CFN-dyncast

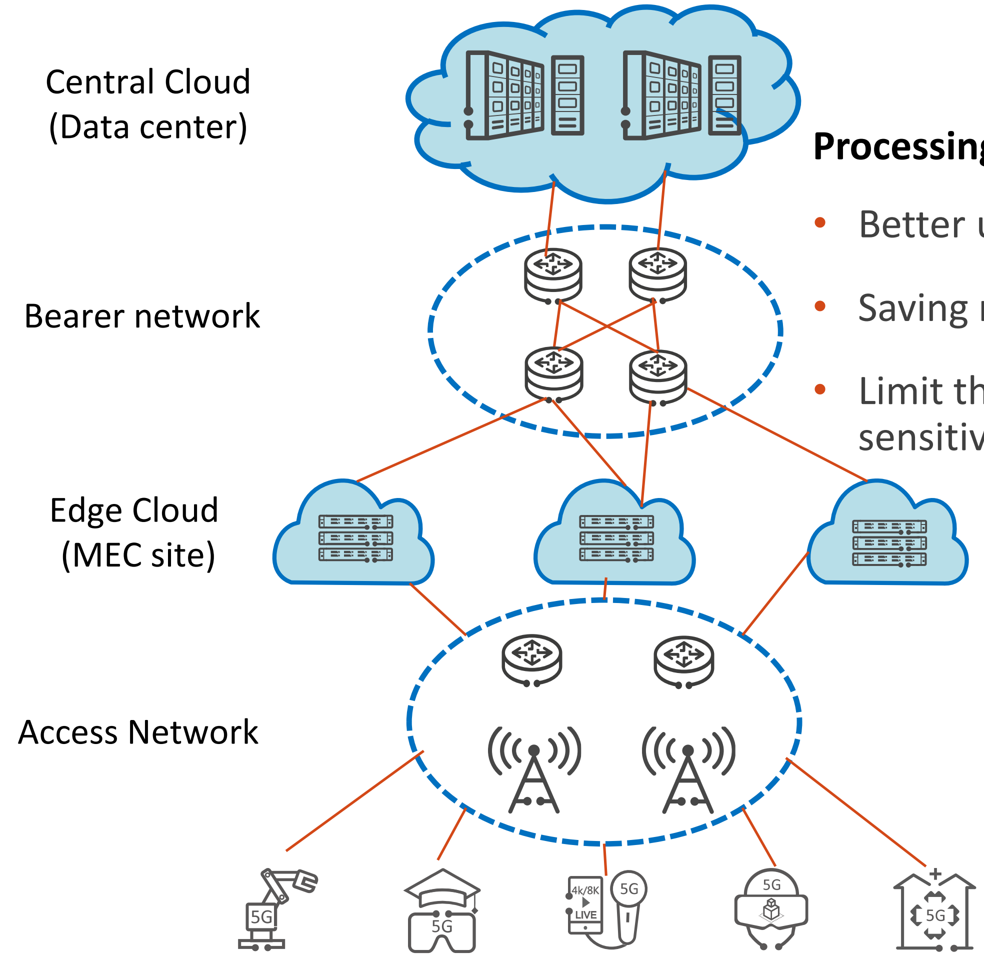
Load Balancing the Edges via the Network

