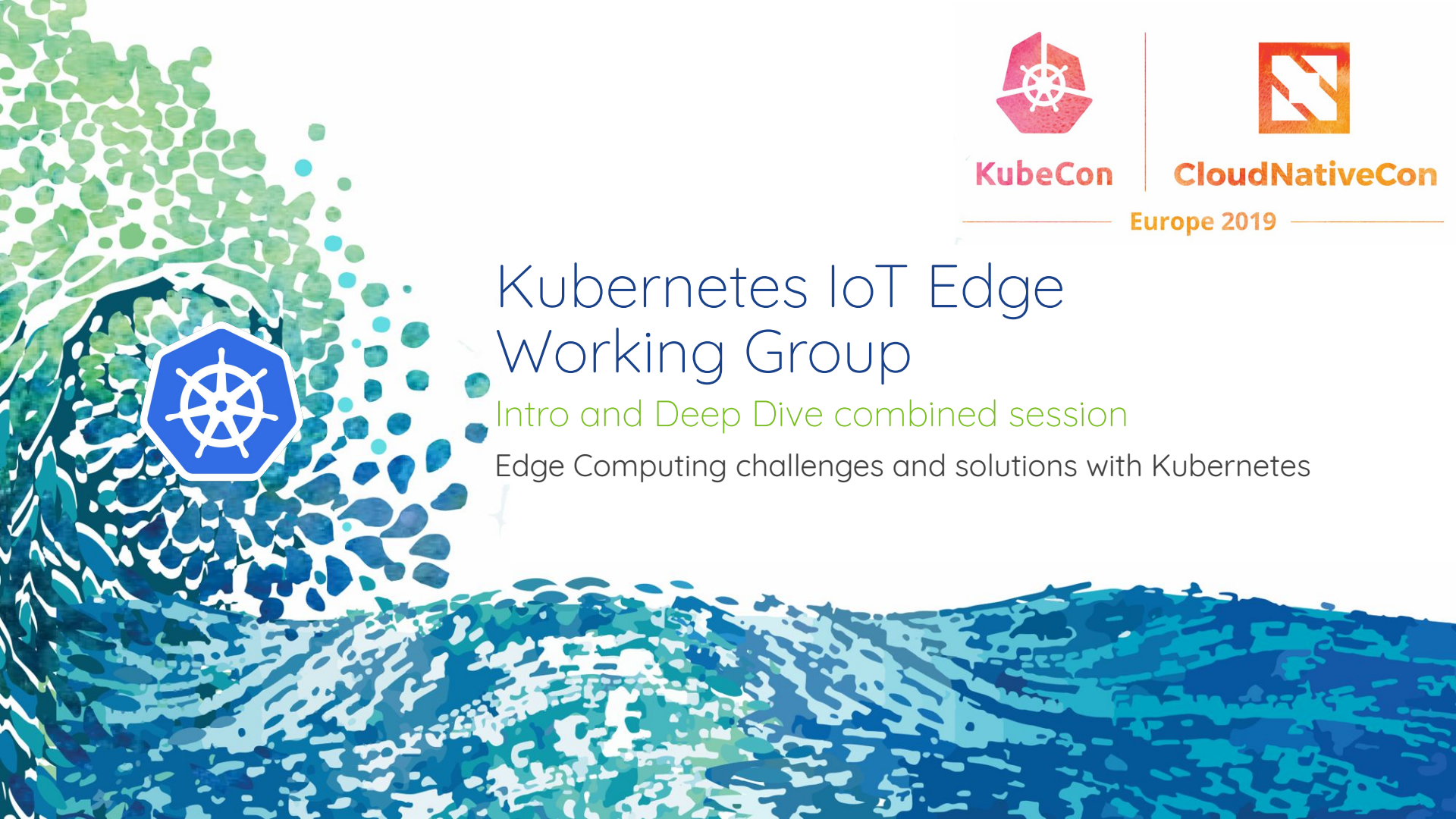


KubeCon | CloudNativeCon

Europe 2019

Kubernetes IoT Edge Working Group

Intro and Deep Dive combined session

Edge Computing challenges and solutions with Kubernetes

Abstract

Hidden slide during presentation – retain for published deck

This session will investigate and catalogue challenges encountered when Kubernetes is deployed in Edge and IoT applications.

We'll start by describing two basic approaches: deploying nodes to the Edge with a central control plane; and deploying whole clusters to the Edge.

This will be followed by a deep dive into Kubernetes architectural features and constraints in the context of both approaches. We'll see which course makes the most sense for some specific use cases.

Next we'll discuss some common challenges to successful deployments, such as resource limits and network availability, and provide some guidance on how to deal with them.

There are opportunities to contribute to the evolution of Kubernetes to better serve edge use cases. We will close with details on how you can get involved with the community effort to help this happen

Speakers

Cindy Xing
Futurewei
@cindyxing

Dejan Bosanac
Redhat
@dejanb

Steve Wong
VMware
@cantbewong

Kilton Hopkins
Edgeworx
@kiltonhopkins

Agenda

Approaches

1. Nodes to edge, with remote central control plane
2. Deploy whole clusters to the edge

Choosing an approach for some specific use cases

Dealing with some common challenges

- Resource limits
- Network Availability
- Data Plane communication – edge to cloud services
- Security

How you can get involved with community efforts

Use cases

Common requirements for edge workloads

1. lowest latency between data and responses and decisions
2. pre-processing (reduction) before data moves to cloud,
3. remotely managed datasets for local access
4. remotely manage software deployment and updates
5. operate offline or with intermittent connectivity

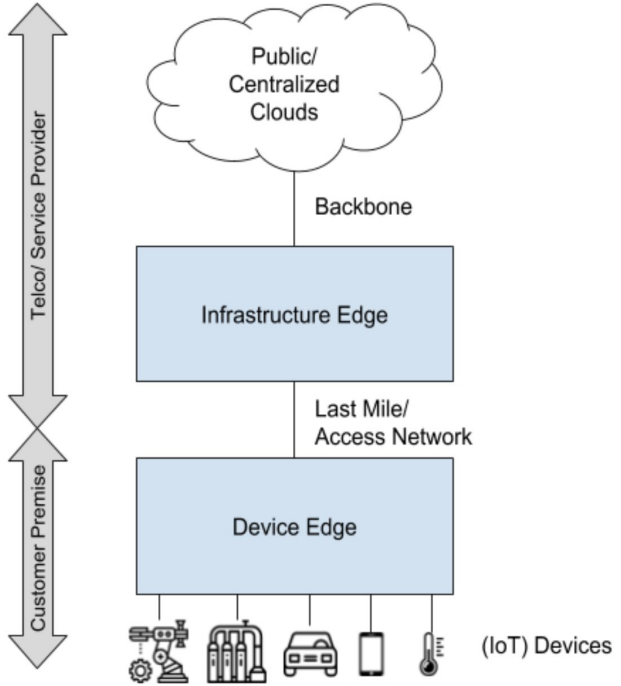
	1	2	3	4	5
Remote office, retail			✓	✓	✓
Sensor data collection, analytics	✓	✓		✓	✓
Physical device control	✓			✓	✓
Gaming	✓	✓	✓	✓	
Telco edge cloud	✓	✓	✓	✓	

