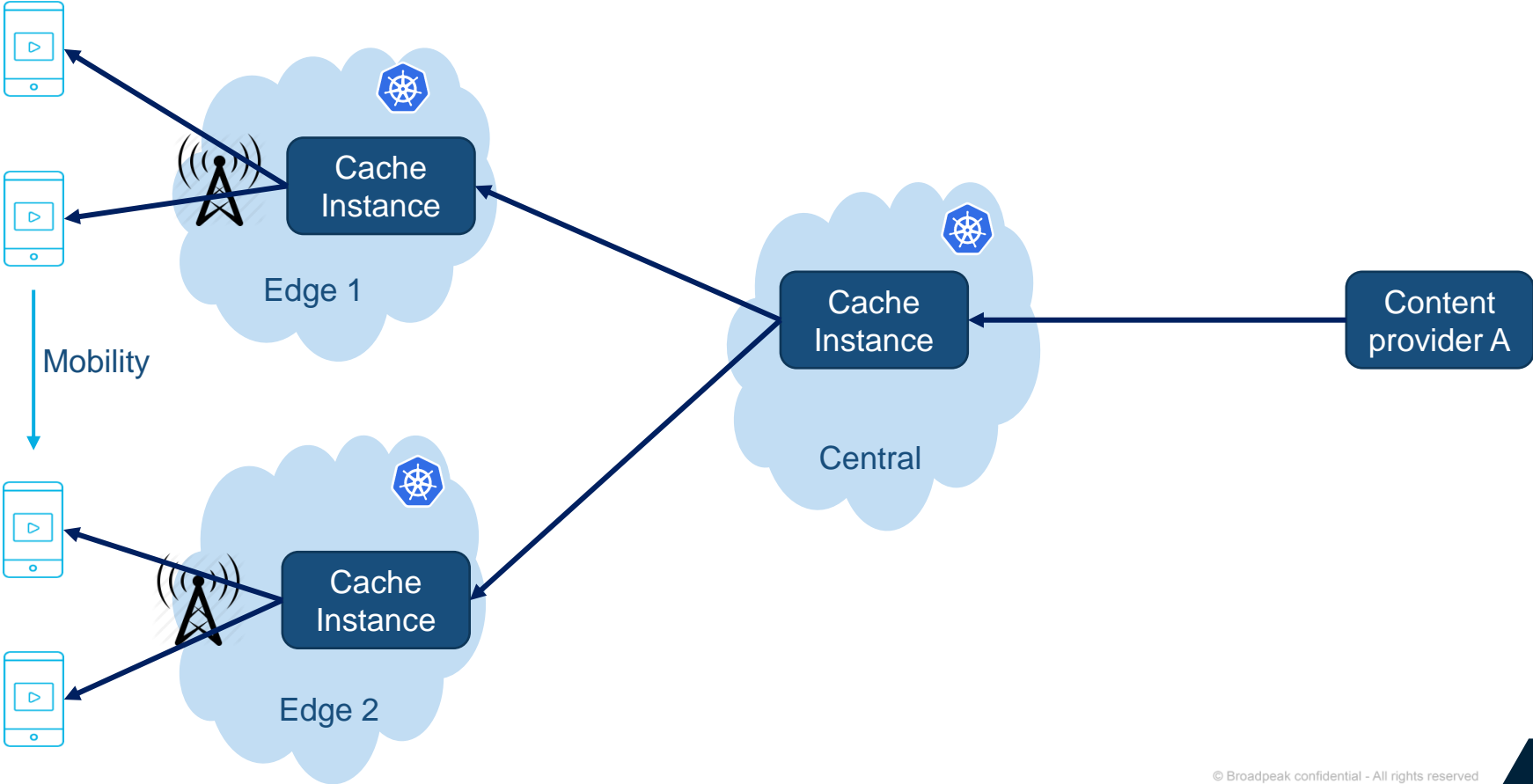
# vCDN concept



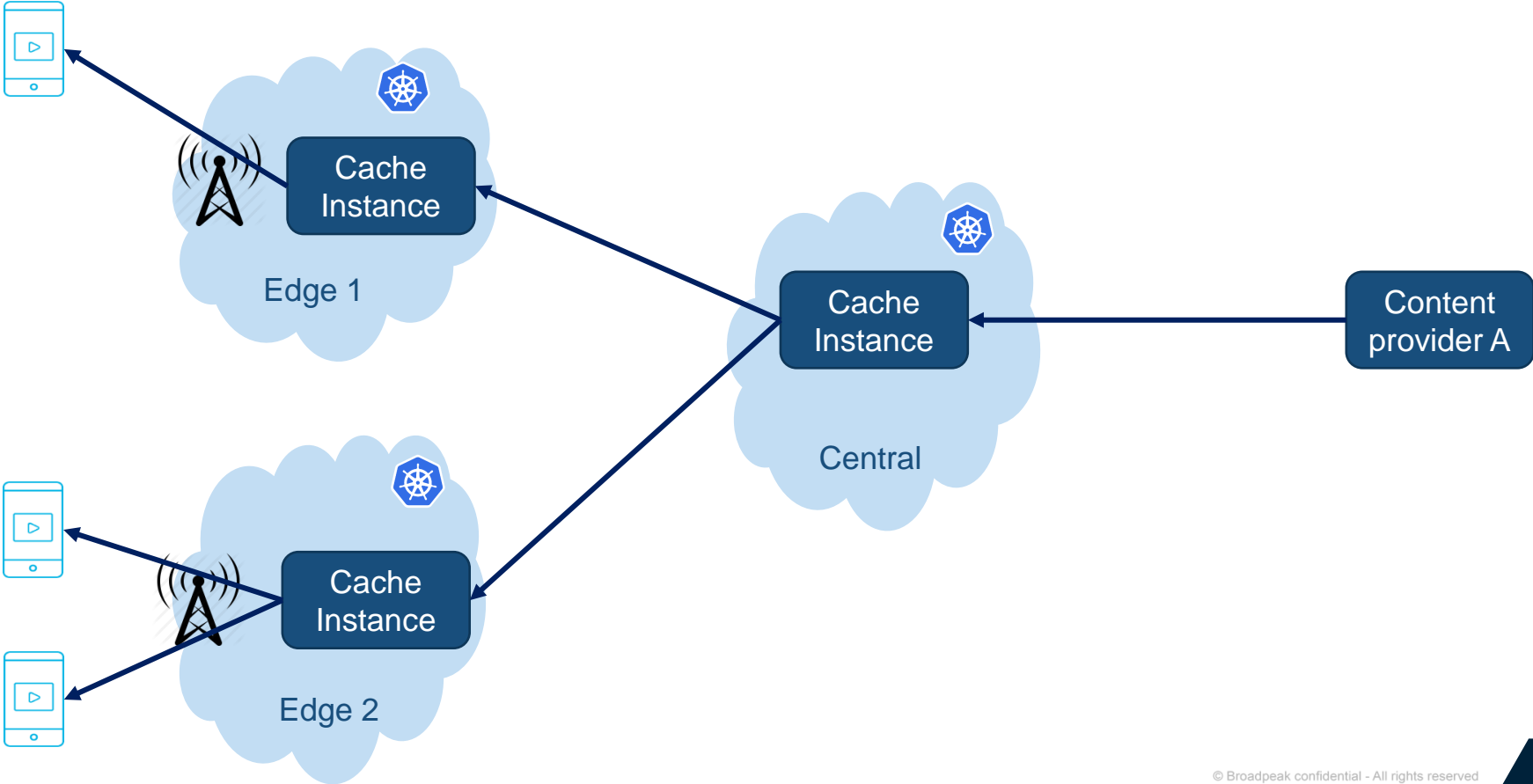
# vCDN concept



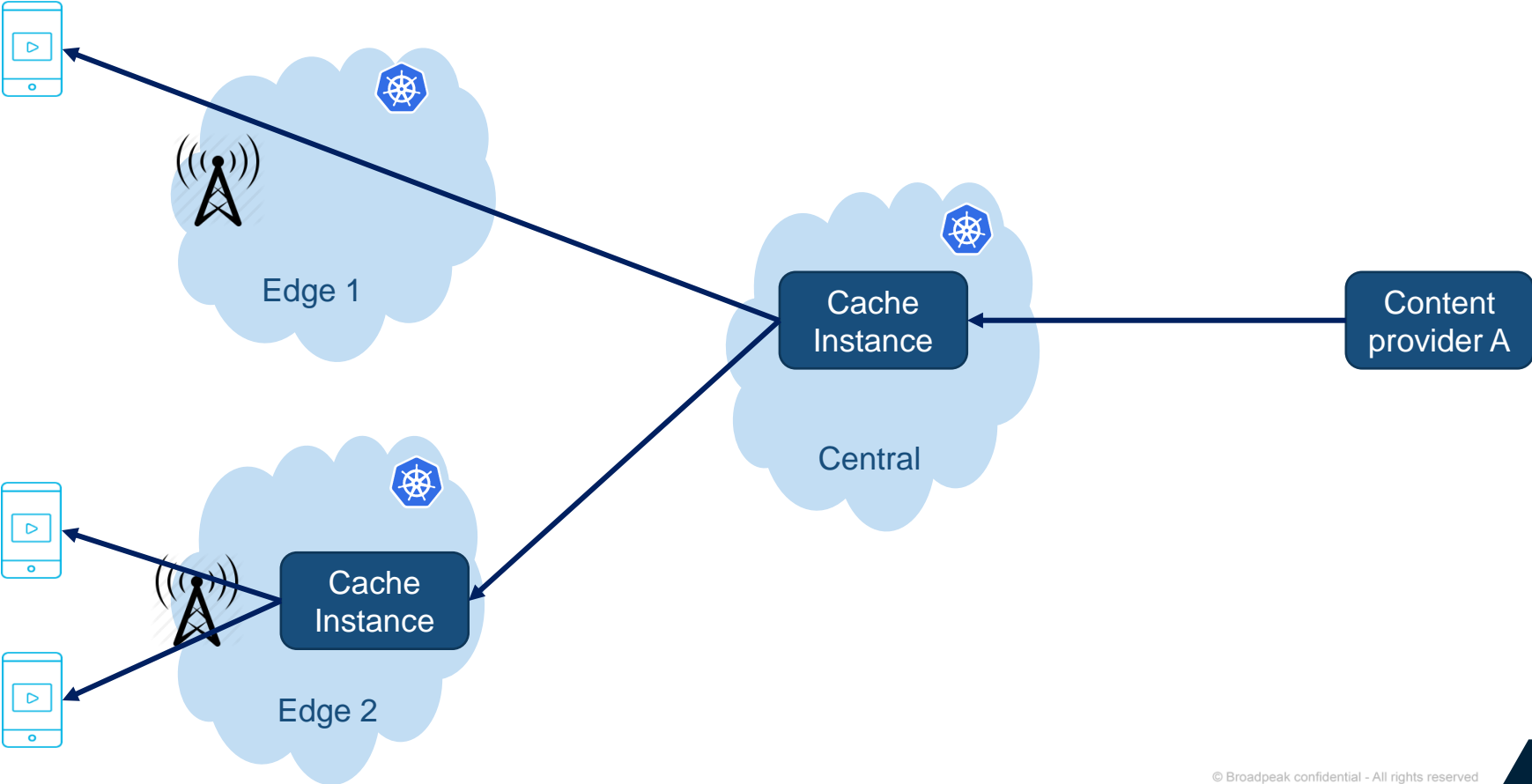
# vCDN concept



# vCDN