



Azure for Operators Overview (and Look Ahead)

Landon Cox

Azure for Operators, Office of the CTO

"5G is the killer app for edge computing."

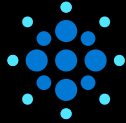
Azure for Operators

Your customer, your service, powered by our technology



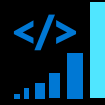
Achieve scale

Carrier grade network functions as a service powered by AI



Operate hybrid seamlessly

Consistent performance and scalability across edge, hybrid, and cloud



Monetize with new business models

Edge compute, IoT, 5G, network slicing



Trust

Global, secure, and compliant, empowers you and your business

Built on a foundation of Telco DNA

Backed by Microsoft's developer ecosystem

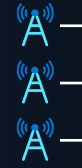
Software-enabled services

- Cloud at the edge
- Virtualized packet and voice core

Enterprise



Edge
Compute



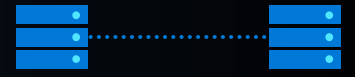
Cell towers
and radios

Edge



Carrier fiber
(WAN)

Core



Virtualized
Packet Core



Virtualized voice
and applications

Cloud-based functions

- 5G control plane in the cloud
- Voice and communication functions as a service

Enterprise



Edge
Compute



Cell towers
and radios

Edge



Carrier fiber
(SD WAN)

Core



Virtualized
Packet Core



Virtualized voice
and applications

Cloud



CNFs/VNFs



5GC-CP

← Orchestration, Security, DevOps →

Cloud-enabled services

- Hybrid control plane
- Network slicing
- Next generation edge services

Enterprise



Edge
Compute



Cell towers
and radios

Edge



vRAN

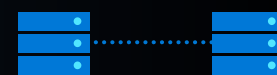


5G UP



Carrier fiber
(SD WAN)

Core



Virtualized
Packet Core



Virtualized voice
and applications

Cloud



5GC-CP



IMS/
VoLTE



CNFs/
VNFs

← Orchestration, Security, DevOps →

My Team's North Star: help edge applications become best-in-class.

Communication Service Providers

Cloud solutions

Gaming

IoT

Collaboration

AR/VR

Network Functions

Voice and applications

Packet core

Radio Access Network

Edge Platform

Enterprise Edge

Network Edge

Cloud Edge

Interconnect

Global WAN

Satellite

SDN

Partners

Communication Service Providers

Cloud
solutions

Gaming

IoT

Collaboration

AR/VR

Javelin

Video ML

Video delivery

Network slicing

Unified Comms

Network
Functions

Voice and applications

Packet core

Radio Access Network

Edge
Platform

Enterprise Edge

Network Edge

Cloud Edge

Interconnect

Global WAN

Satellite

SDN

Partners

Javelin overview



Video ML



Video delivery

Comprehensive video-processing platform
for analytics and distribution.



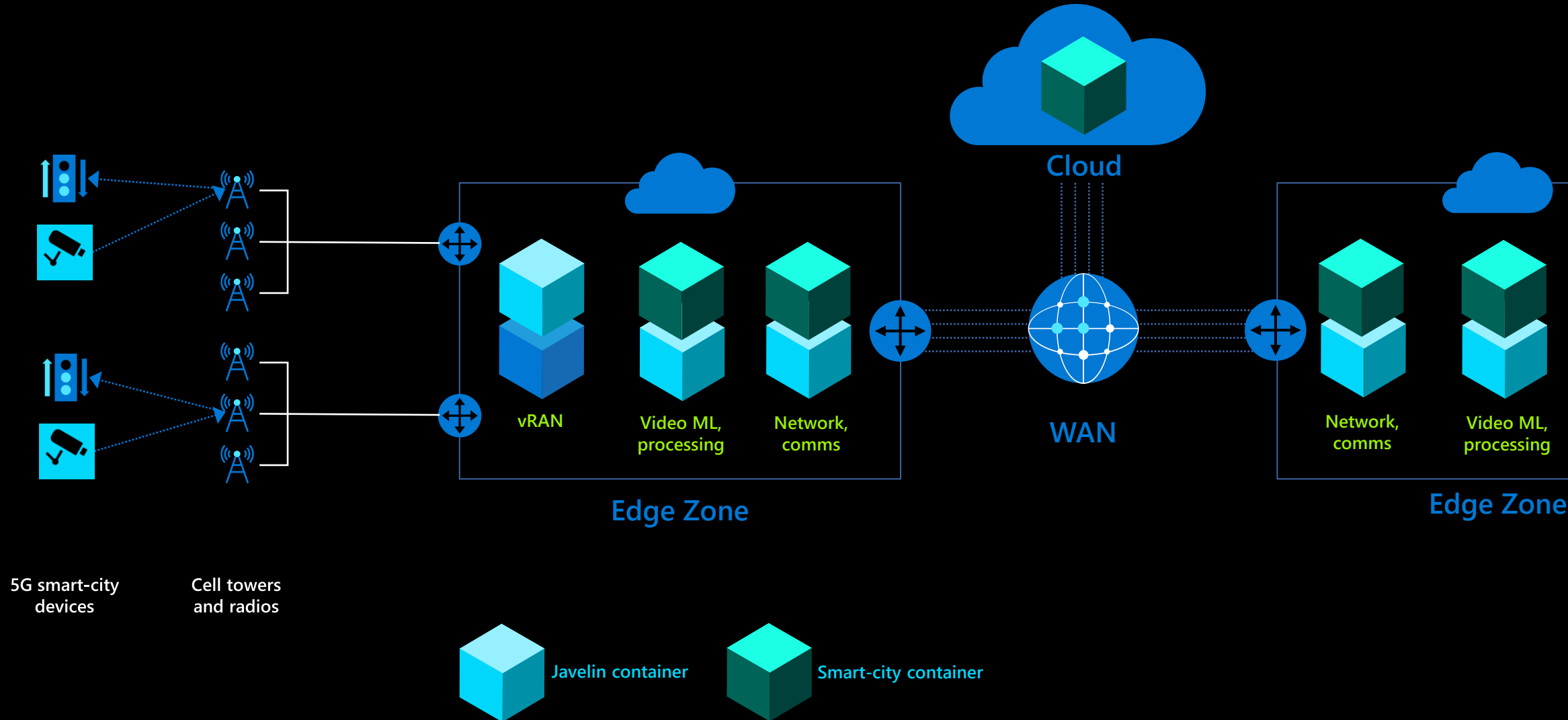
Network slicing



Unified Comms

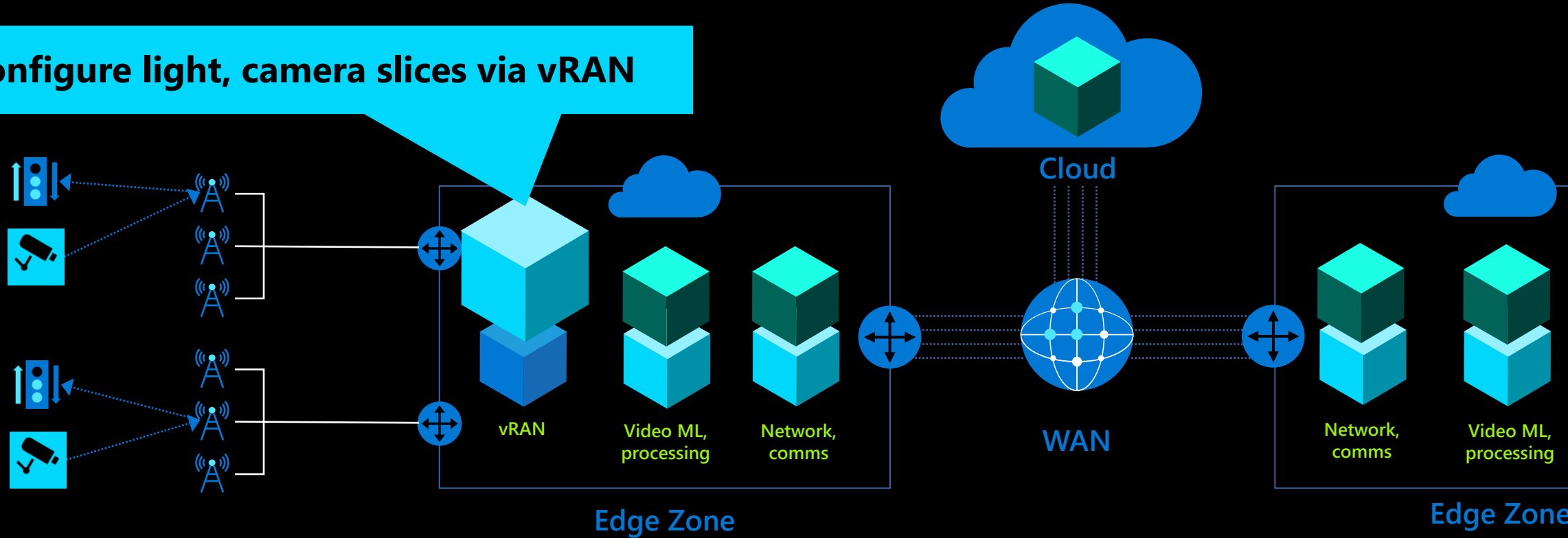
Comprehensive network programmability
across CPaaS, RAN, and WAN.