Approaches

Edge types

- Infrastructure Edge
  - Deploy whole clusters on Edge sites
  - Hybrid-clouds
  - Federated clusters

- Device Edge
  - Deploy cluster nodes outside of the cloud

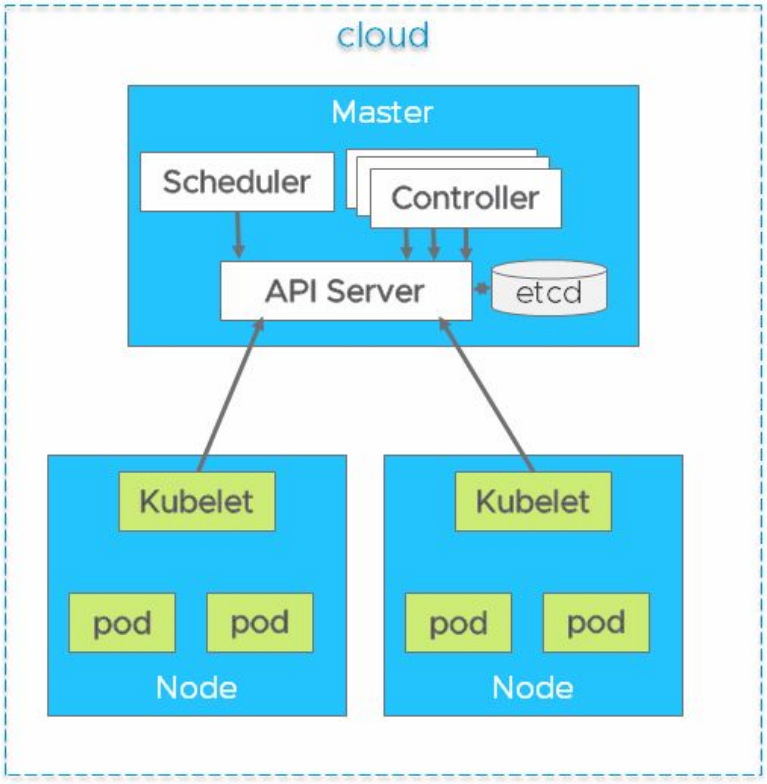


Source: Icons from https://www.flaticon.com/free-icon

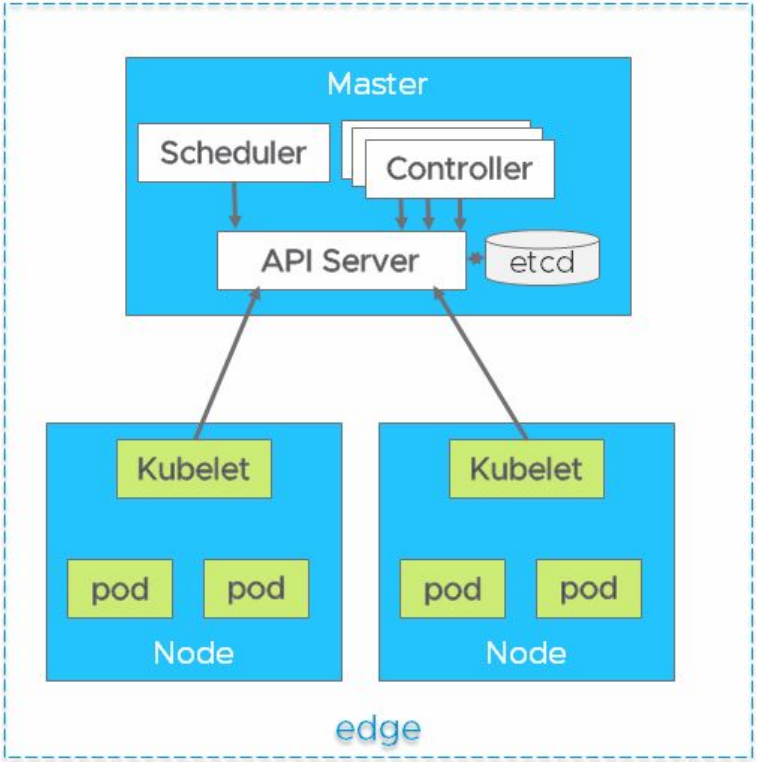
Available Approaches

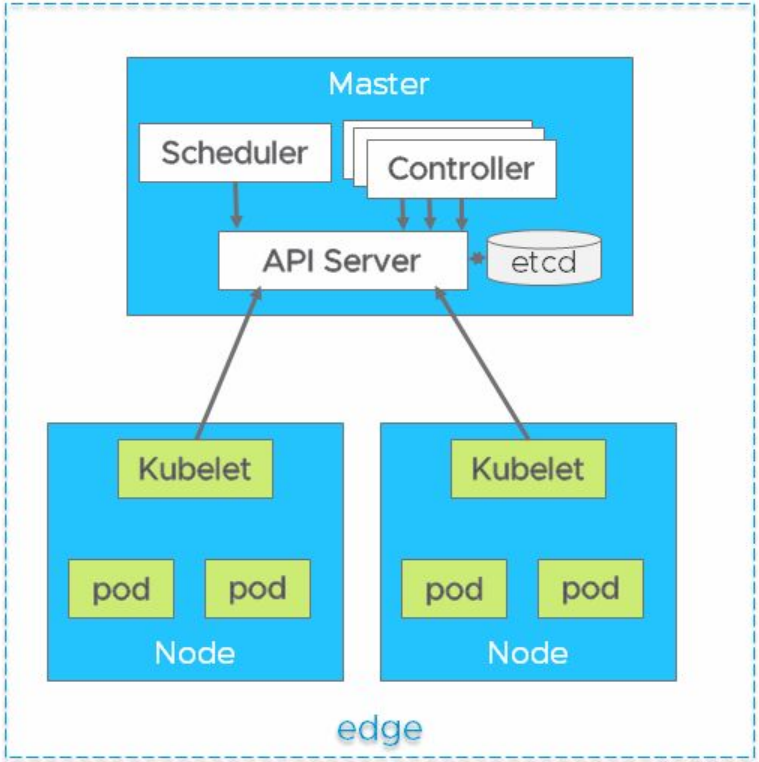
1. Install and manage a Kubernetes cluster at edge locations
    - Reference architecture: K3S
    - Cluster at Edge
2. Manage edge nodes from cloud
    - Reference architecture: KubeEdge
    - Worker Node at Edge
3. Hierarchical cloud + edge
    - Reference architecture: Virtual Kubelet
    - Cluster at Edge and manage from Cloud