Bing Liu, **Jianwei Mao (speaker)**, Ling Xu, Ruizhao Hu, Xia Chen

2021.03.29

HUAWEI

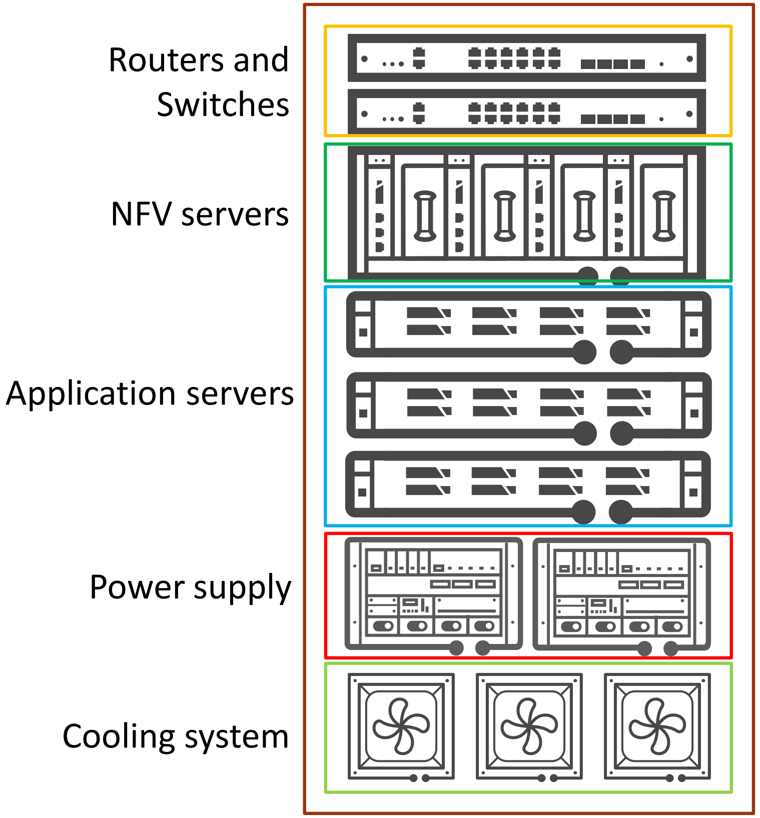
Edge Computing and MEC sites



Processing data at the edge

- Better user experience
- Saving network bandwidth
- Limit the spread domain of sensitive data

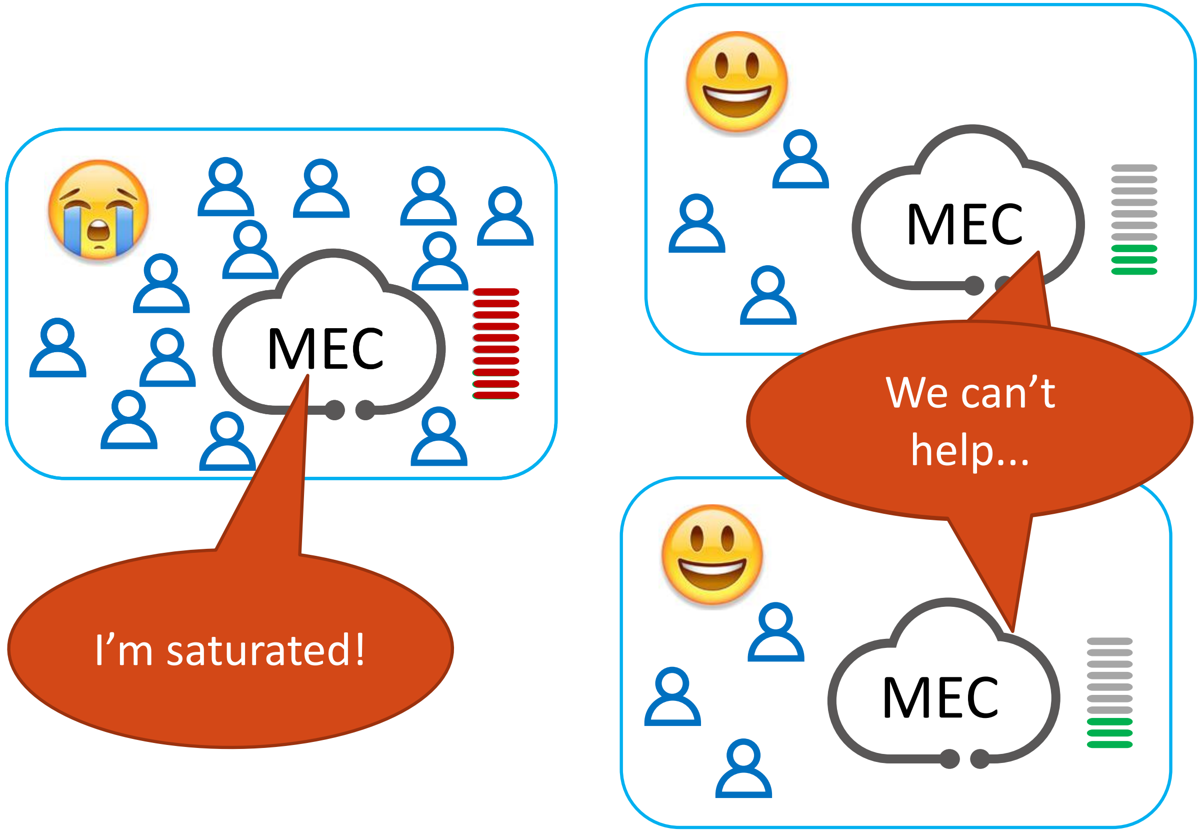
What is in a MEC site?