concept



# vCDN concept





# vCDN architecture on mobile networks

## Mobile network infrastructure

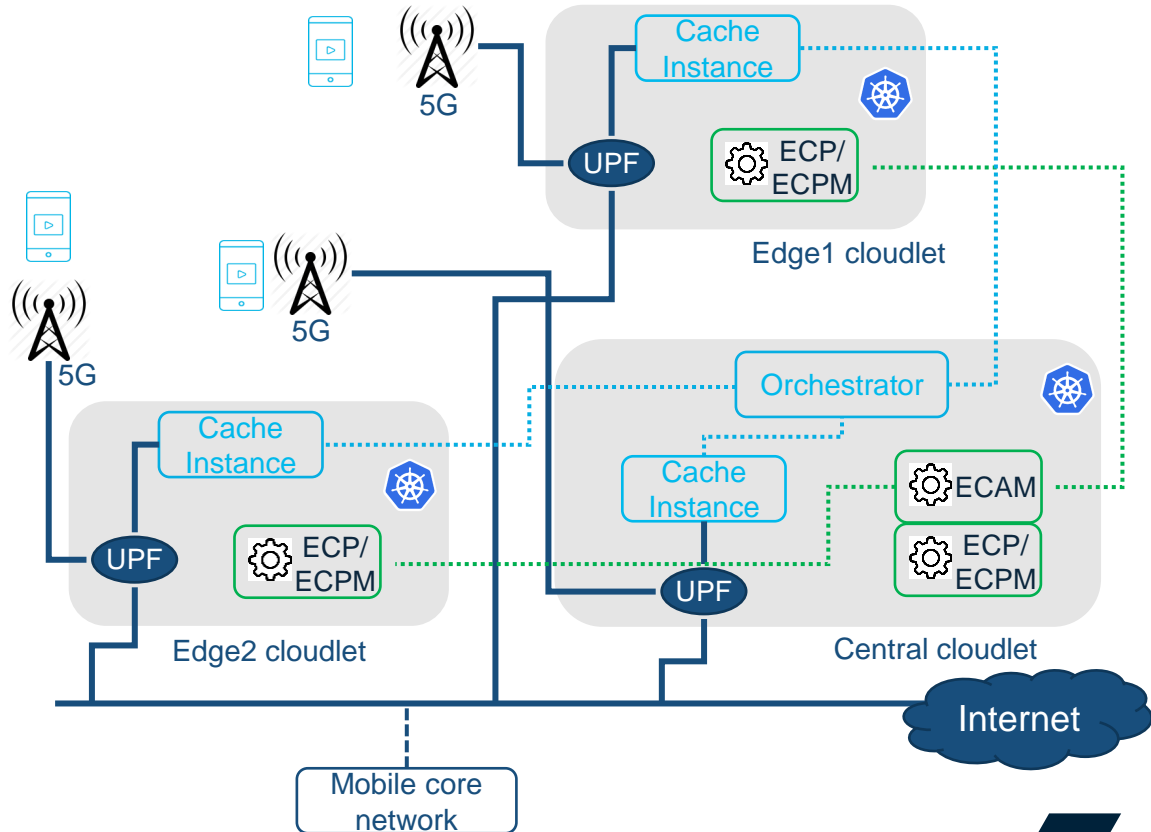
- Radio Access Network (eNB/gNB)
- Core network
- Data gateways (UPF in 5G)

## Edge platform

- ECAM (Edge Cloud Application Manager)
- ECP (Edge Cloud Platform)
- ECPM (Edge Cloud Platform Manager)

## Virtualized CDN infrastructure

- Cache instance = Cache Controller + Cache Server(s)
- Cache Orchestrator



# Cache Orchestrator

## Network discovery



- Available cloudlets
- Cloudlet capabilities, status (e.g., load)