Standard Kubernetes Architecture

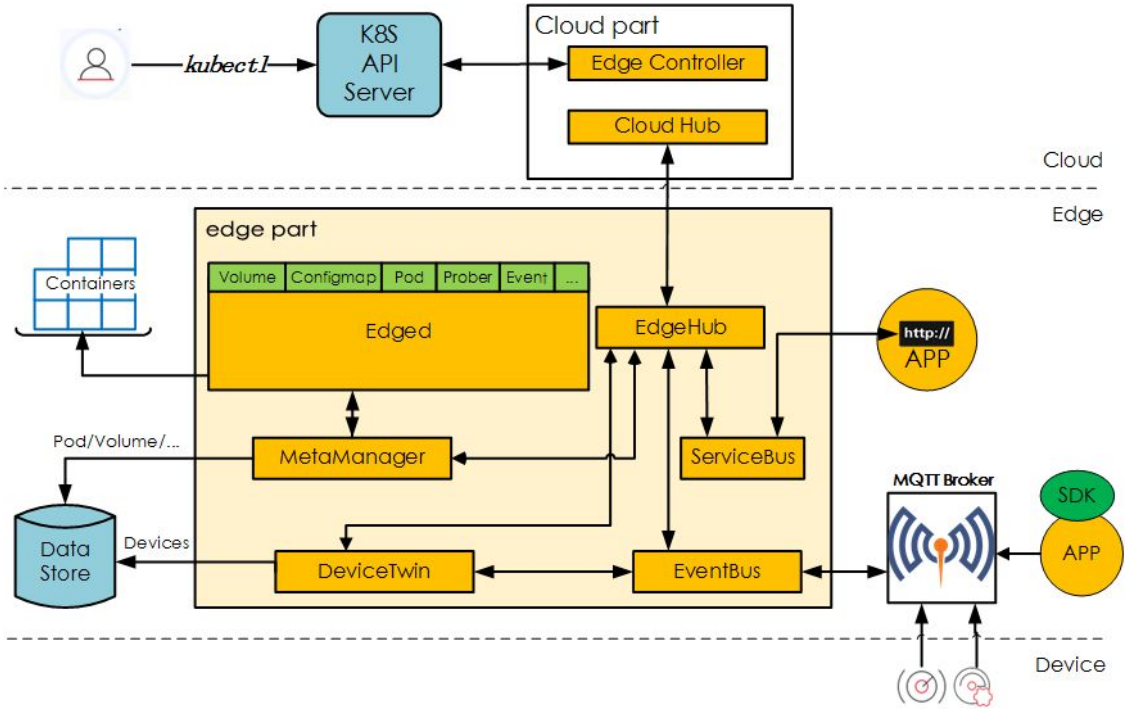


Option 1: whole clusters at edge

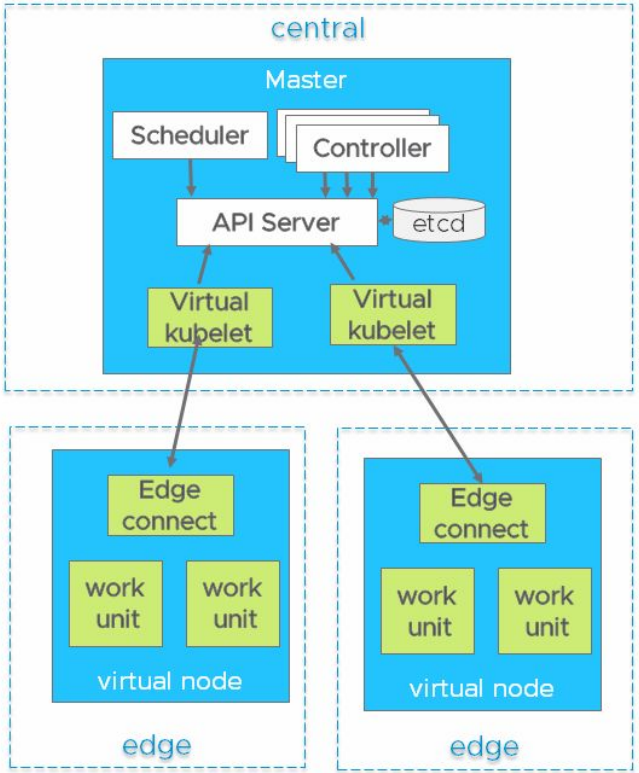




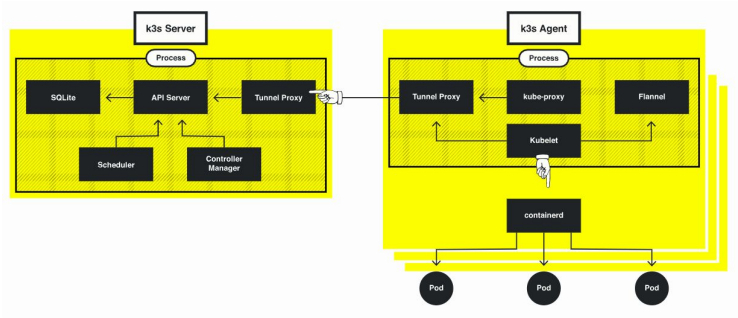
Option 2: central control managing edge

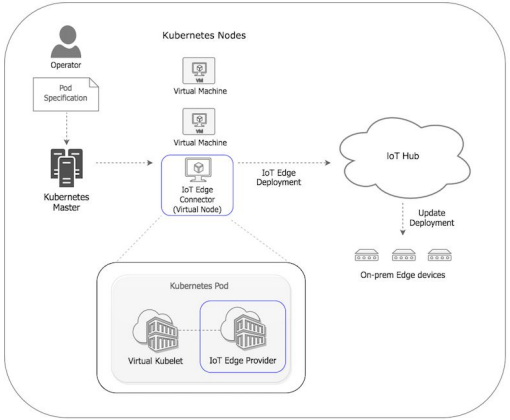


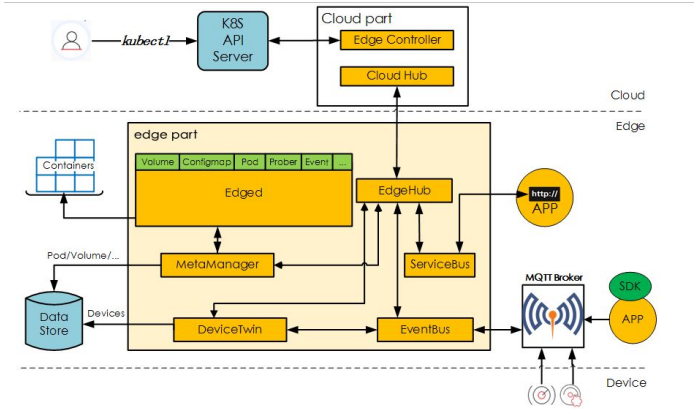
Option 3: hierarchical edge architecture



Comparisons







Area	K3S	KubeEdge	Virtual Kubelet
EdgeNode management	Yes	Yes	Indirect
App. Deployment & Orchestration	Yes	Yes	Yes
Device Management	No	Yes	No
EdgeNode registration	From client	From cloud	From cloud
Master Location	Edge	Cloud	Cloud and Edge
Pure K8s Native	Yes	Yes	With external provider
Extensibility	No	Yes	Yes
Module pluggable	No	Yes	Yes
Lightweight	Yes	Yes	No

Challenges

Edge Challenges

- Infrastructure
  - How to manage resources (nodes and clusters) on the Edge?
- Control plane
  - How to manage workloads on the Edge?
- Data plane
  - How Edge sites communicate with the cloud and between themselves?

Issues Unique to Edge

- Resource constraints
- Network limitations
- Unattended operation
- Physical security

Network