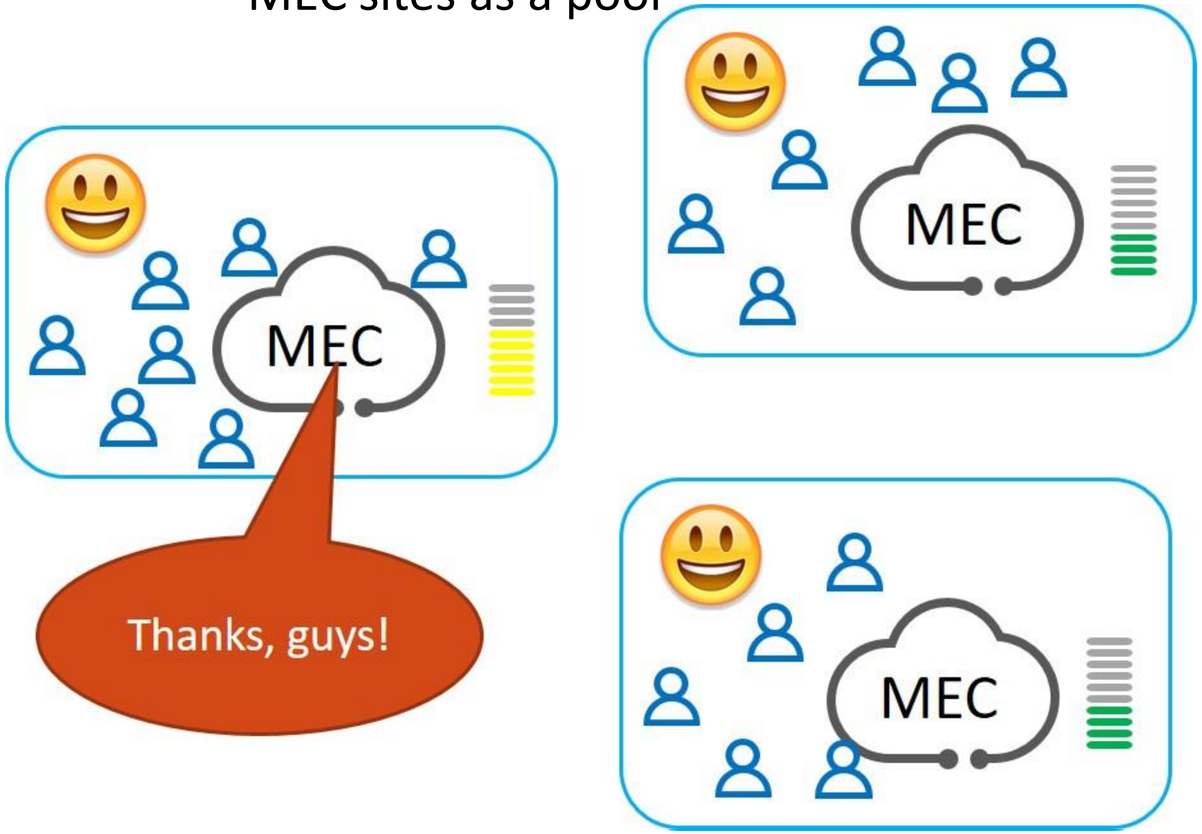
- Limited physical size
- Not easy to scale-out
- Scale-up needs investment

Why Load Balancing is needed for the edges?