Smart-city/IoT scenario



Smart-city/IoT scenario

Configure light, camera slices via vRAN



5G smart-city
devices

Cell towers
and radios



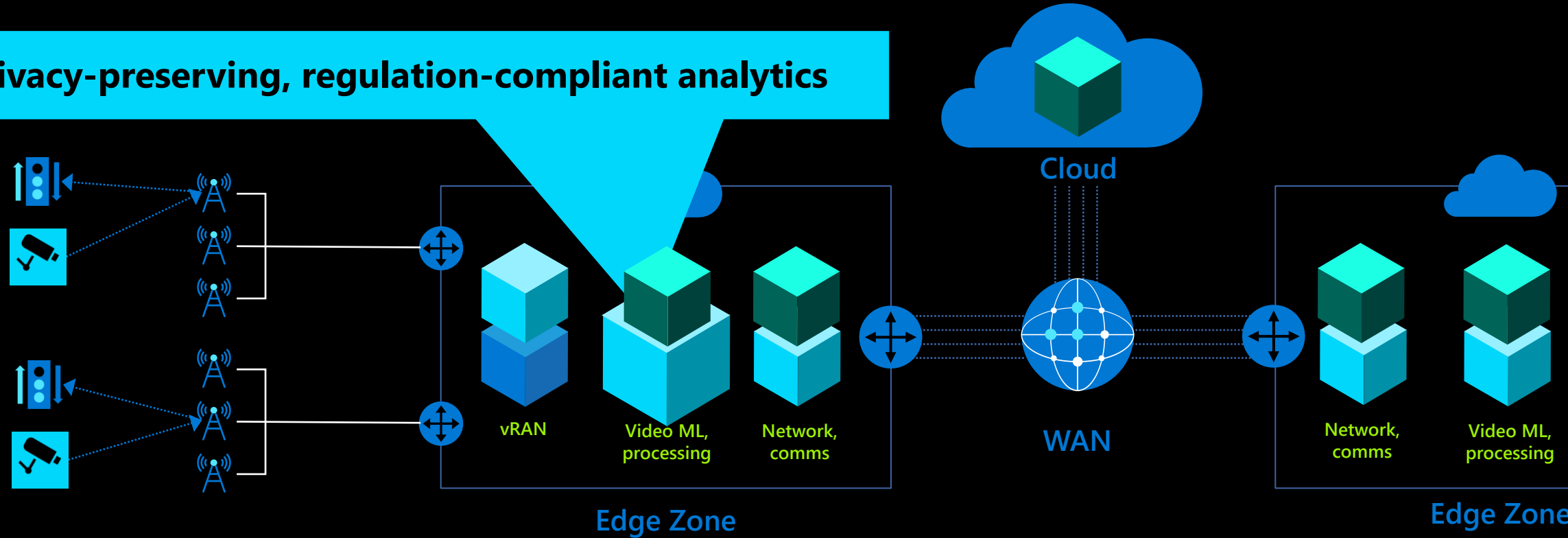
Javelin container



Smart-city container

Smart-city/IoT scenario

Privacy-preserving, regulation-compliant analytics



5G smart-city
devices

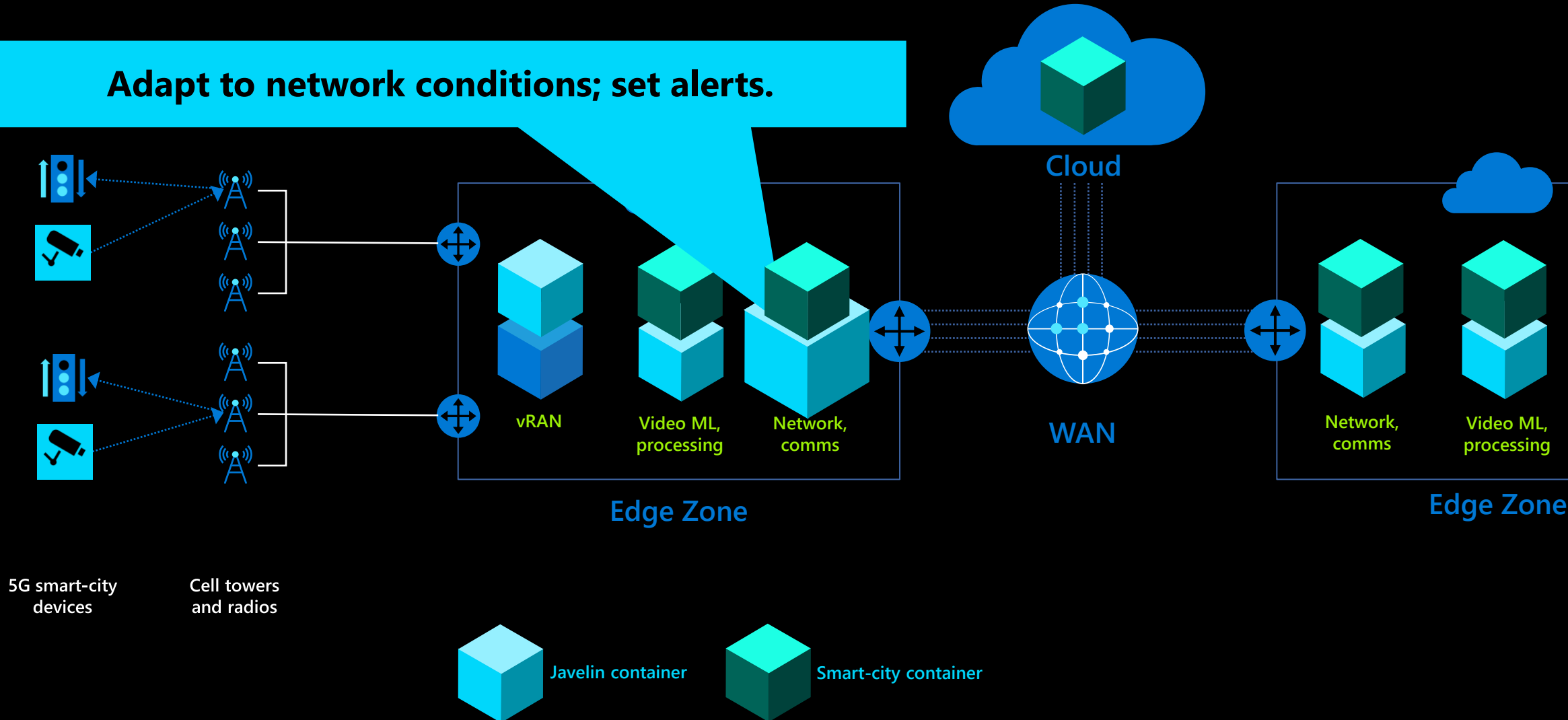