Challenges facing and not necessary having solutions

- Network bandwidth and reliability
- Connectivity
  - Between edge and cloud
  - Between edge nodes
- Network routing
  - North – South
  - East – West
- Discovery
- Network policy and access control
- K8s flat network requirement

Deep dive

Edge challenges - recap

Resources

- Limited number of nodes on the Edge
- No “bursting” to newly provisioned capacity like a public cloud or large datacenter
- Workloads typically have a wide range of priorities
- Need more emphasis on prioritization, triage

Network

- Network capacity can be limited, and variable
- Like resources, different workloads can have different network policies/priorities

A small mistake, big consequences

During Admission, this Pod might be

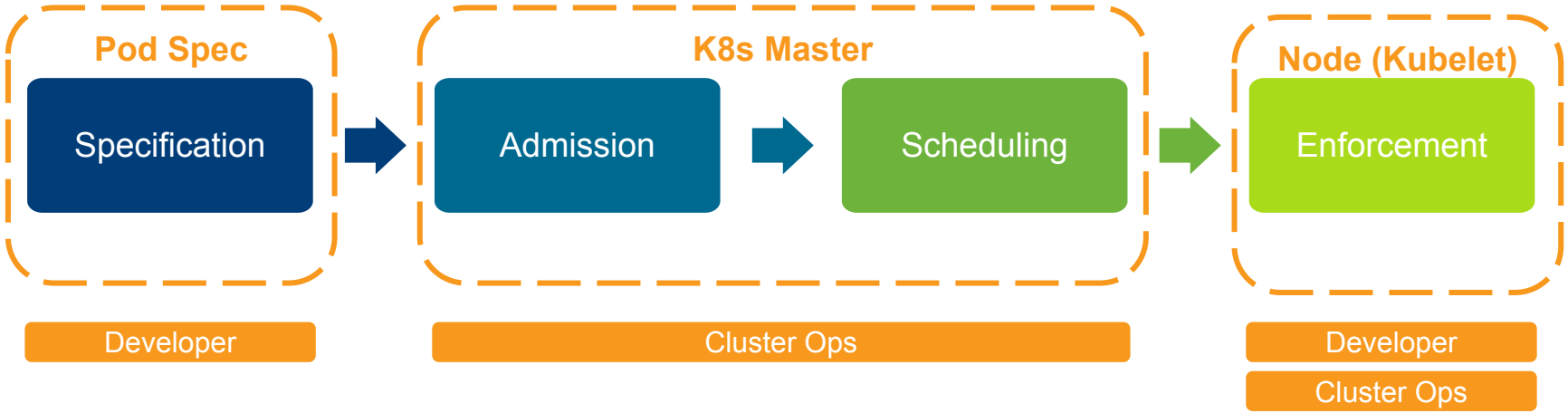
- Rejected (ResourceQuota)
- Modified (LimitRanger)

After Creation, this Pod might

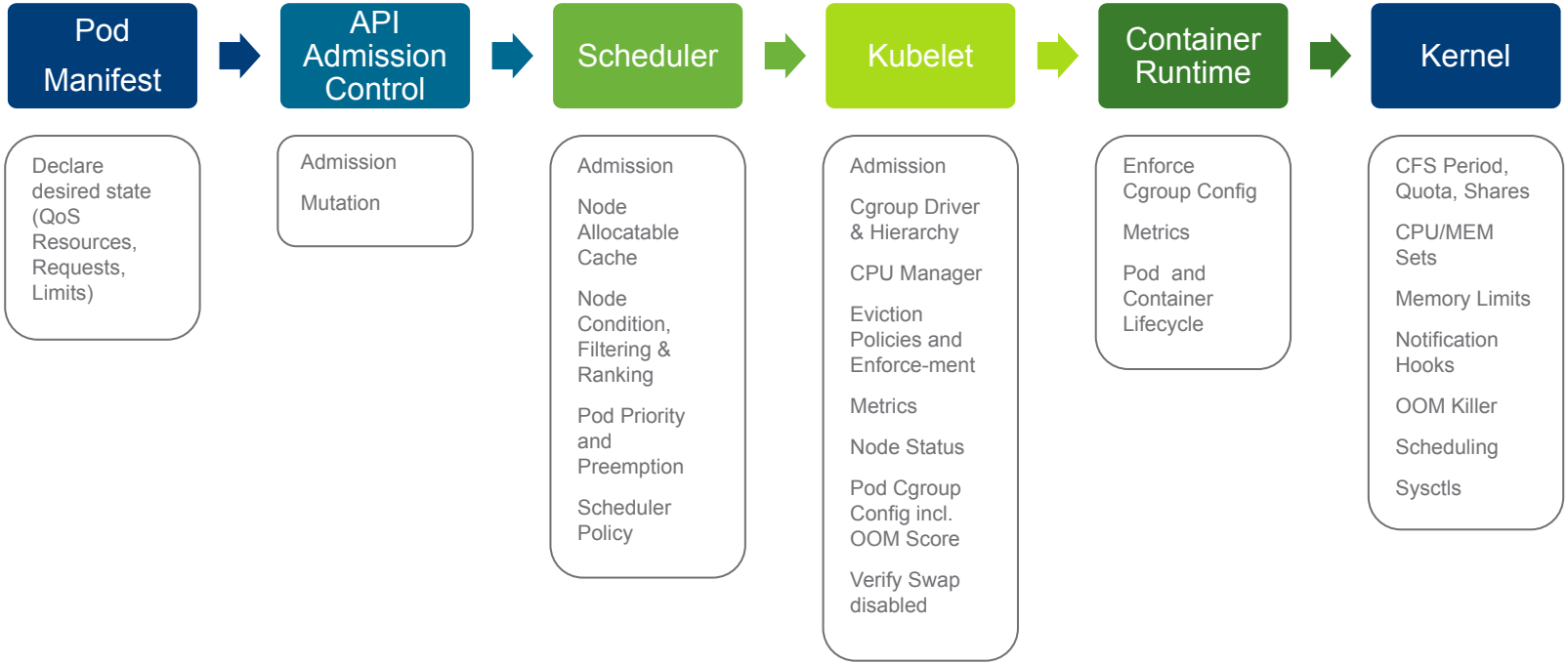
- Not get enough Resources (“Starvation”)
- Negatively affect other Pods or Host Services (“Noisy Neighbor”)
- Be evicted first by the Kubelet
- Be OOM_killed first (OutOfMemory)