MEC sites as silos



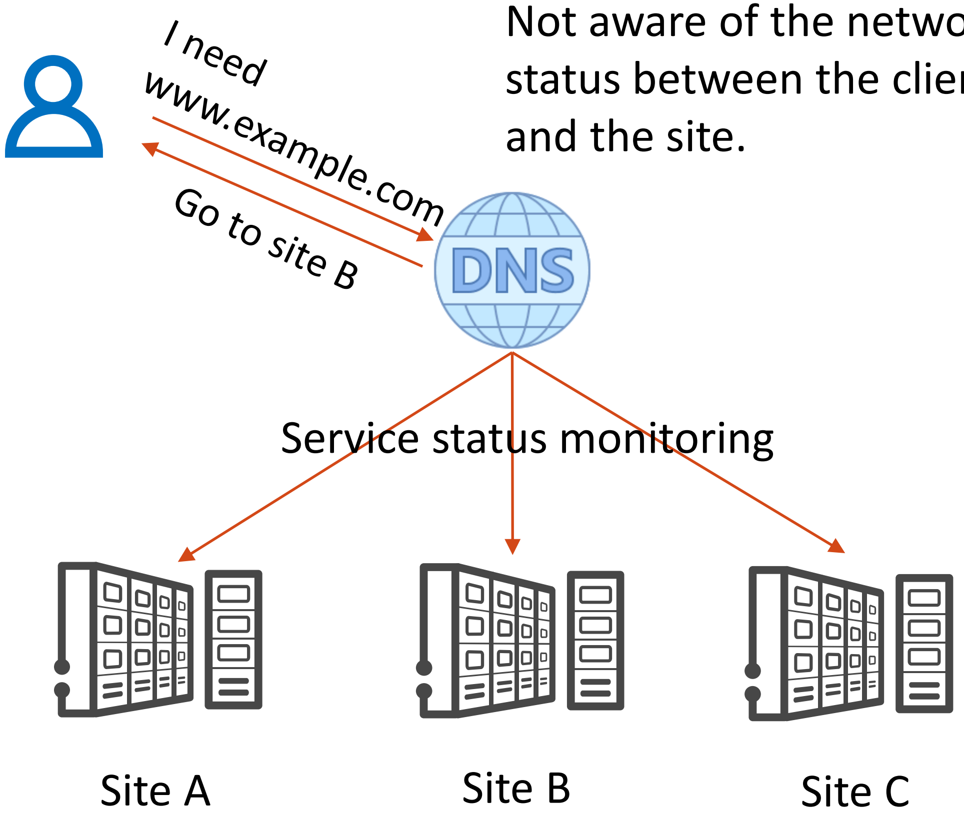
MEC sites as a pool



Some Current Methods

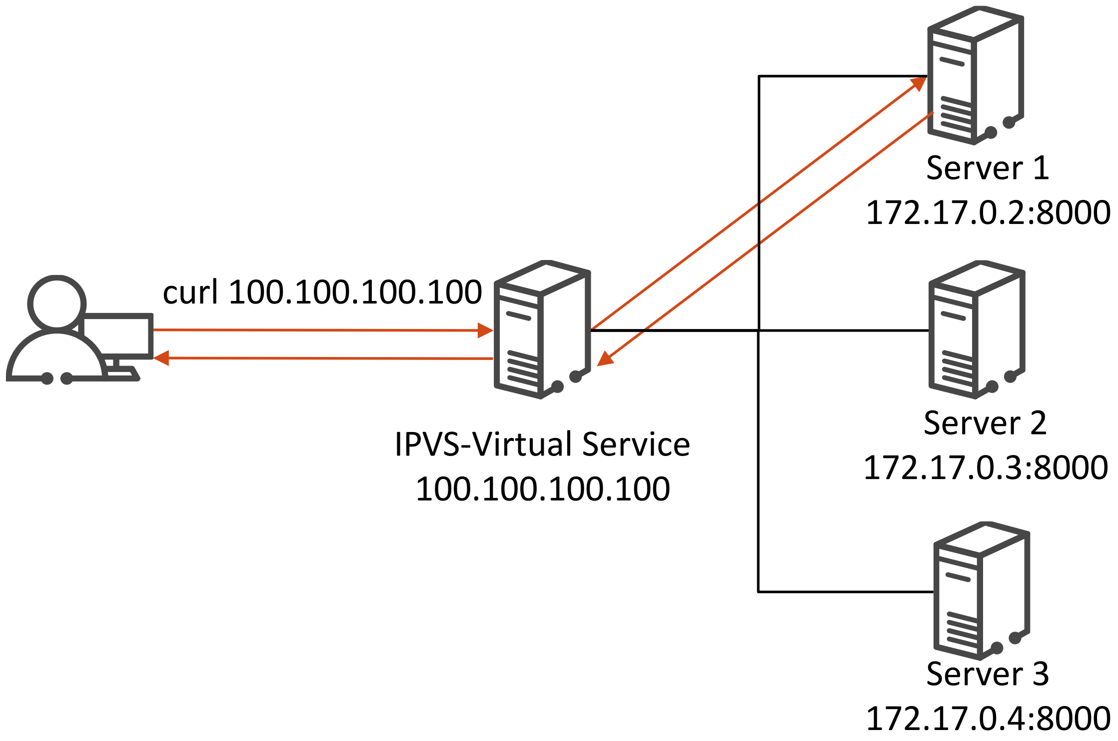
DNS-based Load Balancer

Not aware of the network status between the client and the site.



Layer 4 Load Balancer

Not designed for servers across the WAN



What is CFN-dyncast trying to do ?

How about using the network as a large distributed load balancer?