Cell towers
and radios

Javelin container

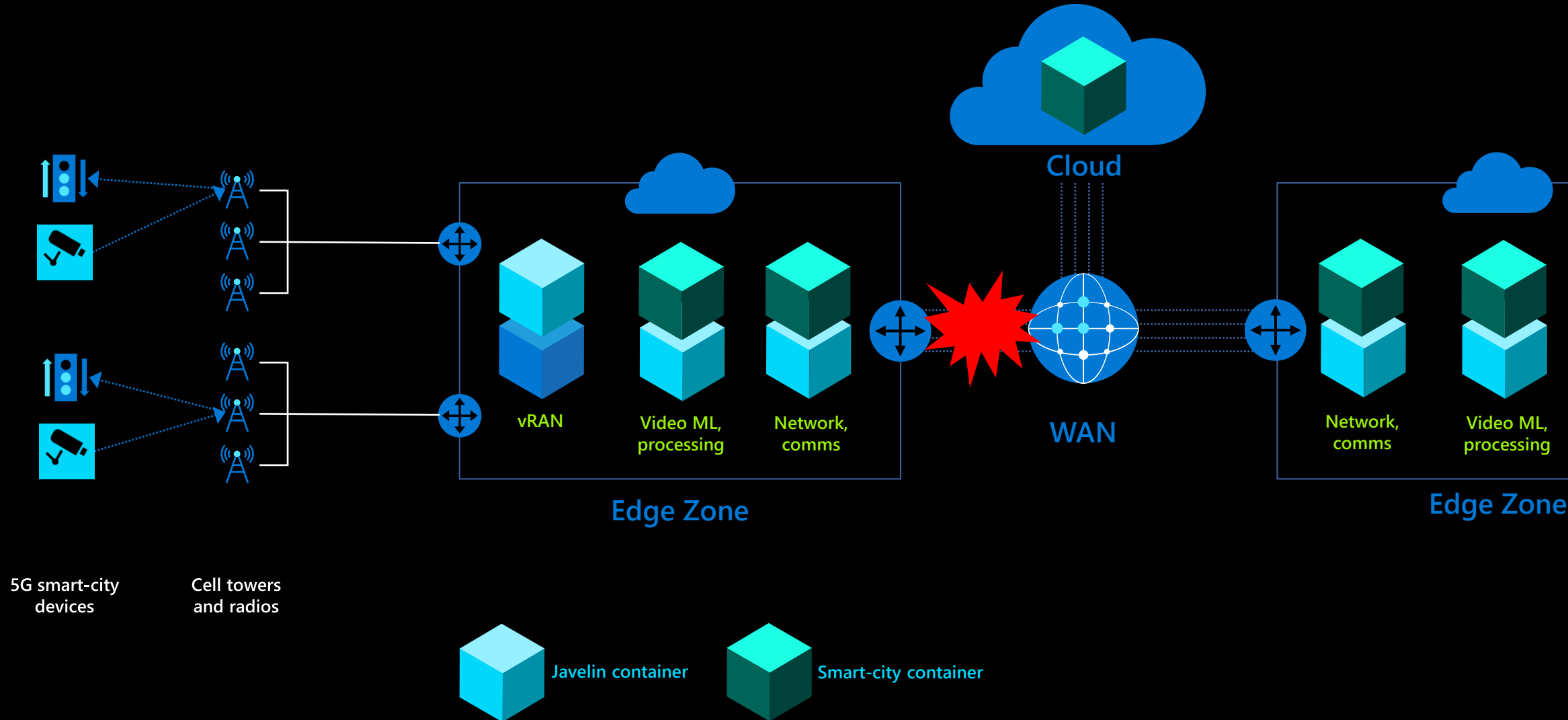
Smart-city container

Smart-city/IoT scenario

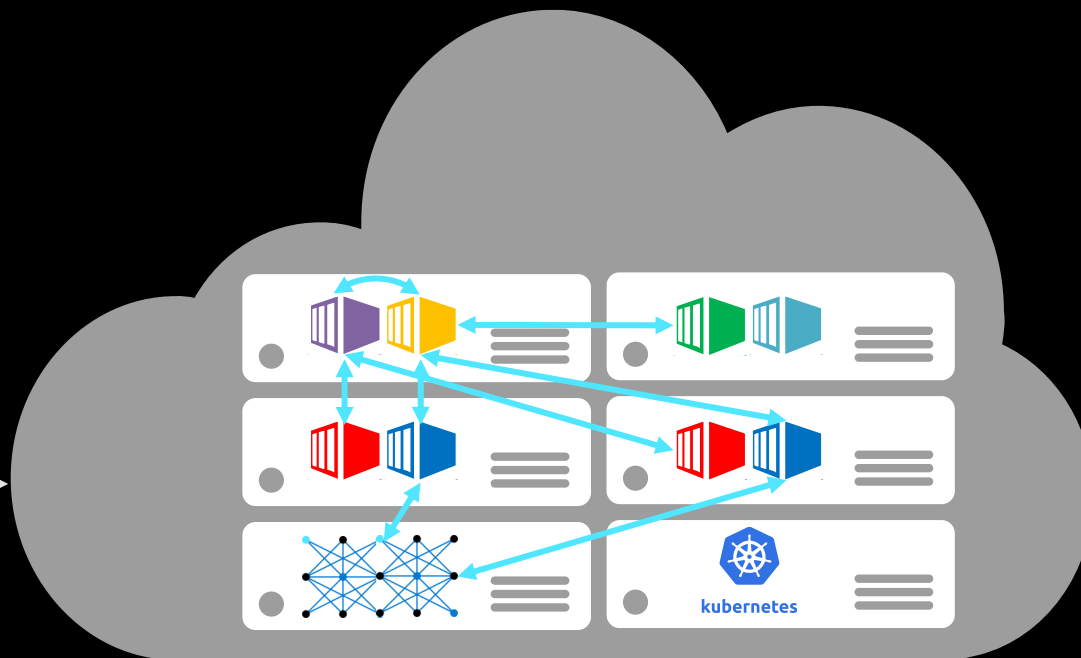
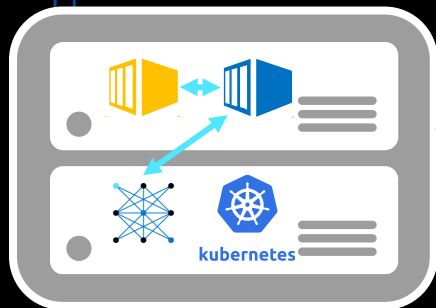
Adapt to network conditions; set alerts.



Smart-city/IoT scenario

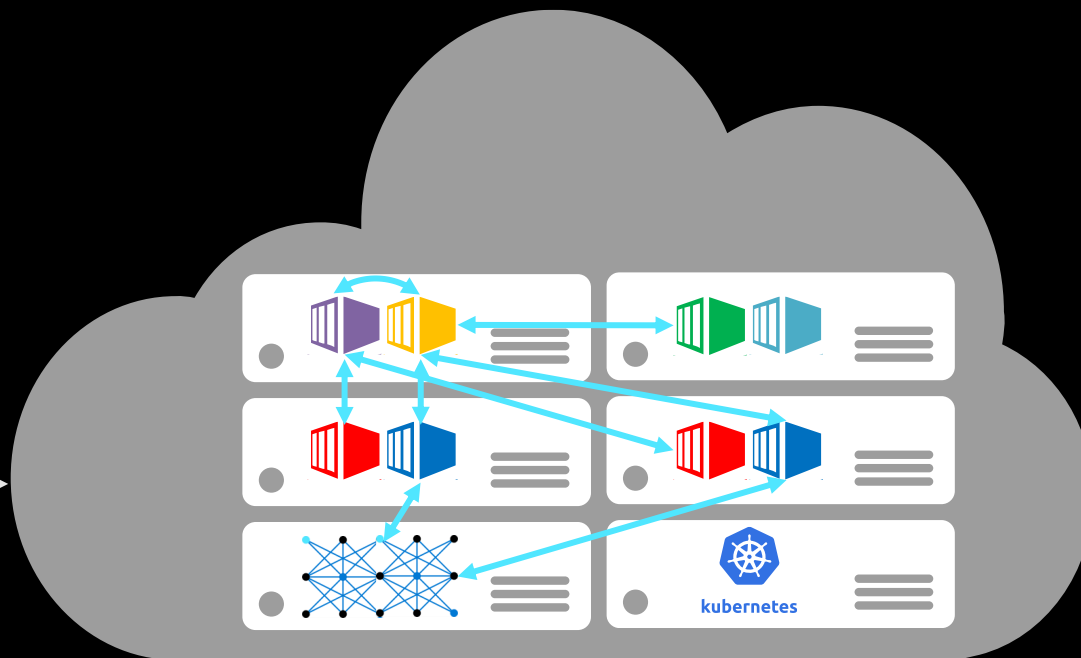
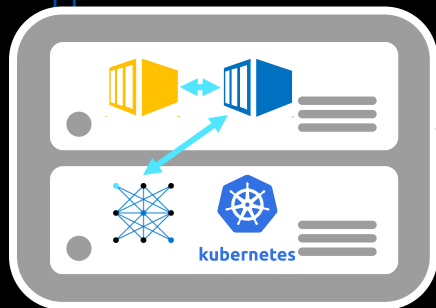