QoS Lifecycle, Admission and Enforcement

30k Feet View



Deeper Dive into Admission & Enforcement



This session by Michael Gasch (KubeCon 2018) highly recommended for more https://sched.co/DqvA

Full specs always better

But essential for edge & IoT

Pods priority and pre-emption

- indicates the importance of a Pod relative to other Pods
- If a Pod cannot be scheduled, the scheduler tries to preempt (evict) lower priority Pods
- also affects scheduling order of Pods and out-of-resource eviction ordering
- No specification = globalDefault (or 0)

Quality of Service (QOS)

- Class determined by resource spec or lack of one
  - Guaranteed - every container in Pod must has memory and cpu limit(& request)
  - Burstable - a container in in pod specifies something
  - BestEffort - no resource spec
- Determines which Pod gets killed first when out of resource
- Note pre-emption is not identical rule set used for eviction

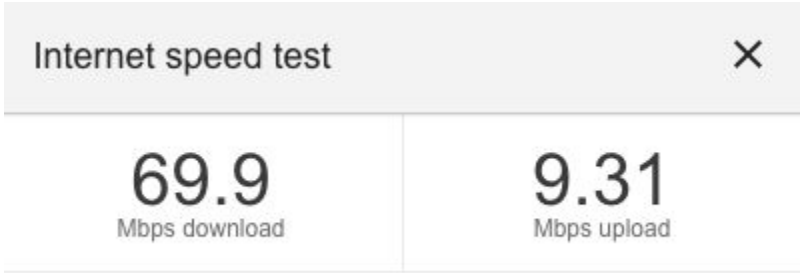
Why “traffic shaping” is needed

Network capacity is a limited resource

- Performance can be asymmetrical + variable

Different workloads may have different priorities or behaviors

- should have different network policies



Network Policy

Technically optional - but at edge, perhaps not

NetworkPolicy resource creation

- Deals with what traffic is allowed
- CNI Network Plugin specific
- Based on 'cluster-external' IPs
- Based on SRC/DST and port
- src/dst can be specified several ways
- May be subject to cluster environment

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: test-network-policy
  namespace: default
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  - Egress
  ingress:
  - from:
    - ipBlock:
        cidr: 172.17.0.0/16
        except:
        - 172.17.1.0/24
    - namespaceSelector:
        matchLabels:
          project: myproject
    - podSelector:
```

Bandwidth plugin

Implemented at a number of layers

- Does not work at a cluster wide view

Pod: bandwidth annotations

CNI: Bandwidth Plugin

`tc` (Traffic Control)

Linux: Network Namespace

Example bandwidth Pod spec

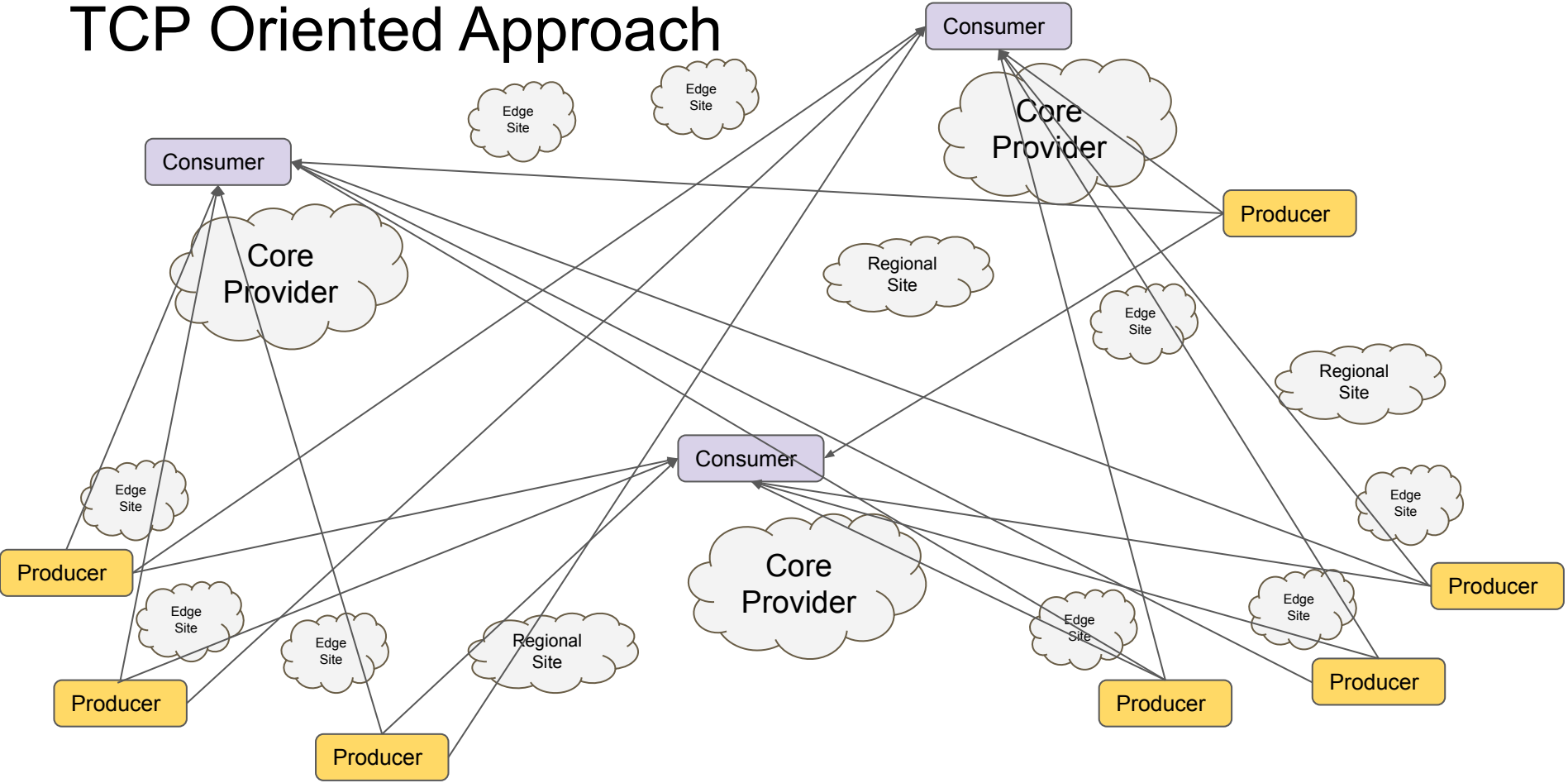