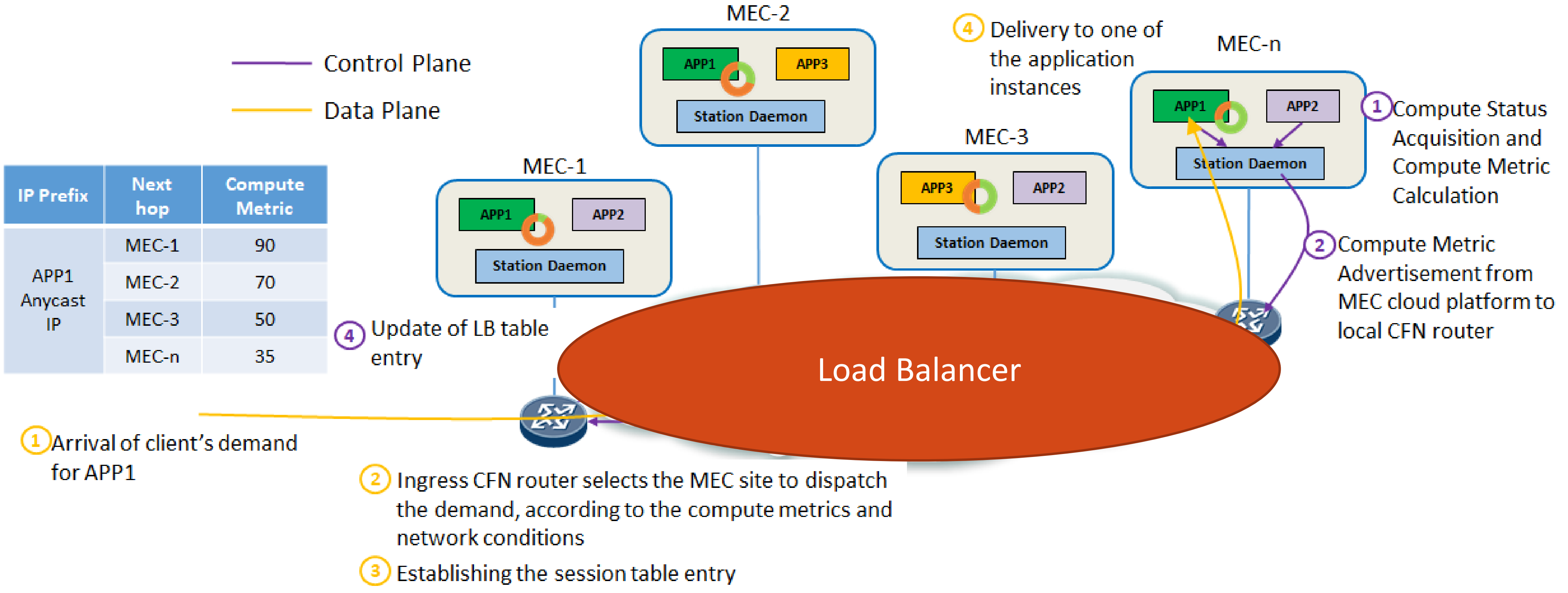
- Be aware of the load of the MEC sites and the network conditions
- Dispatch the client’s requests dynamically to the optimal MEC site
- Transparent to the client

Design considerations

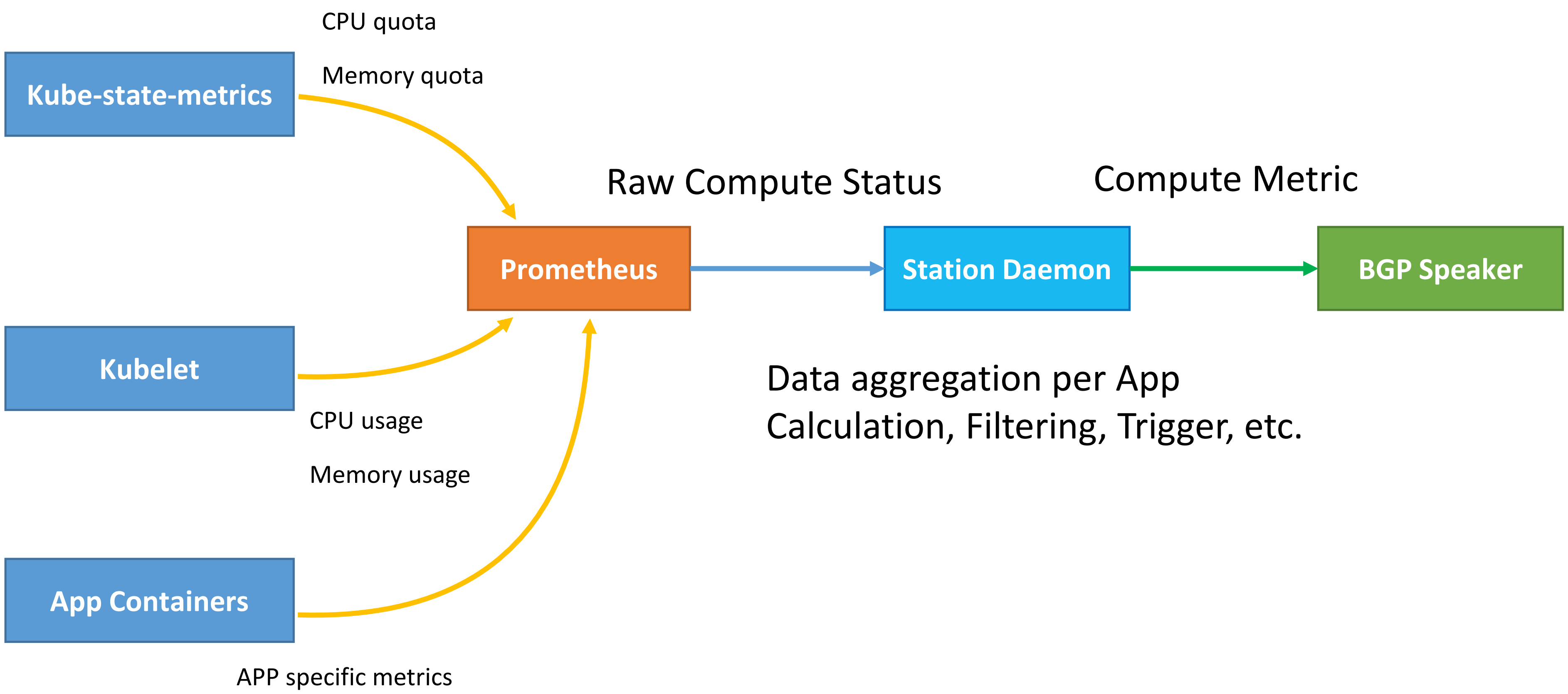
- Compute status acquisition with application granularity
- Compute status advertisement
- Backward compatibility
- Session affinity