## Cache management



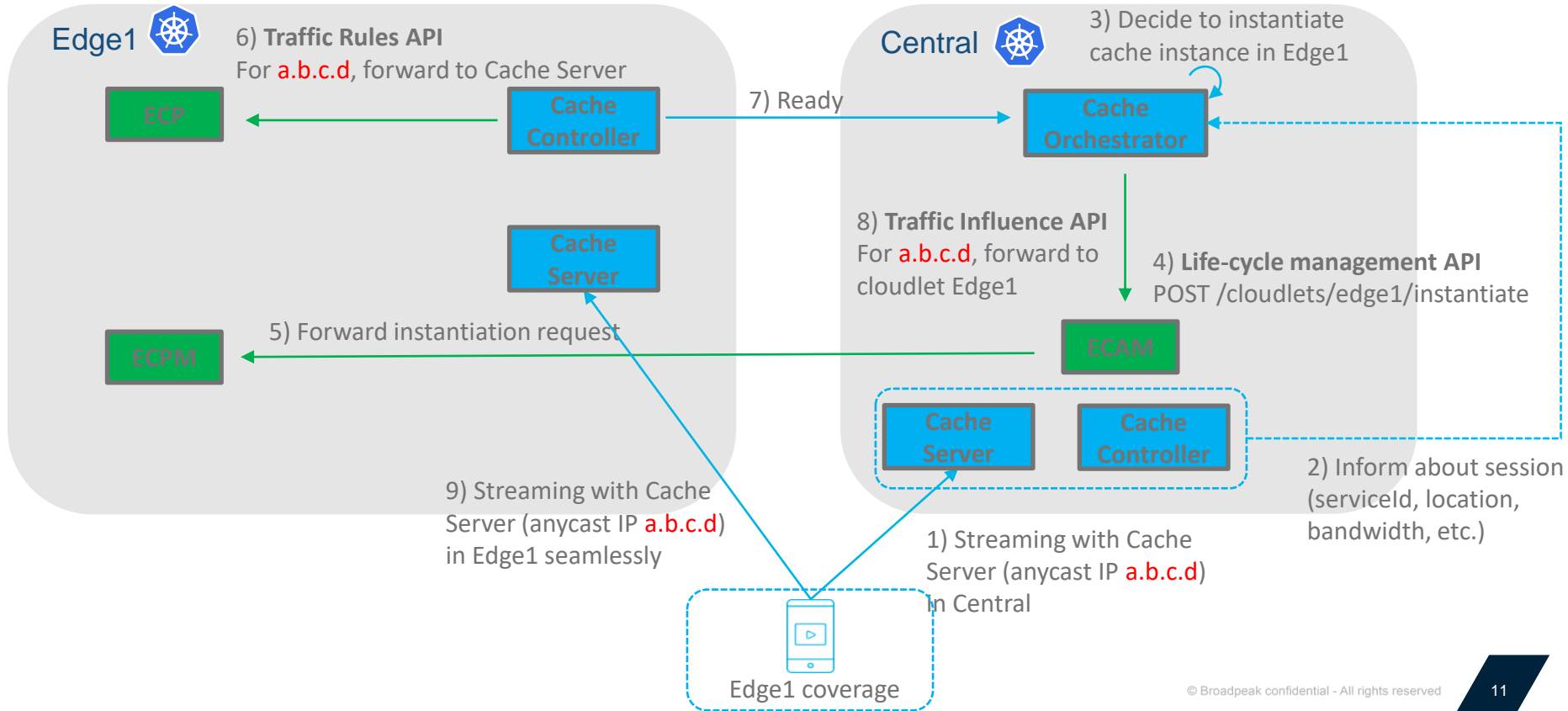
- Cache instantiation/termination based on application metrics within or across cloudlets (e.g., number of sessions, energy consumption)

