

Cloud-native abstractions for network programmability

Landon Cox, Sanjeev Mehrotra, Samir Outalha, and
Ricardo Villarreal

Azure for Operators

What are Network APIs?

Operators provide access to network configuration and telemetry via **Network APIs**

Developer



Operator



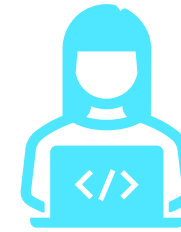
Devices



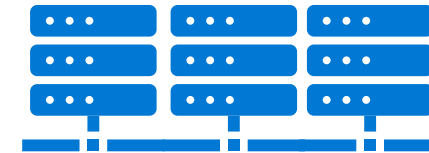
What are Network APIs?

Operators implement **Network APIs** using 3GPP functions like 5G NEF and 4G SCEF

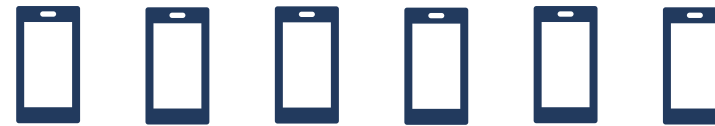
Developer



Operator



Devices



How could we use Network APIs?

NETWORK CORE



Coarse workload placement

- API provides device's coarse location
- Orchestrator places workload on closest edge



Workload adaptation

- API provides near real-time network performance
- Orchestrator adapts pod placement



IoT device management

- API notifies when device's network status changes
- Device provisioning and monitoring



Traffic influence

- API accepts routing rules
- Secure tunneling between device and servers

RADIO ACCESS NETWORK (RAN)



Precise workload placement

- API exposes device UPF location
- Orchestrator places workload on same cluster as UPF



Precise localization

- API provides real-time wireless channel state
- Precise (cm-level) device localization



Precise sleep cycling

- API ensures predictable data-delivery cadence
- Radios can sleep between deliveries



Network QoS

- API ensures predictable network performance
- Low-latency, low-jitter connections to edge/cloud (AR/gaming, robotics)

Network APIs are difficult to use

IRREGULAR INTERFACES

- Each operator's API will be a different flavor (even if NEF based)
- As apps scale, they must interact with more and more APIs
- Requires learning and maintaining code for each operator API



SCALABILITY CONCERNS

- Must build event-driven programming model on REST APIs
- New HTTP request for each notification
- Naïve approach unlikely to scale to billions of devices