How does it work ?

IP Prefix	Next hop	Compute Metric
APP1 Anycast IP	MEC-1	90
	MEC-2	70
	MEC-3	50
	MEC-n	35



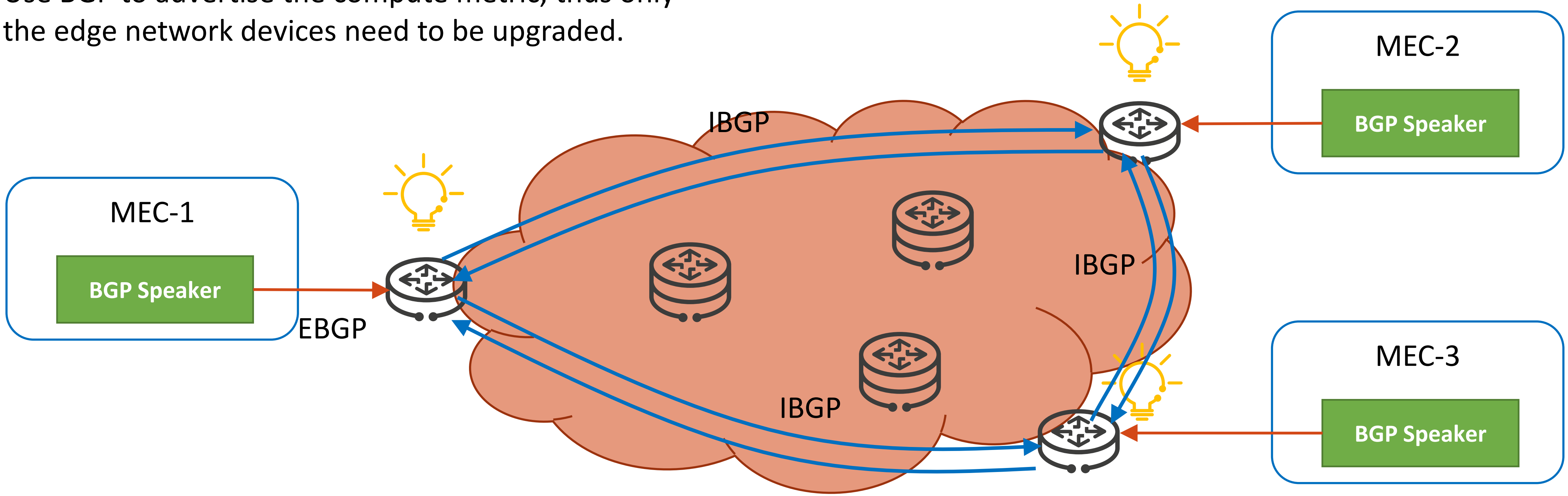
Compute Status Acquisition



Data aggregation per App
Calculation, Filtering, Trigger, etc.

Compute Metric Advertisement

Use BGP to advertise the compute metric, thus only the edge network devices need to be upgraded.