Car-counting application



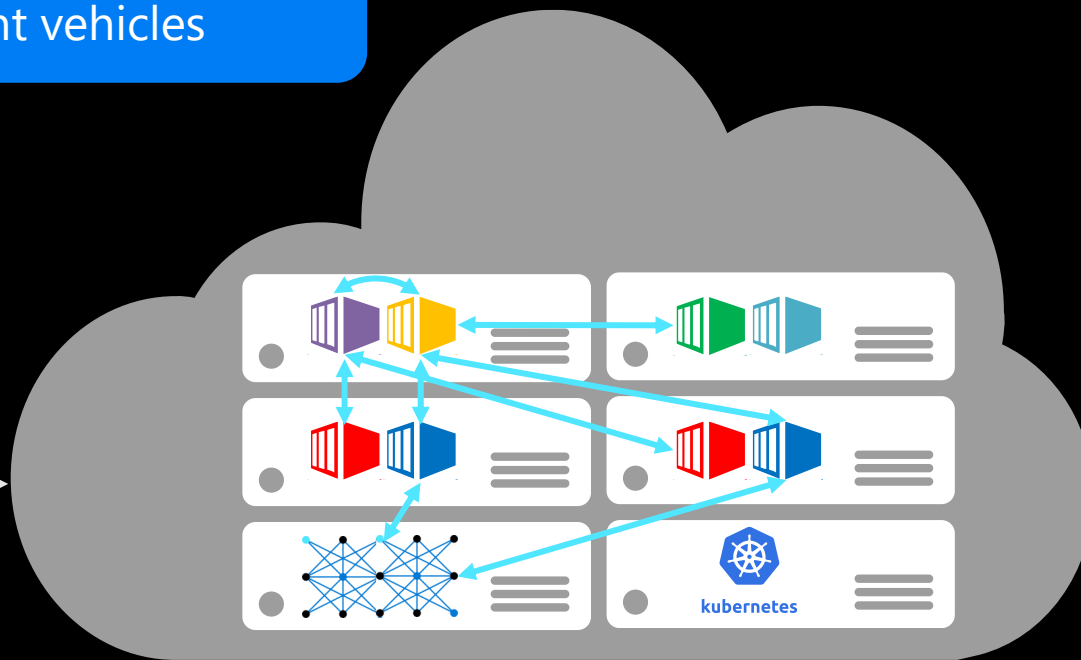
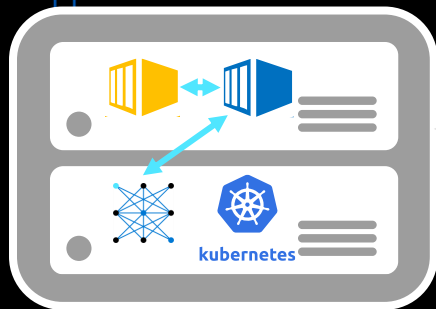


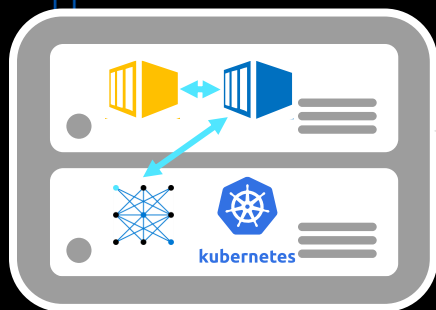
Camera captures passing cars



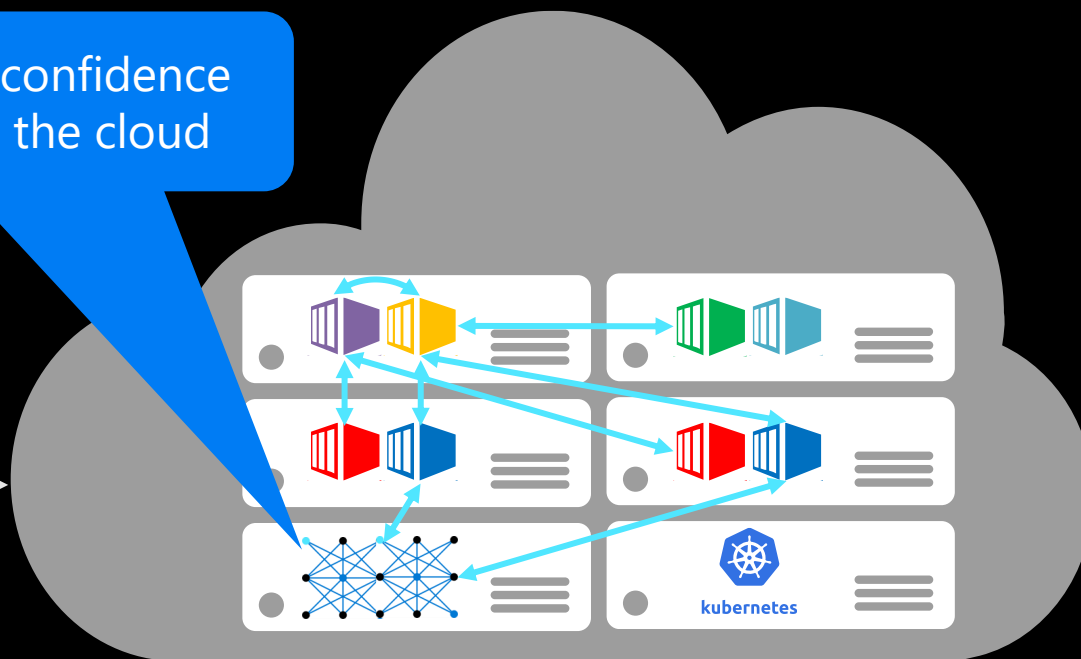


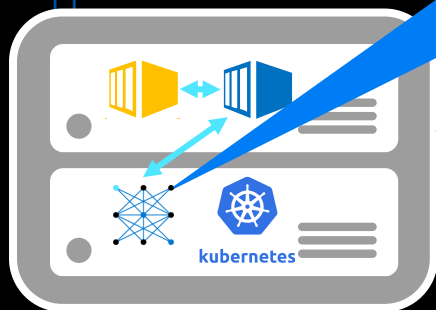
Sends video frames to edge to count vehicles