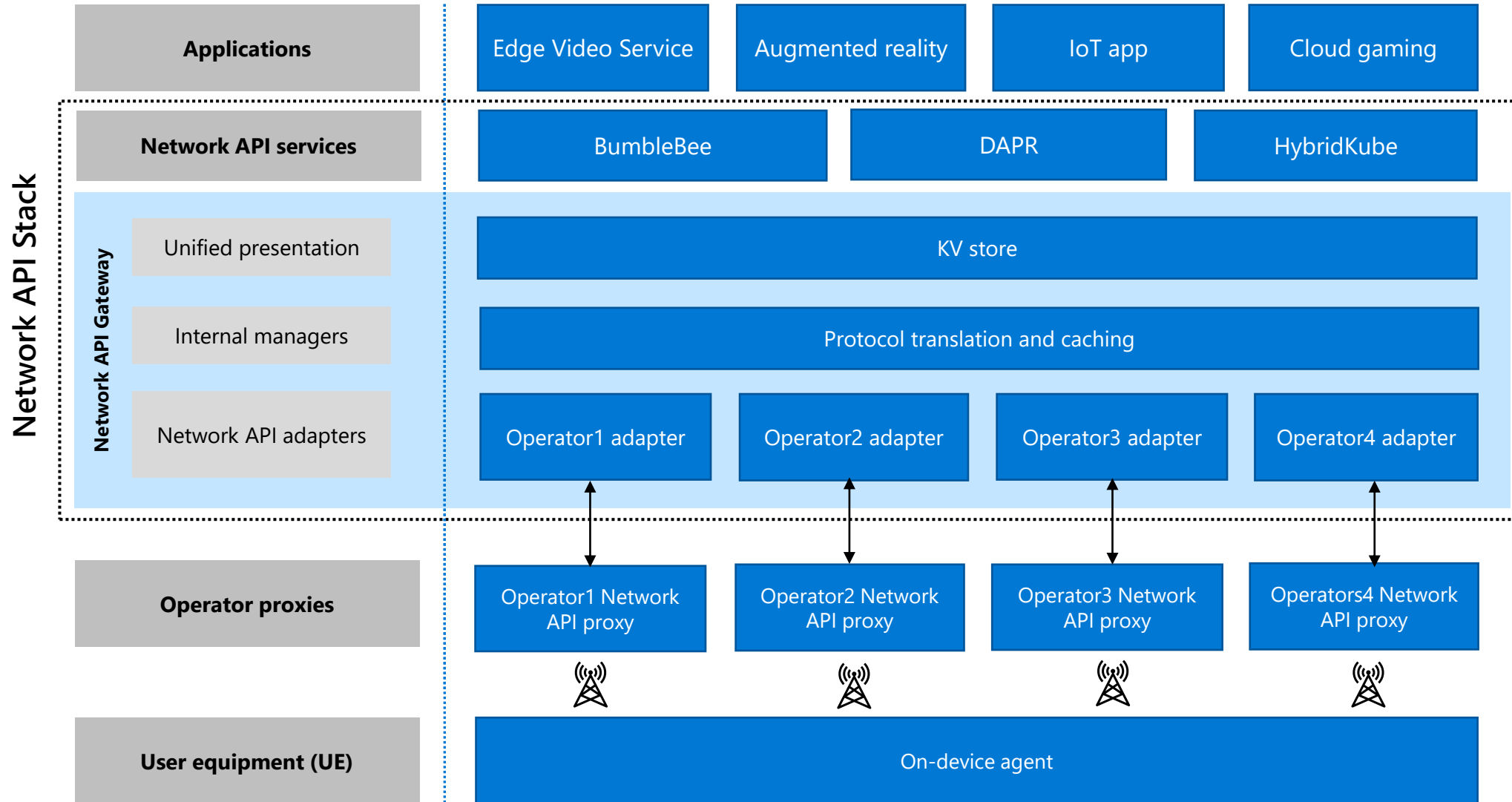


LOW-LEVELS OF ABSTRACTION

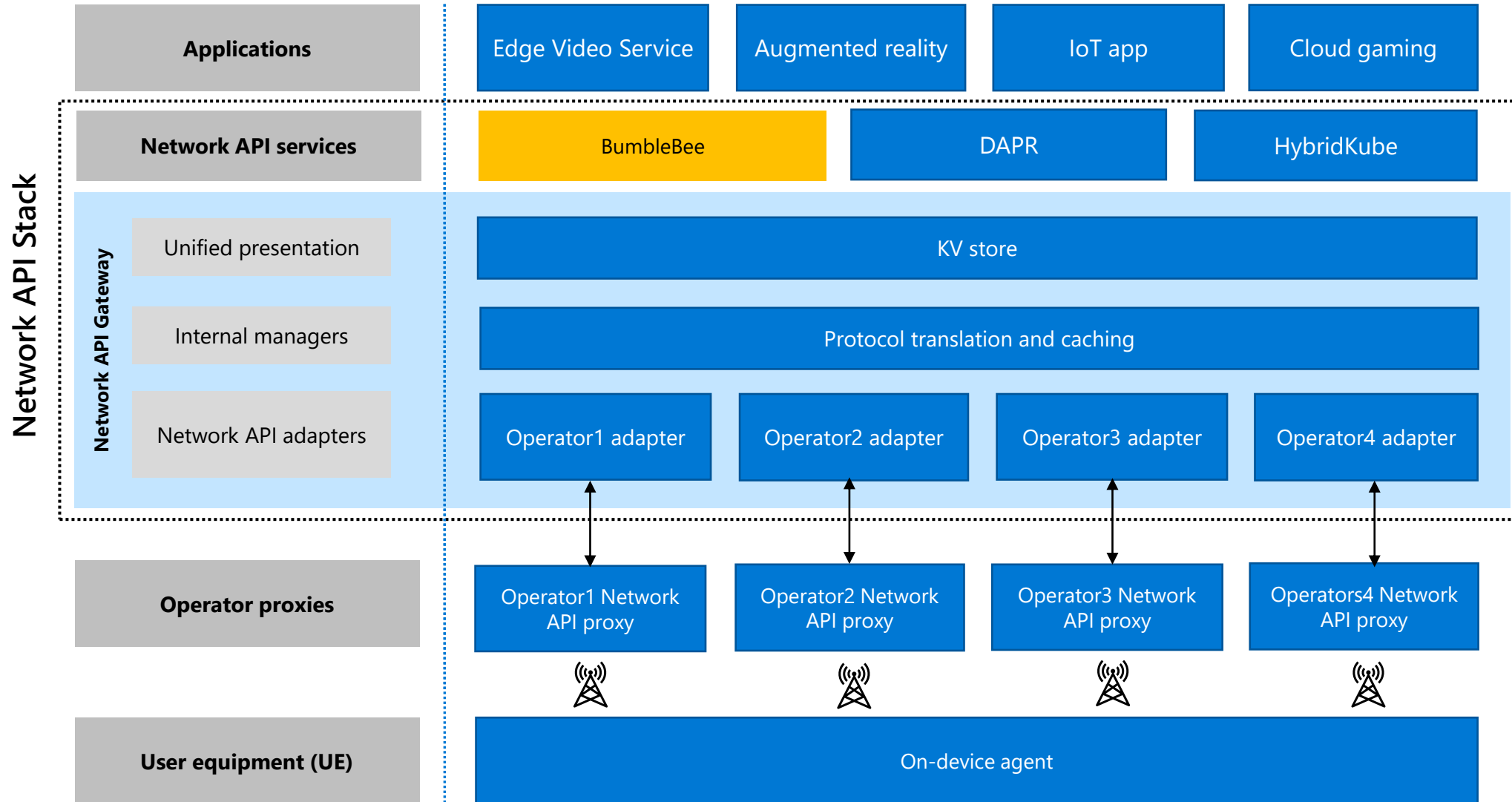
- Network APIs are too low-level for most developers
- Most developers don't care about Network API minutiae
- Want higher-level services that use network programmability