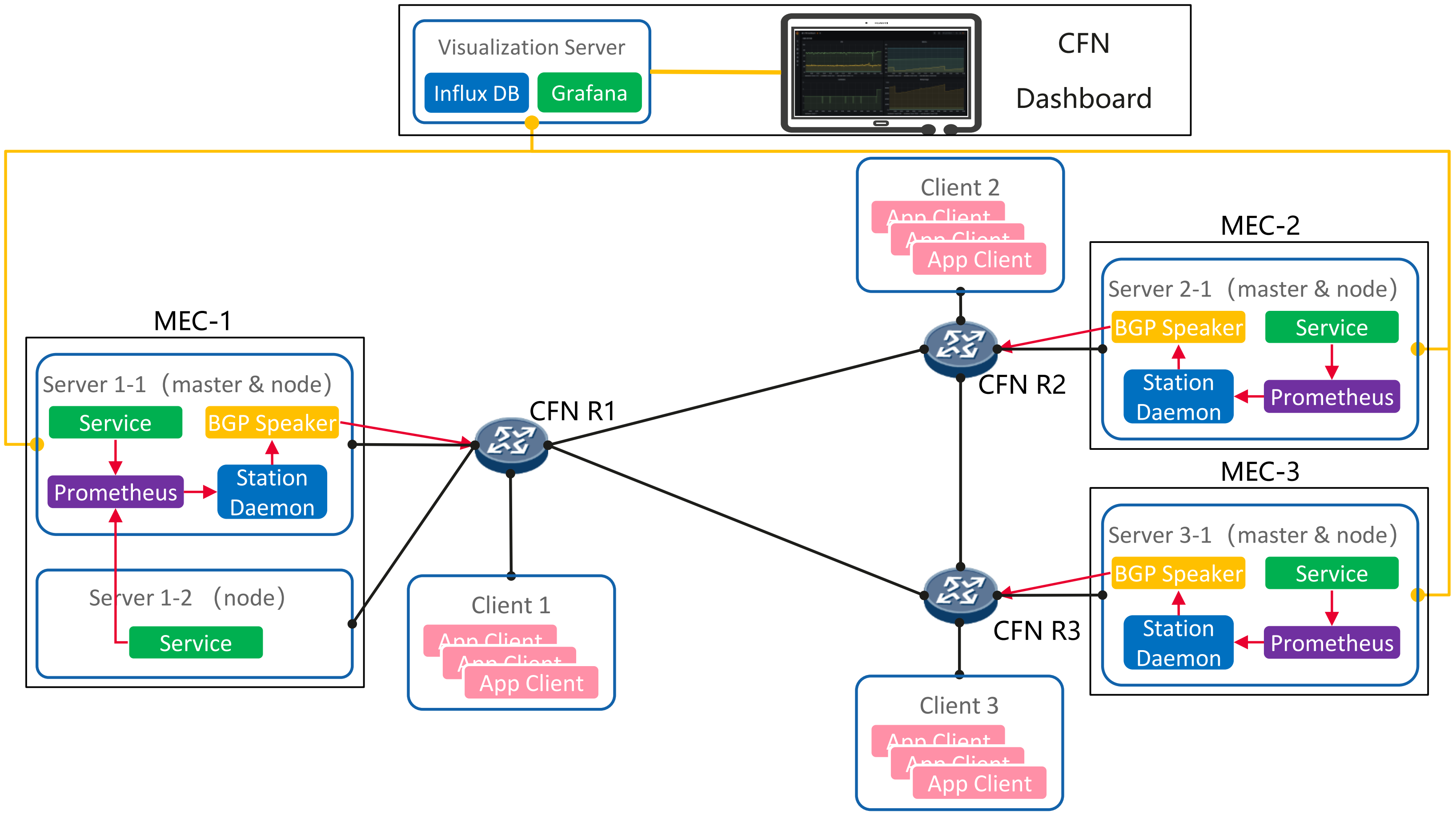
Session Affinity

Session Table

- Built at the ingress CFN router
- An entry is added when the dispatching decision is made
- The subsequent packets are sent to the MEC site as indicated by the table entry

Session Identifier					Egress	Timeout
SRC_IP	DST_IP	SRC_PORT	DST_PORT	PROTO		
Client1	APP1	aaaa	bbbb	TCP	CFN R1	xxx
Client2	APP2	cccc	dddd	TCP	CFN R3	yyy

The Experiment Environment



Three LB methods in the Experiment

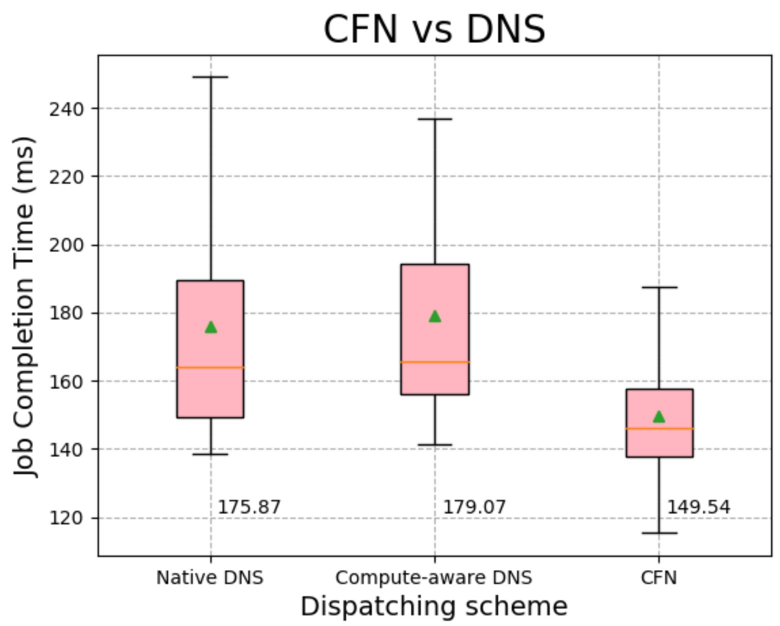
Methods	Description	Destination IP
Native DNS	The weights of the records are set statically based on the capacities of MEC sites. The weight is set to 2:1:1 on our testbed.	IP address of a certain MEC site
Compute-aware DNS	The weights of the records are set dynamically, based on the compute metric.	IP address of a certain MEC site
CFN-Dyncast	All the instances of an application are hidden behind an anycast IP, the MEC sites' addresses are not visible by the clients. The CFN routers are responsible to dispatch the clients' demands.	Anycast IP