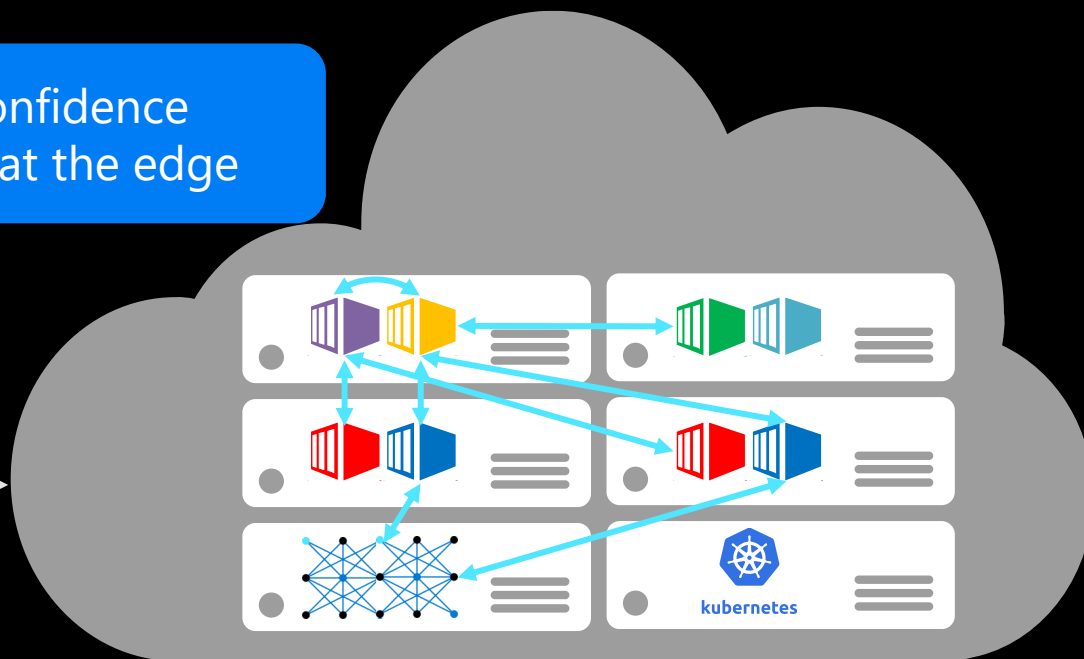


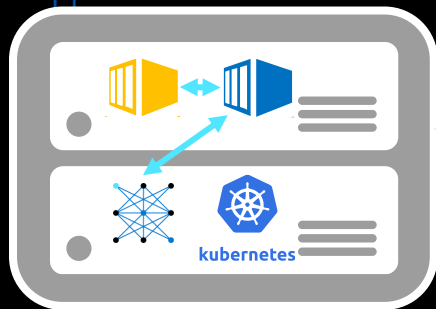
Sophisticated, high-confidence
object-detection in the cloud



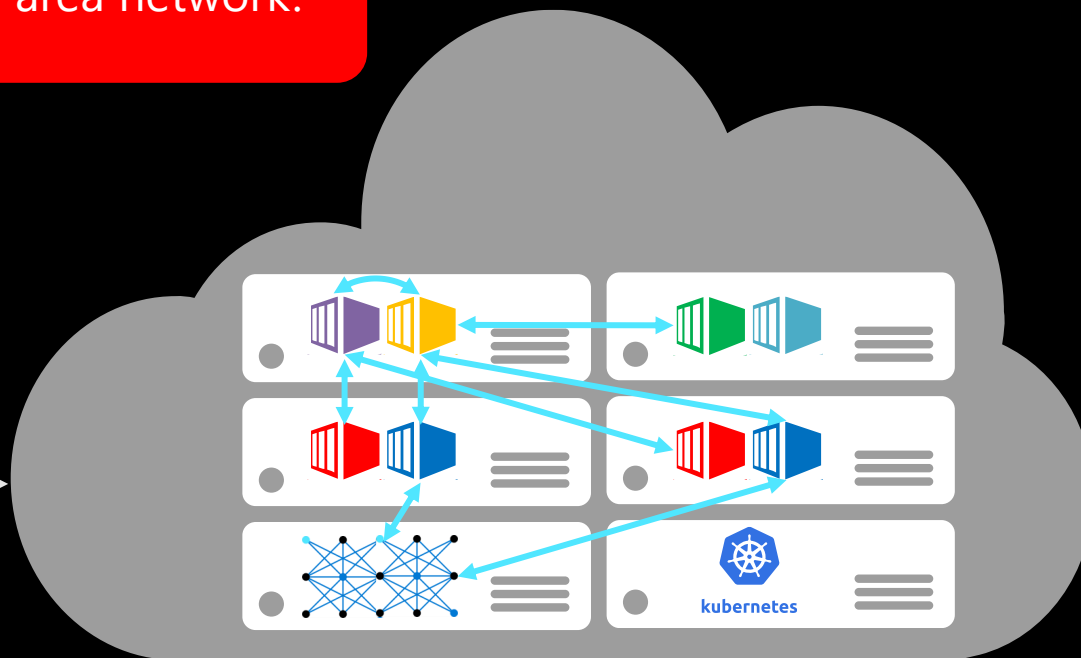


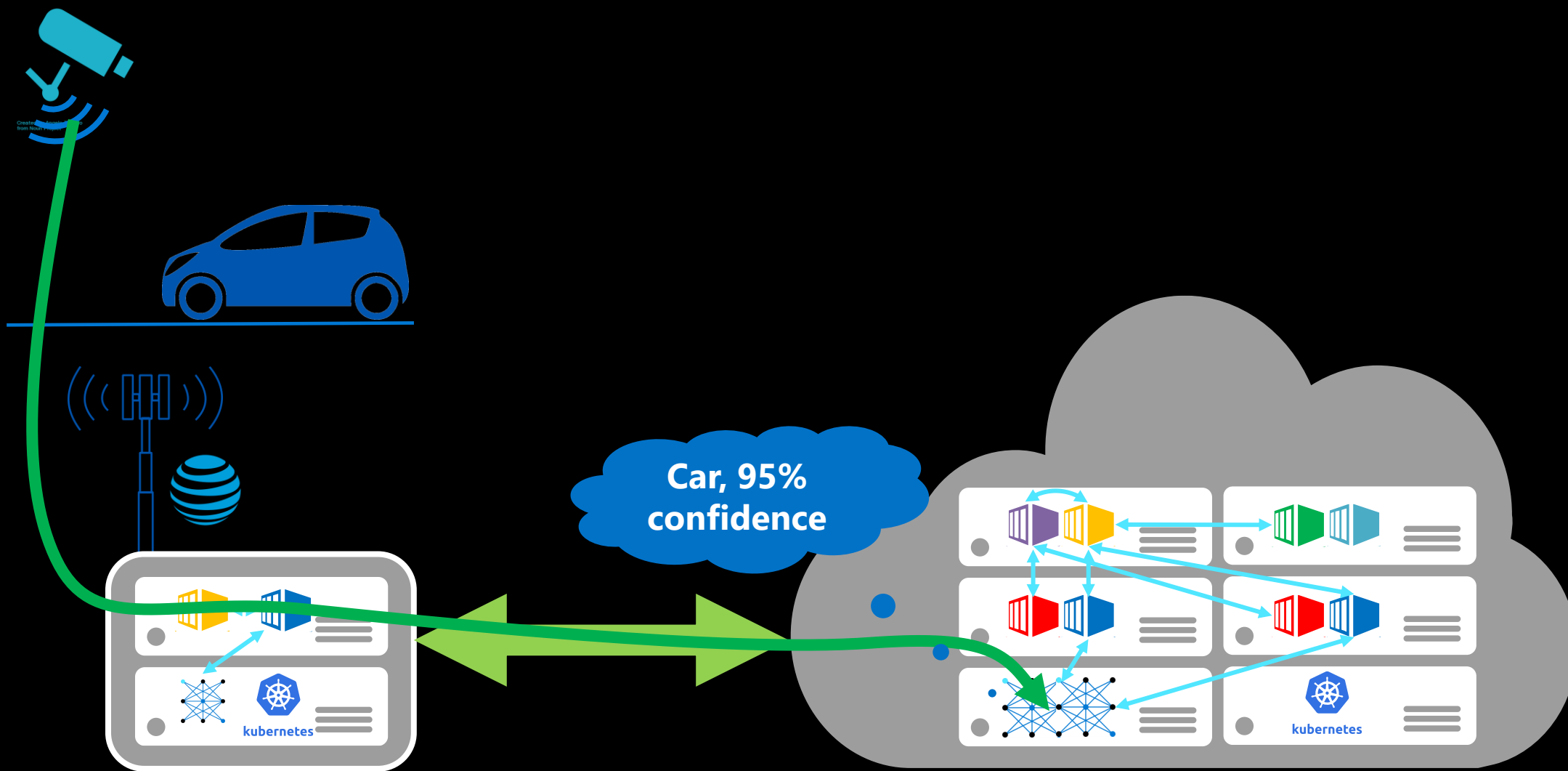
Simpler, low-confidence
object-detection at the edge