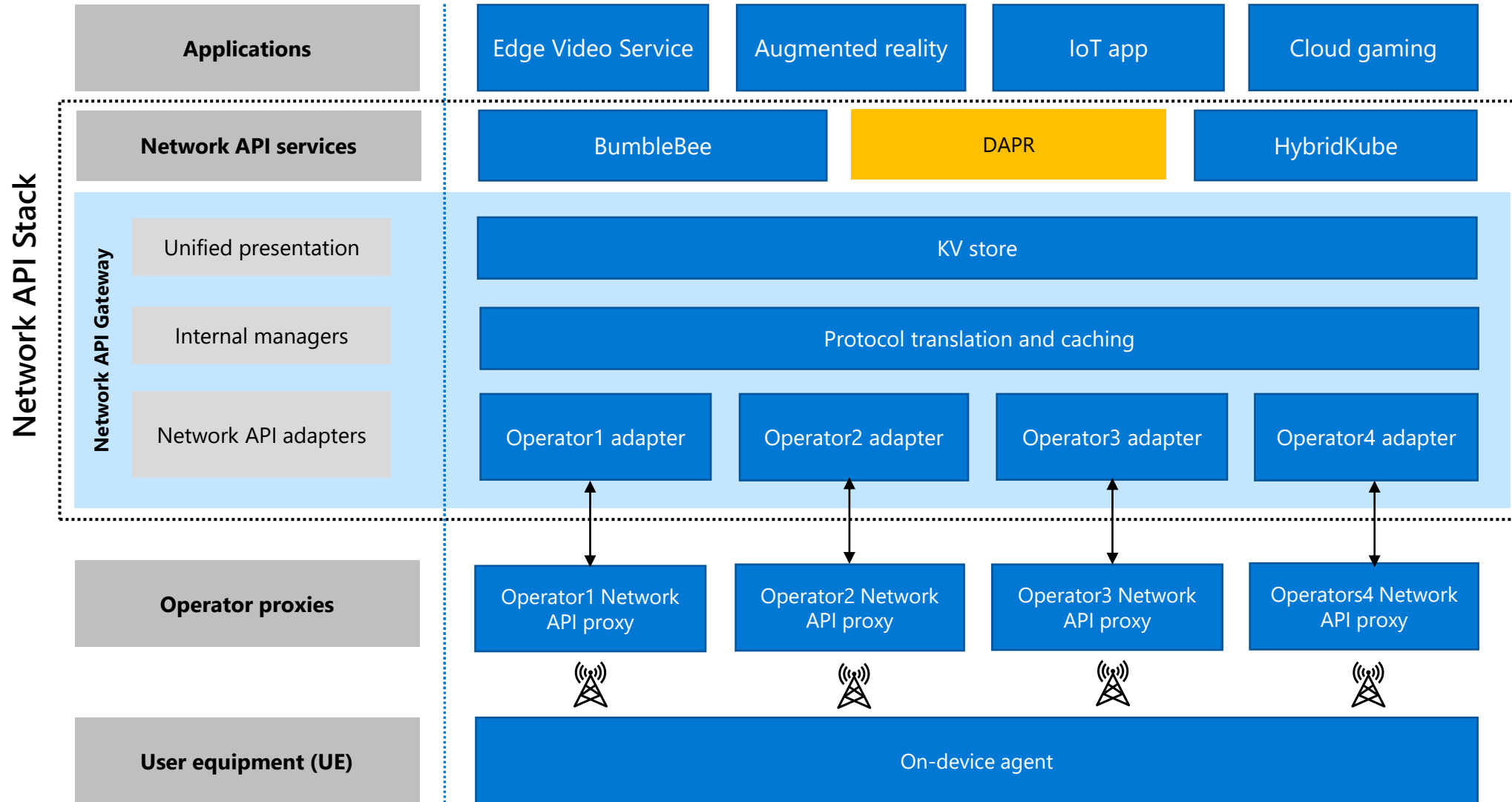
High-level architecture



High-level architecture



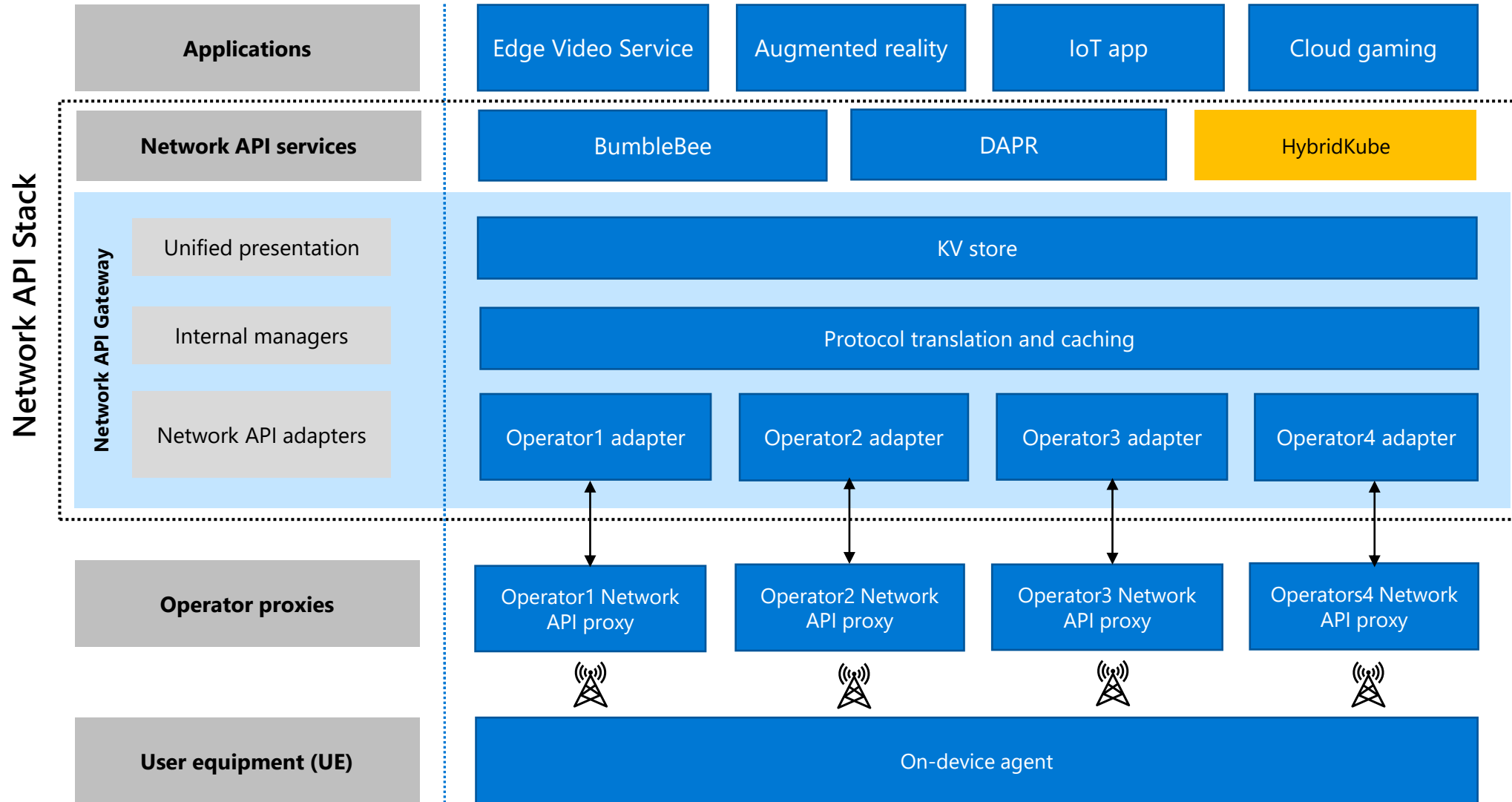
High-level architecture



BumbleBee/DAPR demo from MSBuild 2020