```
{
    "kind": "Pod",
    "metadata": {
        "name": "iperf-slow",
        "annotations": {
            "kubernetes.io/ingress-bandwidth": "10M",
            "kubernetes.io/egress-bandwidth": "10M"
        }
    }
}
```

Audience Poll / Break

Use cases
Remote office, retail
Sensor data collection, analytics
Physical device control
Gaming
Telco edge cloud

Data plane

TCP Oriented Approach

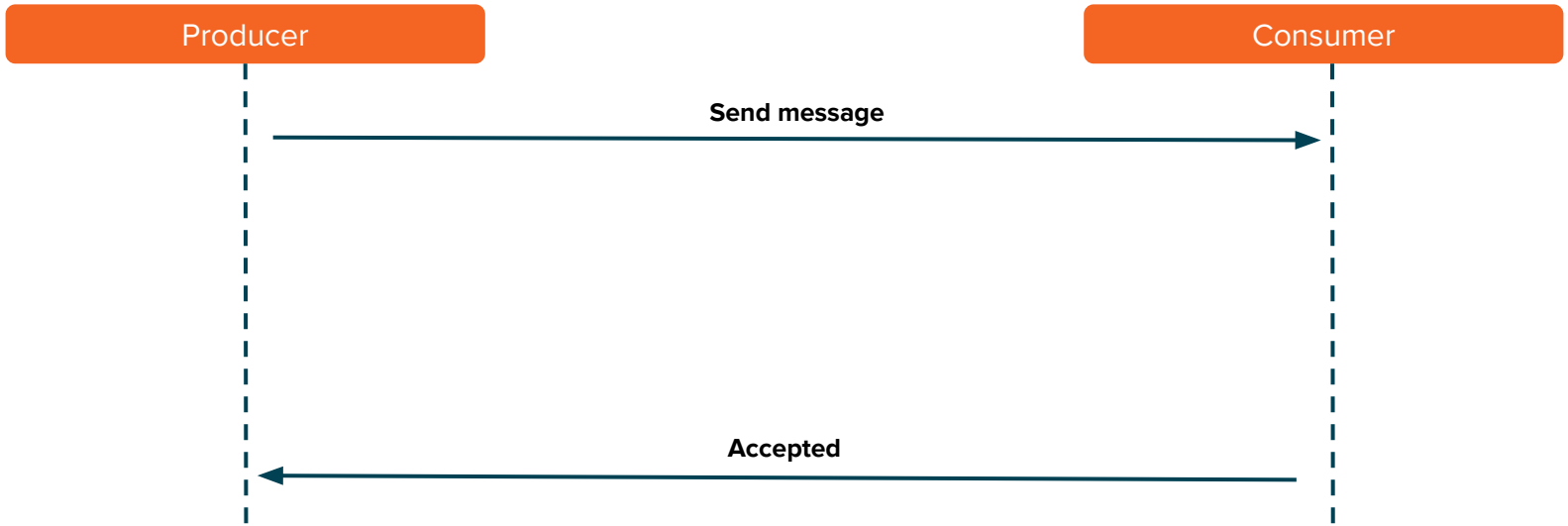


AMQP 1.0 Features

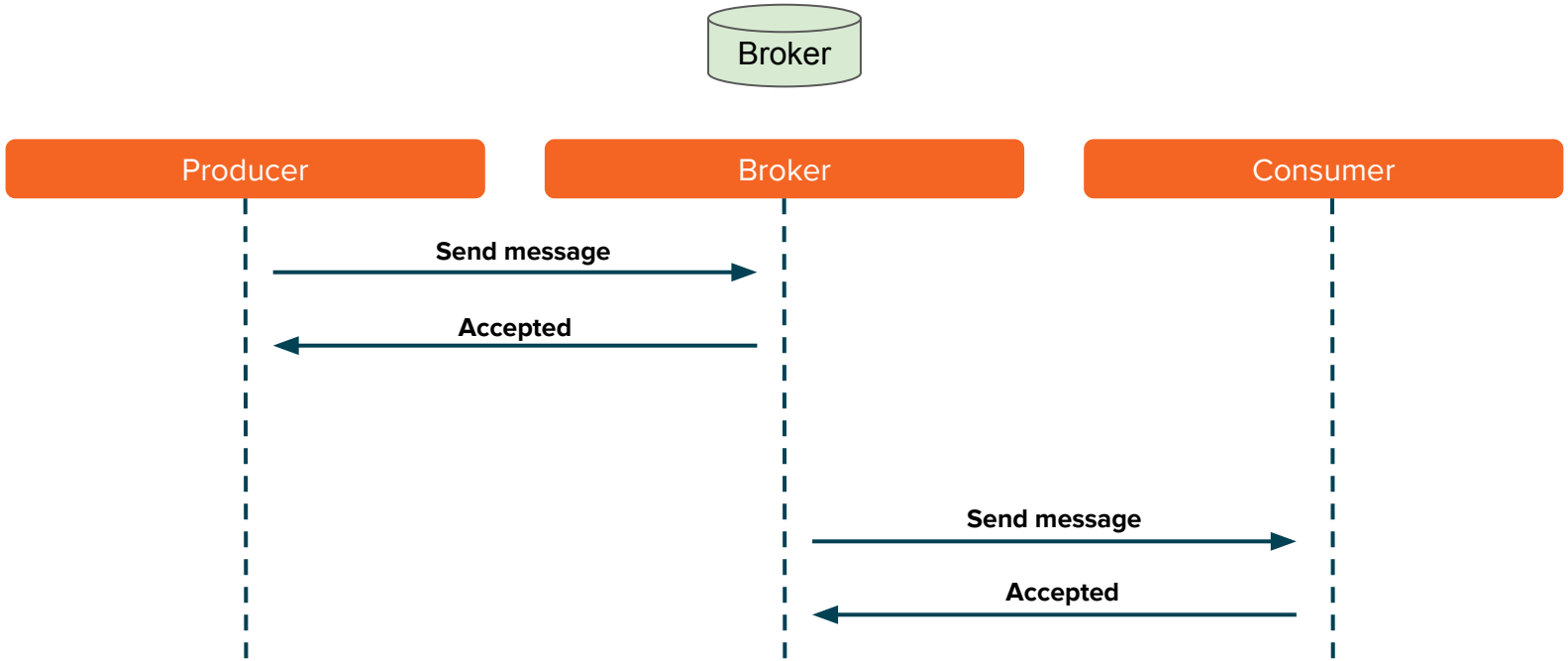
- Middleware: application level protocol