## Traffic steering

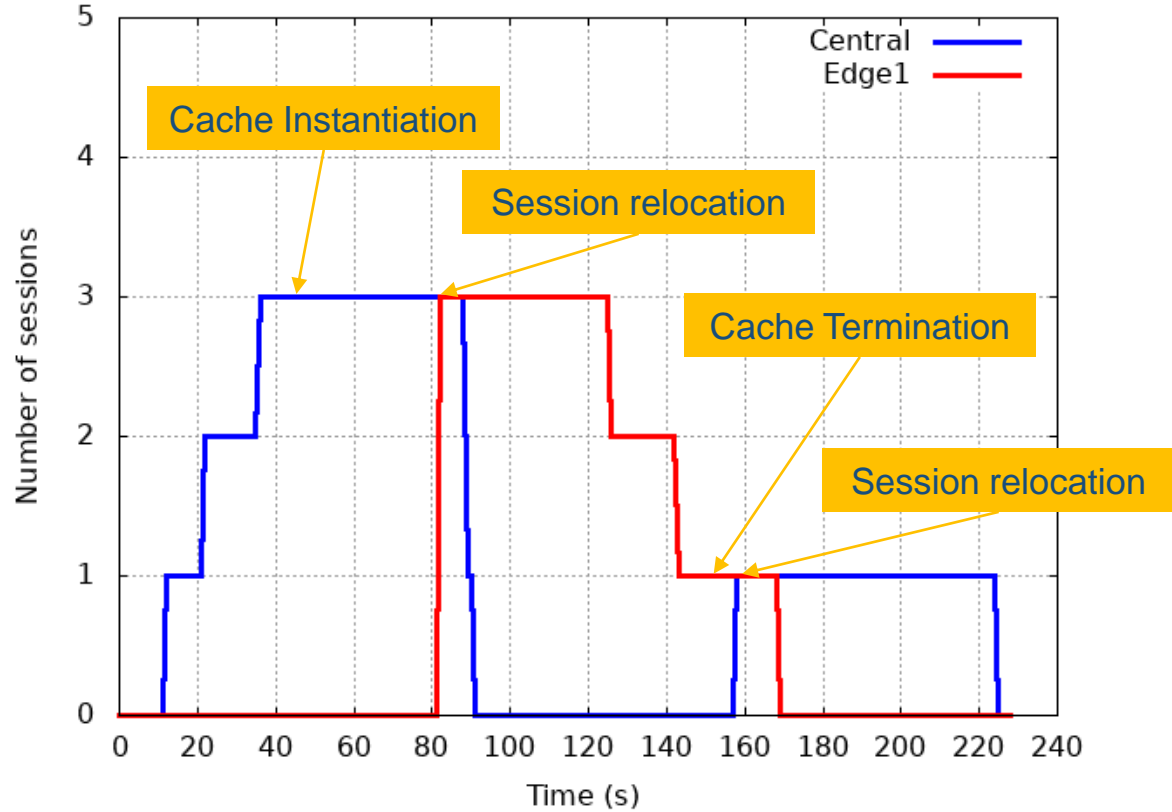


- Traffic Influence
- Traffic Rules
- IP anycast

# Cache instantiation workflow

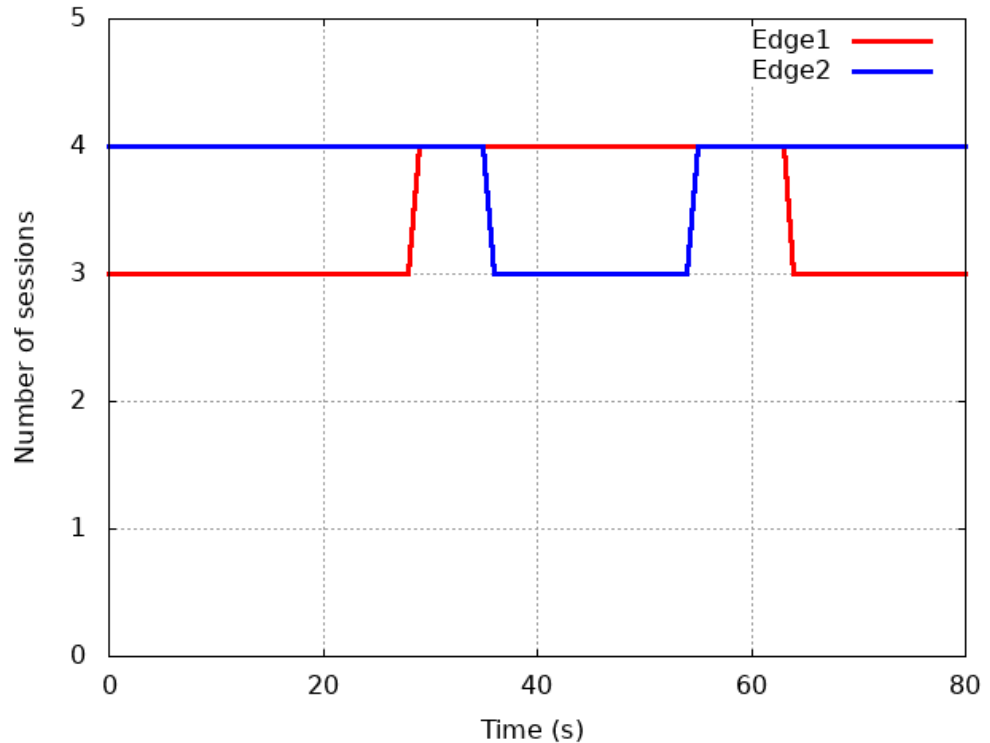


# Experimentations (1/2) – Cache Instantiation/Termination



# Experimentations (2/2) – Handover & dash.js issue

- Packet loss may occur during handover (especially at the end of segment delivery) → may lead to playback stall
- Propose adding a configurable timer to monitor download progress from last event
- GitHub [PR #4126](#) accepted



# Concluding remarks

- Fully virtualized CDN infrastructure deployed on private Kubernetes clusters
- Ported on public cloud
- Applicable to both mobile and fixed broadband networks
- Demo at #C8 booth

The image displays two screenshots related to a virtualized CDN infrastructure.

The left screenshot shows the **BkM100™ CDN Topology** dashboard. It features a tree view of the CDN components and a table of sessions and inputs. The table data is as follows:

Sessions	Bitrate (Mbit/s)
3	1000
7	300

Inputs	Bitrate (Mbit/s)
1	200
1	1000

Additional configuration details include:

- Type: BLS400
- Live management:
- Live server level 1:
- Live server repeater:
- Use BKE live multicast:
- VOD repeater:
- Control URI (IP-Port): 192.168.1.21:30287
- Streaming URI (IP-Port Label (Version)): 192.168.1.21:30291
- Provisioning URI (IP-Port): 192.168.1.21:21
- Server delivery types:  Smooth Streaming,  HLS,  Streaming HTTP,  HDS,  DASH
- Congestion management: Low threshold: 80%, High threshold: 90%

The right screenshot shows a Kubernetes cluster management interface. It displays a list of pods in the 'edge2-bbcc-bcb-cc-5fb4b65d9c-S...' namespace. The pods are:

Name	Restarts	Age	Status
edge2-bbcc-bcb-cc-5fb4b65d9c-S...	0	9s	Running
edge2-bks400-0	0	9s	Running

A text overlay on the interface states: "4 streaming sessions served in Edge 1 area are load balanced".

The **broadpeak** logo is visible at the bottom center of the image.

# Thank you !

[broadpeak.tv](https://broadpeak.tv)



A stylized graphic above the word 'broadpeak' consisting of a solid blue peak on the left and a dotted blue peak on the right, both curving downwards.

**broadpeak**

This is streaming at its peak