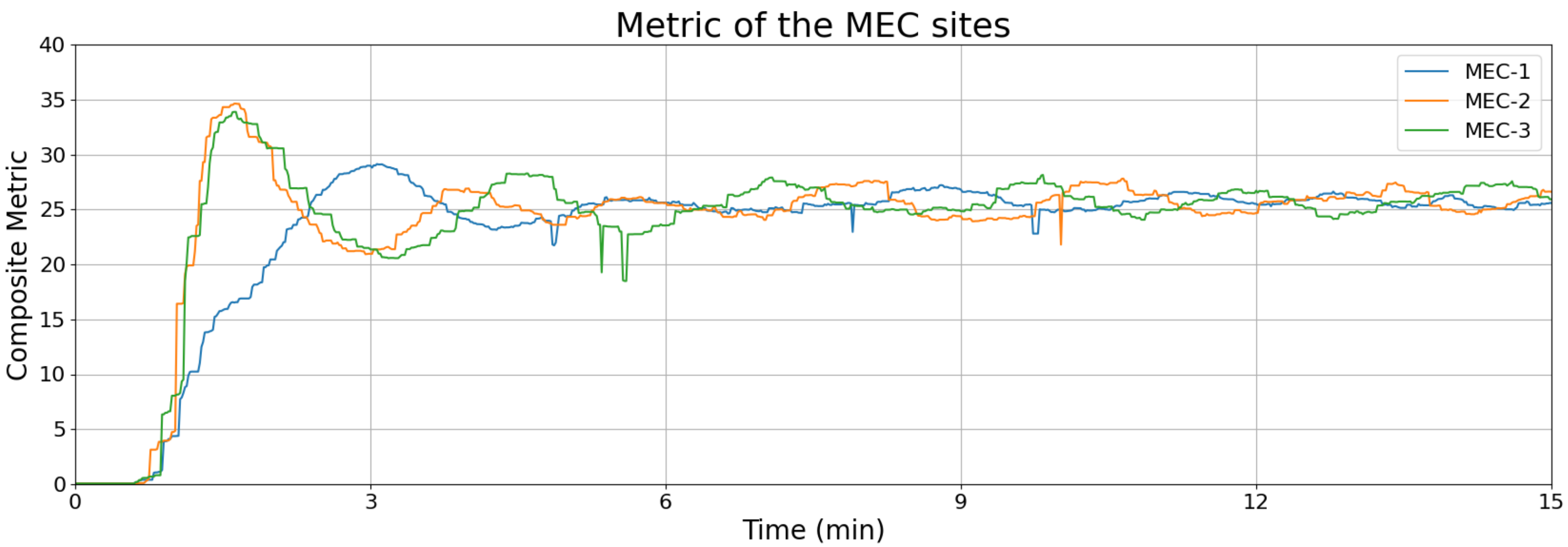
Results

Job Completion Time (JCT)

The time duration that the client has to spend until it gets the response from the server.

	Native DNS	Compute-aware DNS	CFN-dyncast
Average JCT (ms)	175.87	179.07	149.54
Span between the upper and lower JCT bound (ms)	110.259	95.449	72.256
No. of completed job within 20min	7236	7278	7989





Analysis

Why CFN-dyncast performs better than DNS-based schemes?

a) CFN just spends time on at most a single DNS request. And for compute-aware DNS, the client has to initiate DNS request once its local DNS cache expires, which add extra time to the JCT.

b) The optimal MEC site in a client's local DNS cache can be outdated. Before the cache expires, the actual optimal MEC changes to another one, while the client keeps sending requests to an outdated and suboptimal site, which leads to longer JCT.

Conclusions and future works

- CFN-dyncast, a technique that aims to load balance the MEC sites in consideration of the compute status in application granularity and the network conditions.
- Evaluation result shows that
  - CFN-dyncast is capable to dynamically maintain the load of different MEC sites at the same level.
  - Compared to centralized dispatching schemes, CFN-dyncast helps the clients get replied by the servers in a shorten period.
- Next steps, we will try to evaluate CFN-dyncast on wide area network, in which the network condition could be a more remarkable element.

Thank you!