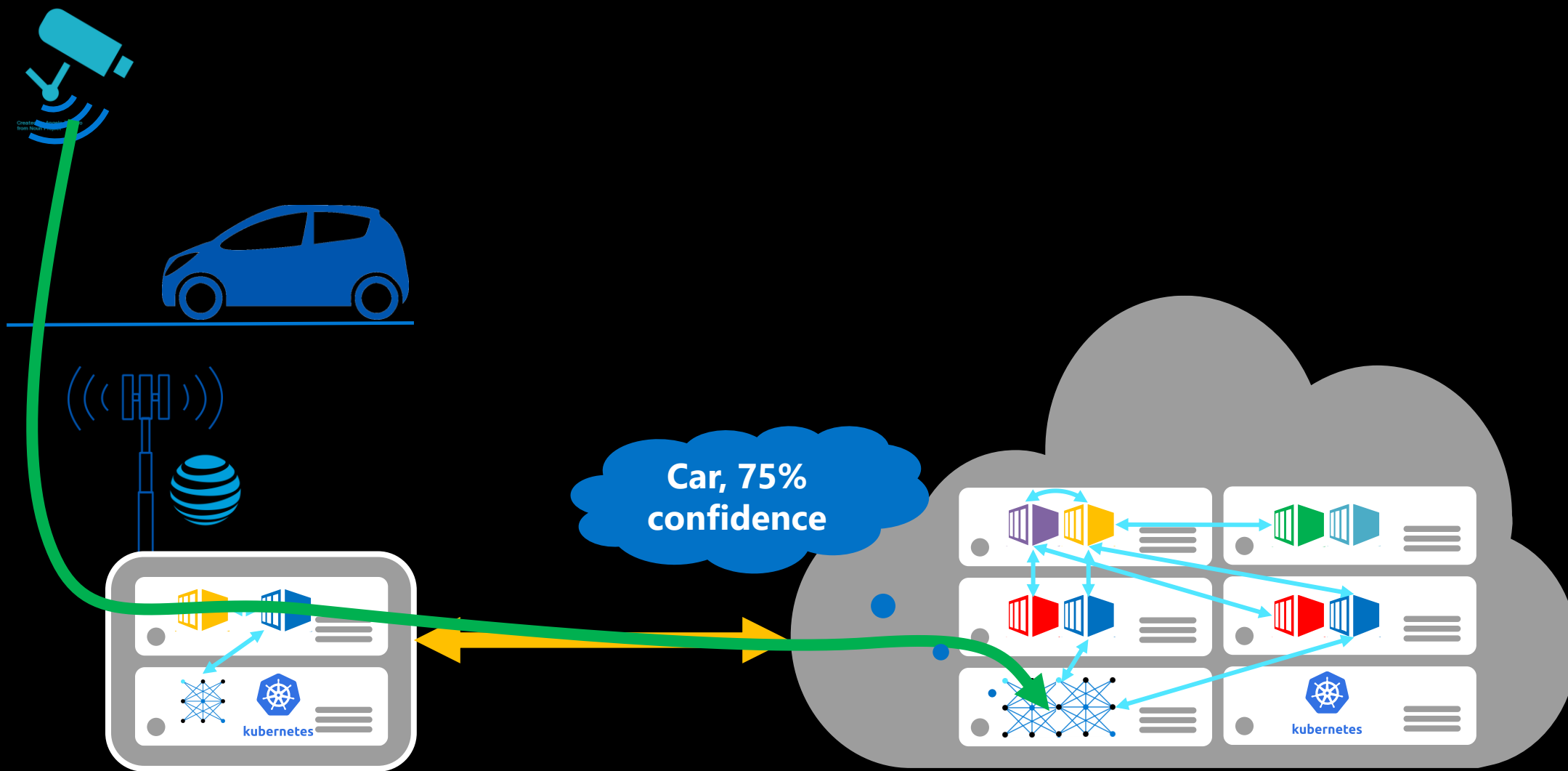




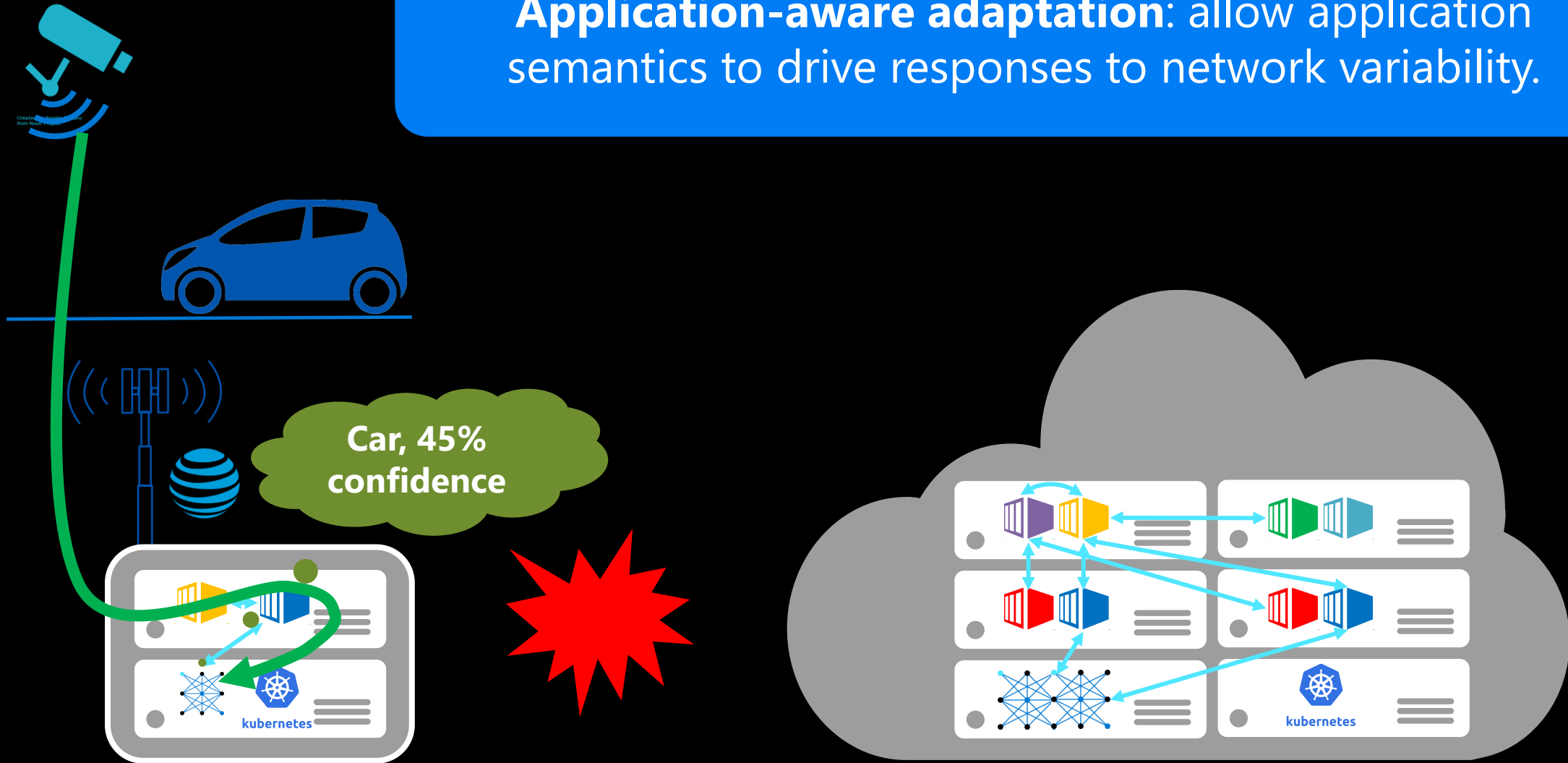
Variable wide-area network.