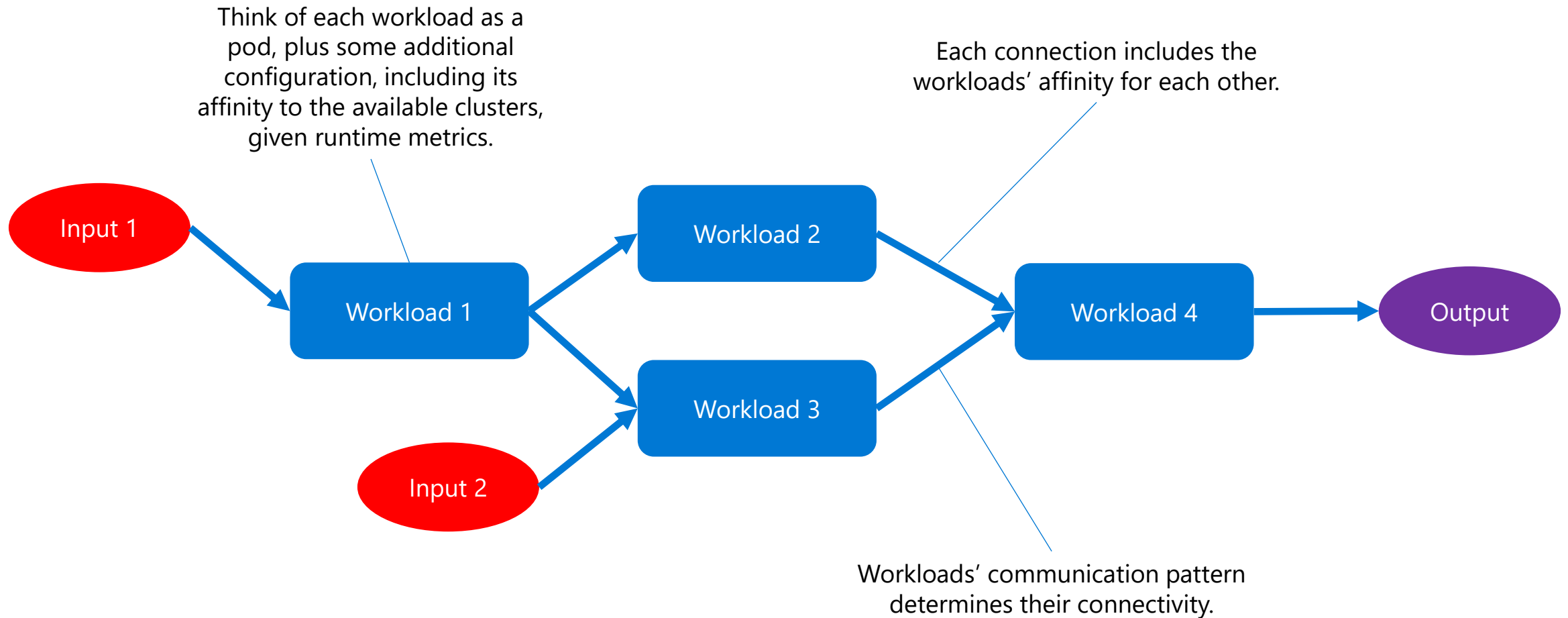
High-level architecture



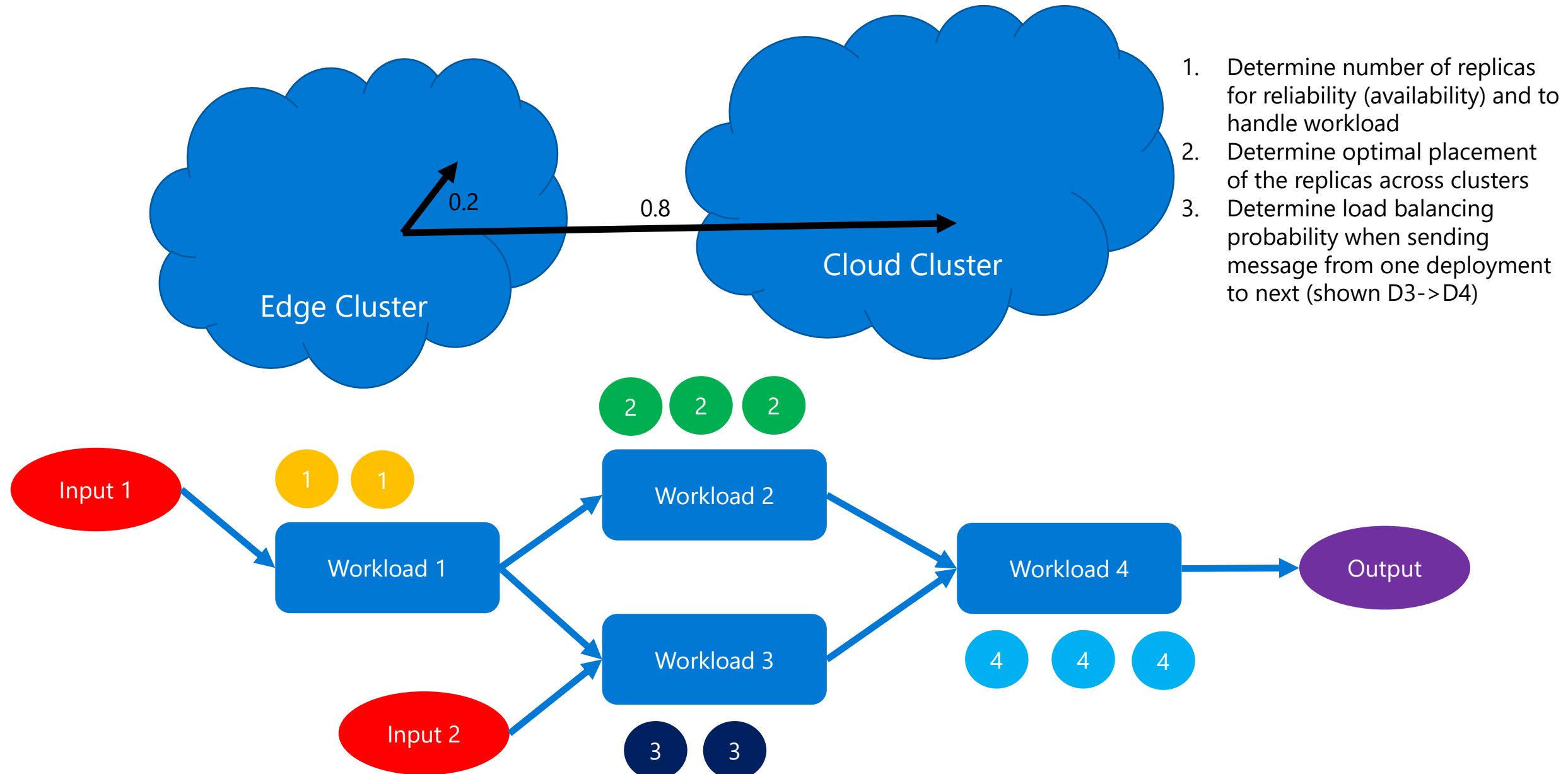
HybridKube

- Kubernetes extension that manages cross-cluster workloads
 - e.g., an edge cluster and a cloud cluster
- Exposes a declarative **pipeline abstraction** defining
 - Workloads (e.g., pods + config)
 - Connectivity between workloads
 - Affinity between connected workloads
 - Affinity of workloads to clusters (modulo runtime metrics from Network API)
- Optimizer continuously monitors and adjusts workloads' placement