Application-aware adaptation: allow application semantics to drive responses to network variability.



BumbleBee: separate core logic from adaptation

Application logic



Encapsulate application logic
in unmodified containers

BumbleBee: separate core logic from adaptation

Application logic



Devs focus on functionality,
not environment

BumbleBee: separate core logic from adaptation

Application logic

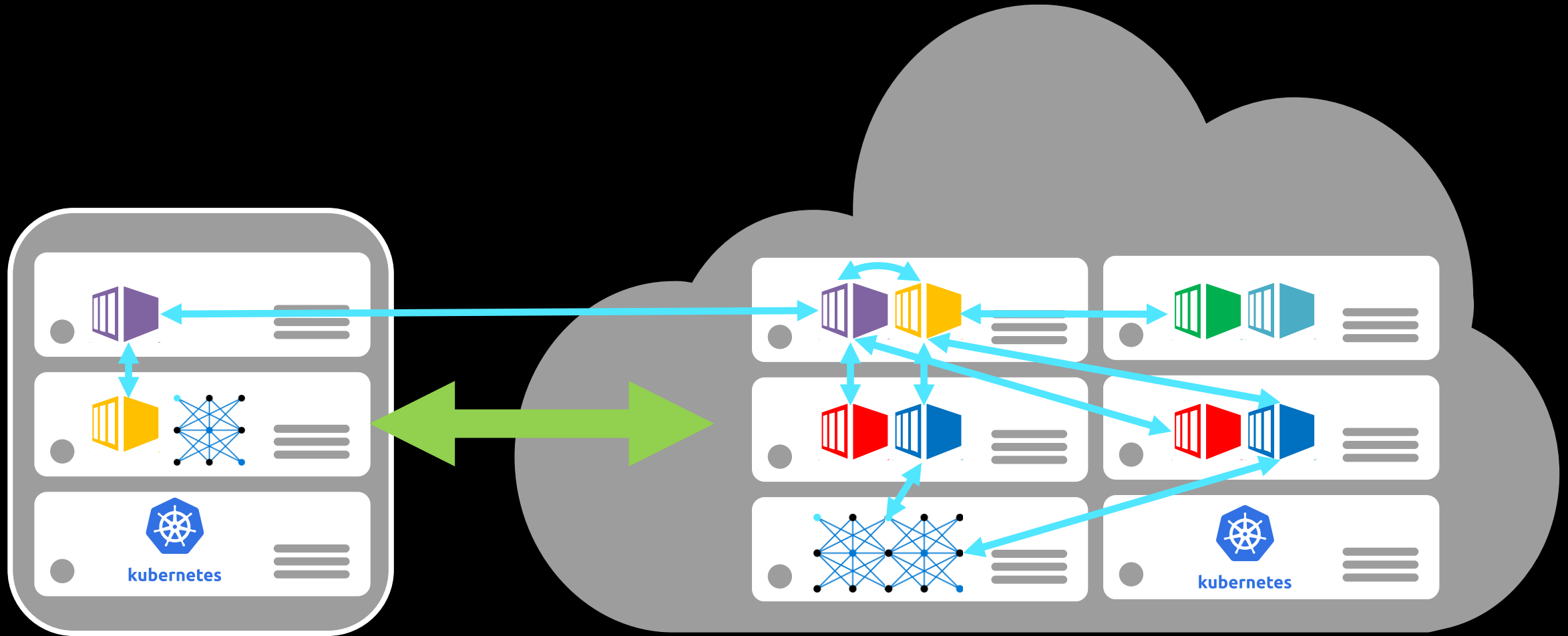


Adaptation strategies

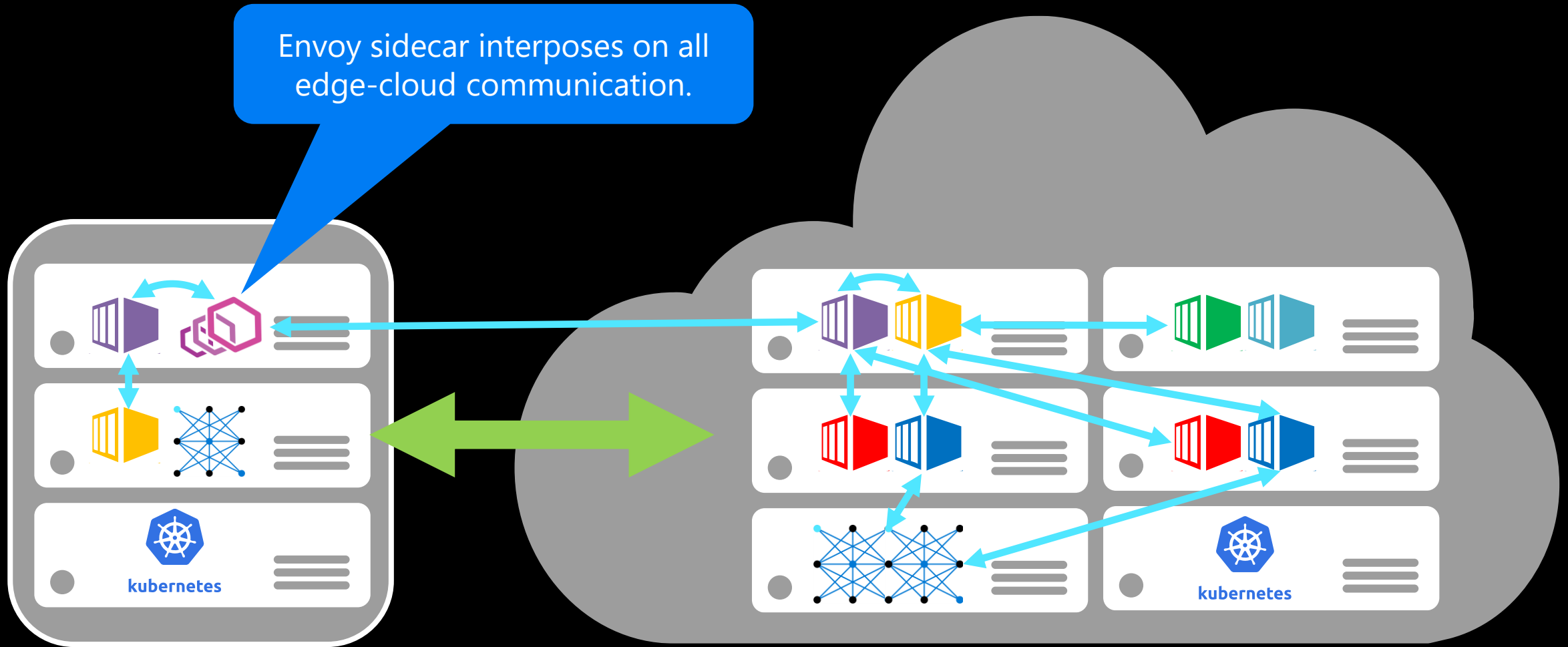


Transparent network proxy
(sidecar)

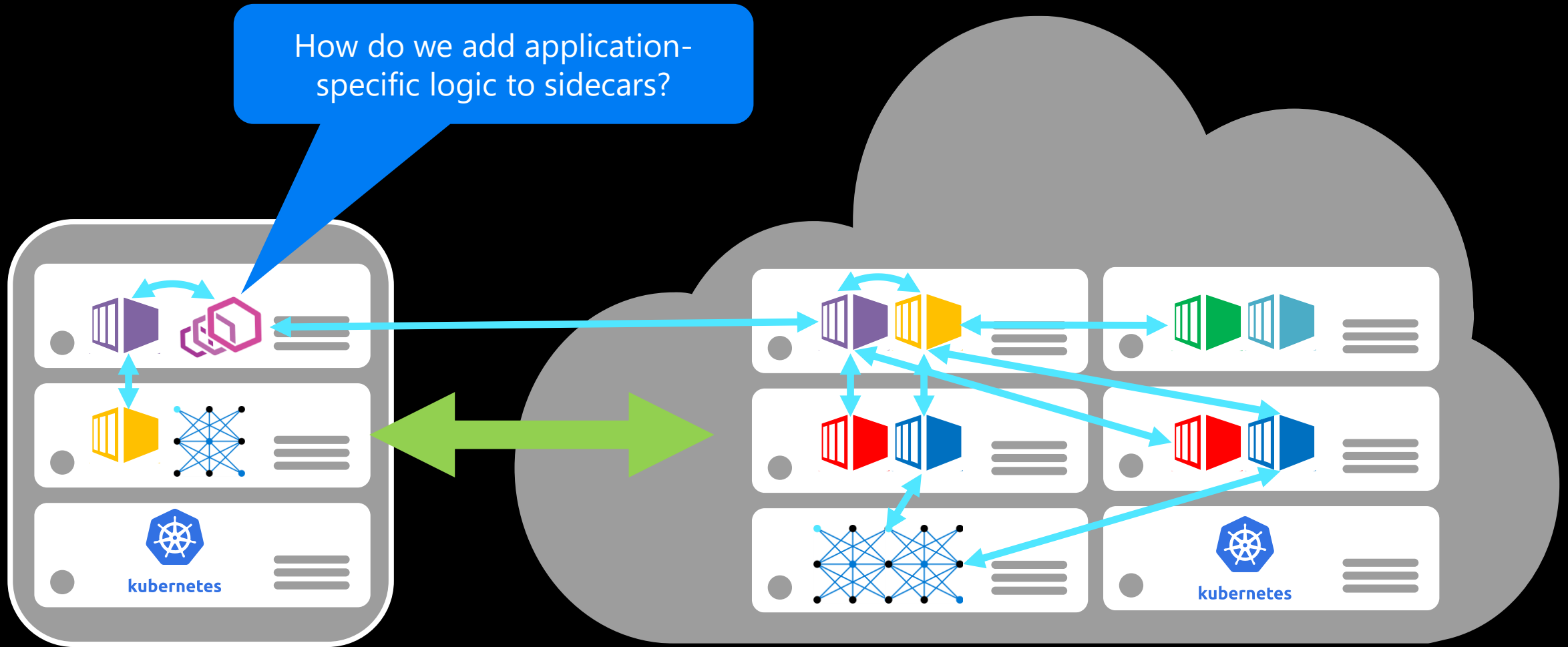
BumbleBee: application-aware adaptation



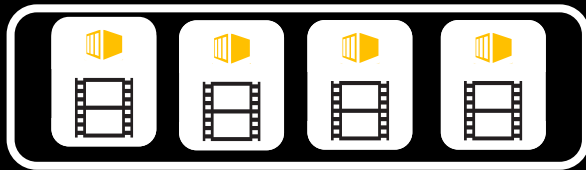
BumbleBee: application-aware adaptation



BumbleBee: application-aware adaptation



BumbleBee API



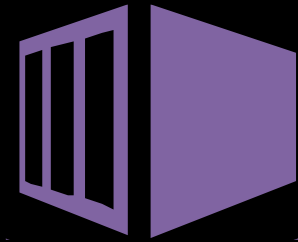
Message queues

Iterate
Enqueue
Dequeue
Bandwidth
Avg. latency



Messages

GetDelay
Drop
Reroute
MoveToFront
MoveToBack



Asynchronous callbacks

Message transform
Notify

API is exposed to short Lua scripts.

Car-counting script

```
function envoy_on_request(h)
  -- for each sink
  for queue in h:Queues():getQueue() do
    -- check the route
    route = queue:route()
    if string.find(route, "cloud") then
      -- check current bandwidth estimate
      bw = queue:getBW()
      if bw == 0 then
        -- if disconnected
        -- redirect request to the edge
        h:redirect("edge-detector")
      elseif bw < required then
        -- if bw is too low
        -- transform the request to low-res
        h:transform("180p")
      end
      if bw < required/2 then
        -- if bw drops well below required
        -- notify the request source
        h:notify(bw)
      end
    end
  end end end
```

Car-counting script

Access a message queue

```
function envoy_on_request(h)
  -- for each sink
  for queue in h:Queues():getQueue() do
    -- check the route
    route = queue:route()
    if string.find(route, "cloud") then
      -- check current bandwidth estimate
      bw = queue:getBW()
      if bw == 0 then
        -- if disconnected
        -- redirect request to the edge
        h:redirect("edge-detector")
      elseif bw < required then
        -- if bw is too low
        -- transform the request to low-res
        h:transform("180p")
      end
    end
    if bw < required/2 then
      -- if bw drops well below required
      -- notify the request source
      h:notify(bw)
    end
  end
end end end
```

Car-counting script

Redirect requests to edge
when bandwidth falls to zero

```
function envoy_on_request(h)
  -- for each sink
  for queue in h:Queues():getQueue() do
    -- check the route
    route = queue:route()
    if string.find(route, "cloud") then
      -- check current bandwidth estimate
      bw = queue:getBW()
      if bw == 0 then
        -- if disconnected
        -- redirect request to the edge
        h:redirect("edge-detector")
      elseif bw < required then
        -- if bw is too low
        -- transform the request to low-res
        h:transform("180p")
      end
      if bw < required/2 then
        -- if bw drops well below required
        -- notify the request source
        h:notify(bw)
      end
    end
  end end end
```

Car-counting script

Asynchronously transform frames when bw drops

```
function envoy_on_request(h)
  -- for each sink
  for queue in h:Queues():getQueue() do
    -- check the route
    route = queue:route()
    if string.find(route, "cloud") then
      -- check current bandwidth estimate
      bw = queue:getBW()
      if bw == 0 then
        -- if disconnected
        -- redirect request to the edge
        h:redirect("edge-detector")
      elseif bw < required then
        -- if bw is too low
        -- transform the request to low-res
        h:transform("180p")
      end
      if bw < required/2 then
        -- if bw drops well below required
        -- notify the request source
        h:notify(bw)
      end
    end
  end end end
```


Car-counting script

```
function envoy_on_request(h)
  -- for each sink
  for queue in h:Queues():getQueue() do
    -- check the route
    route = queue:route()
    if string.find(route, "cloud") then
      -- check current bandwidth estimate
      bw = queue:getBW()
      if bw == 0 then
        -- if disconnected
        -- redirect request to the edge
        h:redirect("edge-detector")
      elseif bw < required then
        -- if bw is too low
        -- transform the request to low-res
        h:transform("180p")
      end
      if bw < required/2 then
        -- if bw drops well below required
        -- notify the request source
        h:notify(bw)
      end
    end
  end end end
```

Asynchronously notify app
when bw significantly falls

Adaptation demo from MSBuild



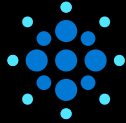
Azure for Operators

Your customer, your service, powered by our technology



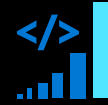
Achieve scale

Carrier grade network
functions as a service
powered by AI



Operate hybrid seamlessly

Consistent performance
and scalability across edge,
hybrid, and cloud



Monetize with new business models

Edge compute, IoT, 5G,
network slicing



Trust

Global, secure, and
compliant, empowers you
and your business

Built on a foundation of Telco DNA

Backed by Microsoft's developer ecosystem

Javelin overview



Video ML



Video delivery

Comprehensive video-processing platform
for analytics and distribution.



Network slicing



Unified Comms

Comprehensive network programmability
across CPaaS, RAN, and WAN.

I'm happy to answer questions.