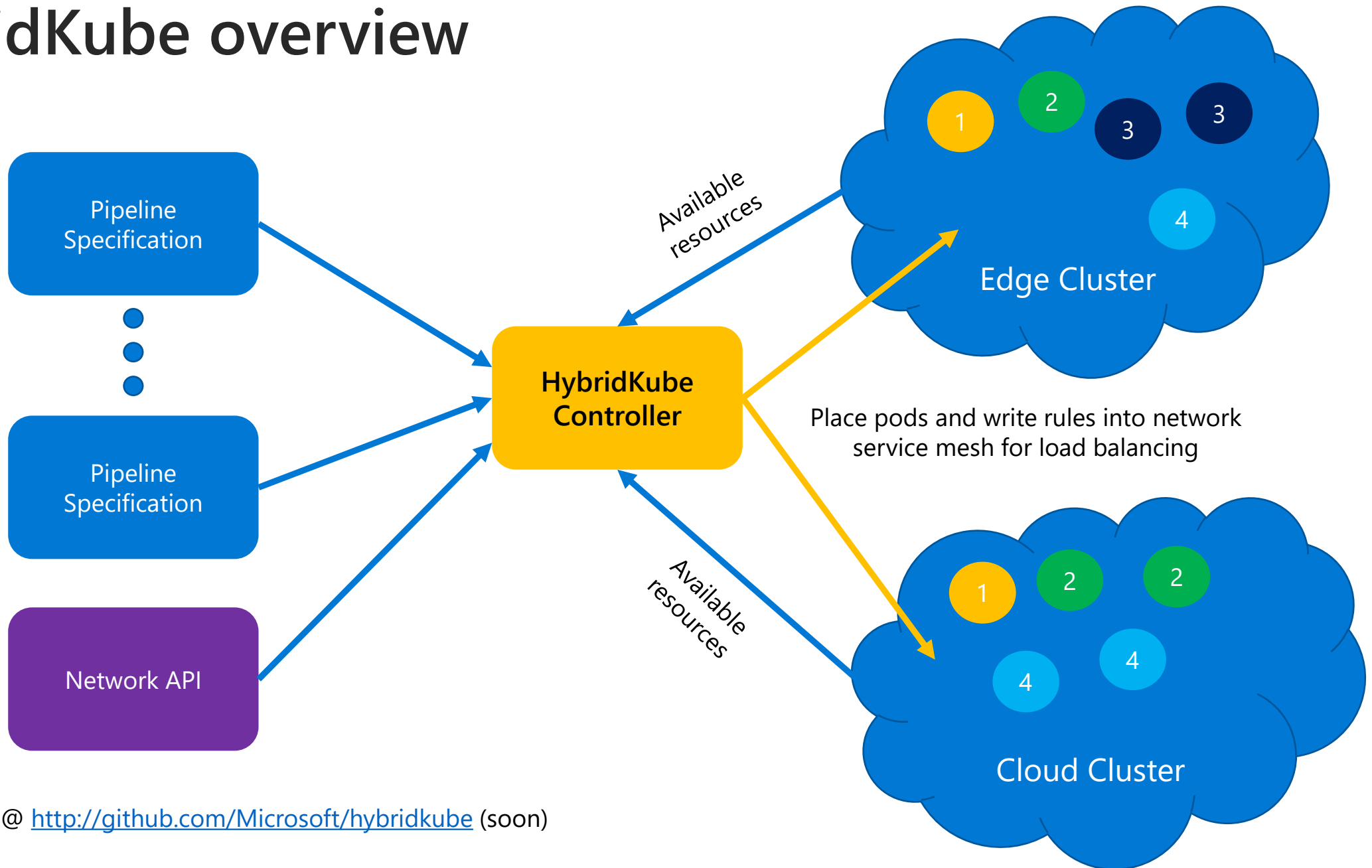
Example pipeline



Example placement



HybridKube overview



Code will be @ <http://github.com/Microsoft/hybridkube> (soon)

Network APIs ~~are difficult to use~~ *made simpler*

~~IRREGULAR INTERFACES~~

k8s

YAML

- Each operator's API will be a different flavor (even if NEF based)
- As apps scale, they must interact with more and more APIs
- Requires learning and maintaining code for each operator API



SCALABILITY CONCERNS

?

- Must build event-driven programming model on REST APIs
- New HTTP request for each notification
- Naïve approach unlikely to scale to billions of devices



~~LOW LEVELS OF ABSTRACTION~~

Pipelines

- Network APIs are too low-level for most developers
- Most developers don't care about Network API minutiae
- Want higher-level services that use network programmability



Thank you

- Project links
 - Azure for Operators: <https://azure.microsoft.com/en-us/industries/telecommunications/>
 - BumbleBee: <https://arxiv.org/pdf/2008.11868.pdf>
 - Dapr: <https://dapr.io>
 - HybridKube: <https://www.microsoft.com/en-us/research/project/hybridkube/>
- Please email me if you have follow-up questions
 - Landon Cox (landon.cox@microsoft.com)