  - Not O.S. dependent (aside from TCP)
- Support for common Messaging Patterns:
  - Request/Response (RPC)
  - Fan Out (Pub/Sub, Topics)
  - Queuing (Store and Forward)
- Strict Flow Control
- Peer to Peer protocol
  - Intermediaries NOT required (but allowed)
- Application defined Addressing (Layer 7)
  - Separate from Network Addressing
  - Simple UTF-8 Strings

Routing vs Brokering

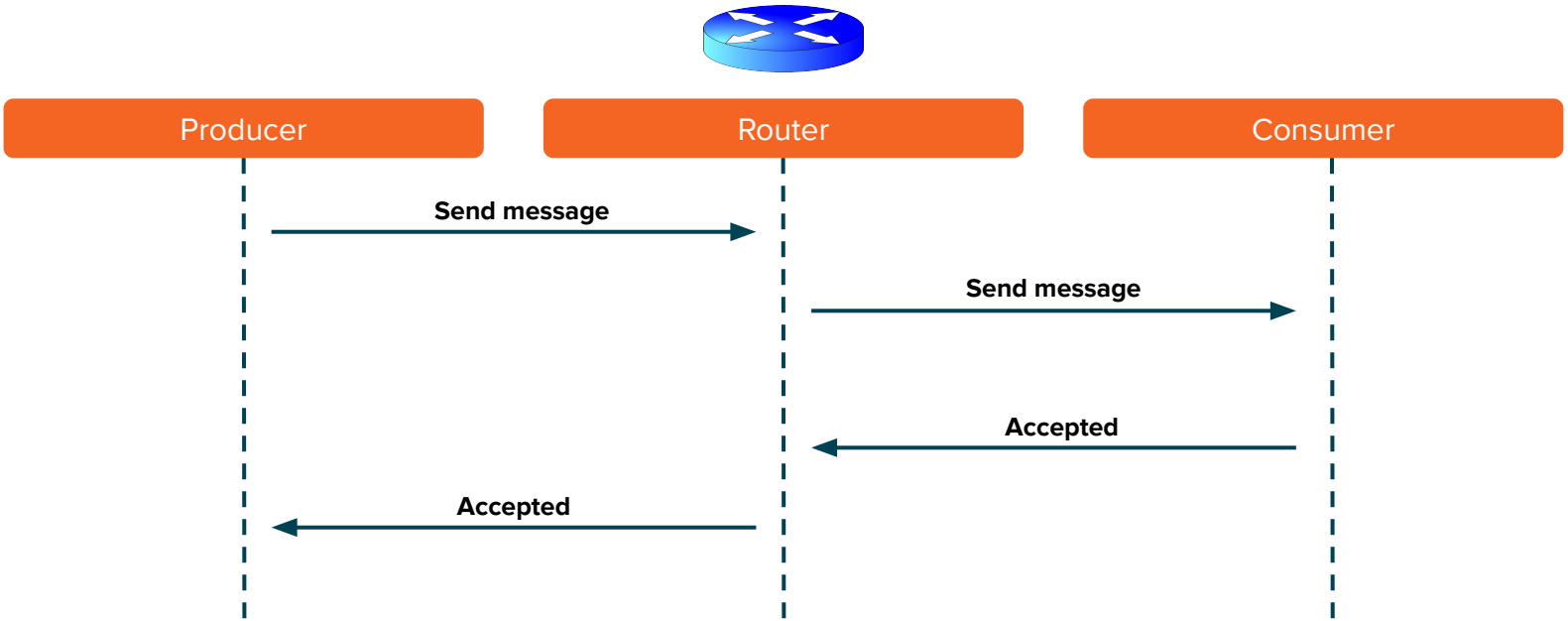
Direct



Routing vs Brokering



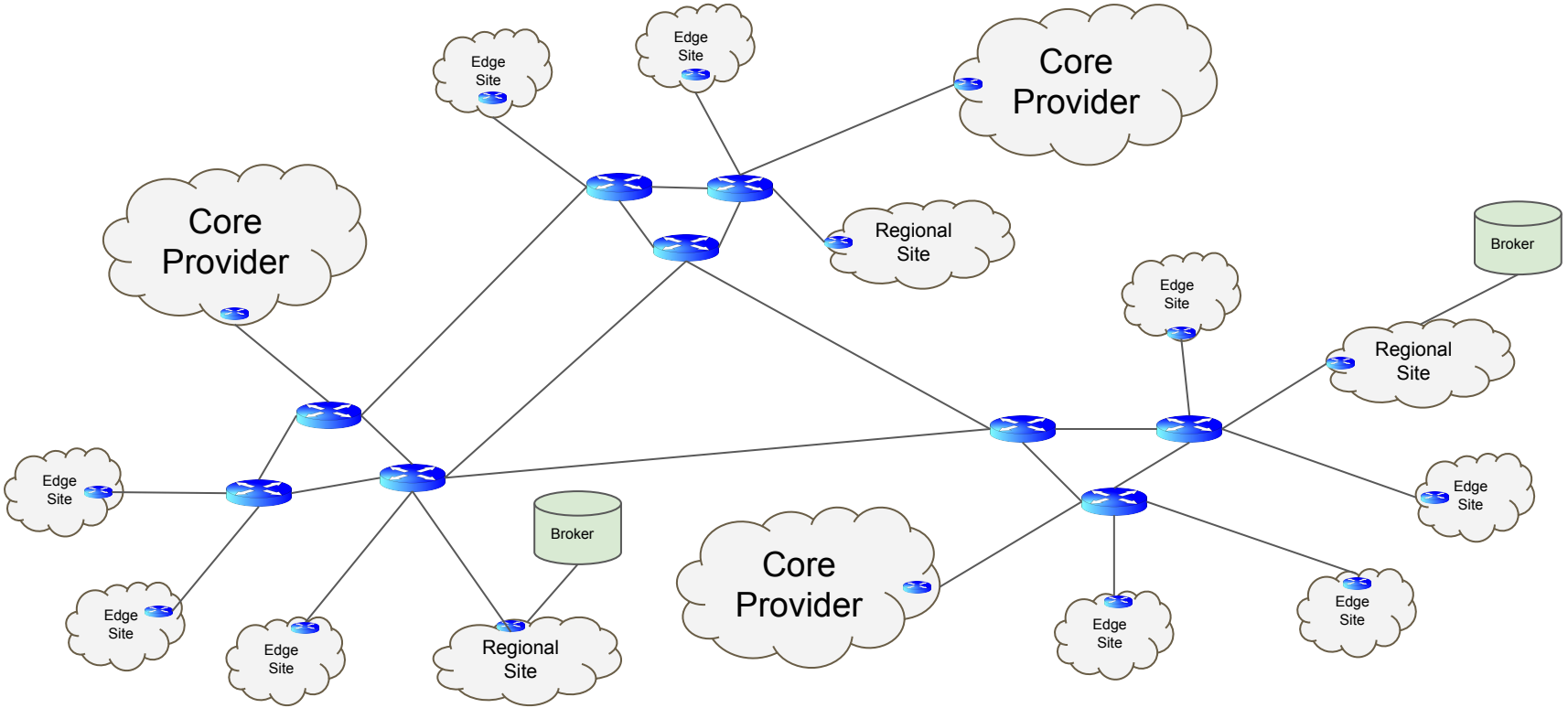
Routing vs Brokering



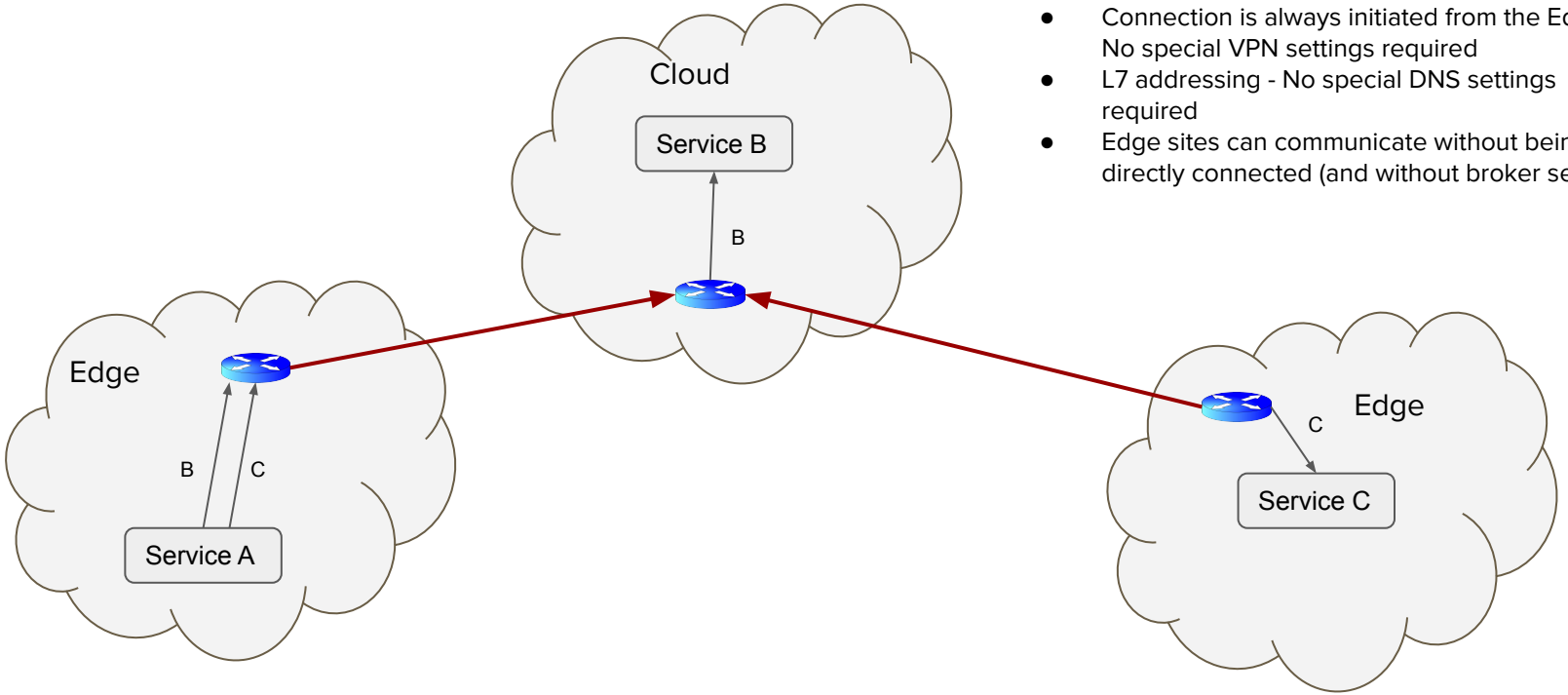
AMQP 1.0 Infrastructure Components

- Apache ActiveMQ Artemis (http://activemq.apache.org/artemis/)
  - Classical Broker provides queuing and pub/sub services
- Apache Qpid-Dispatch-Router (https://qpid.apache.org/components/dispatch-router/)
  - Stateless “bump in the wire” AMQP 1.0 message router (no message queueing)
  - Dynamically learns addresses of messaging endpoints
- Apache Qpid-Proton (https://qpid.apache.org/proton/)
  - High performance messaging library
  - Go, java, c++, python, ruby

Router Based Multi Region Message Bus

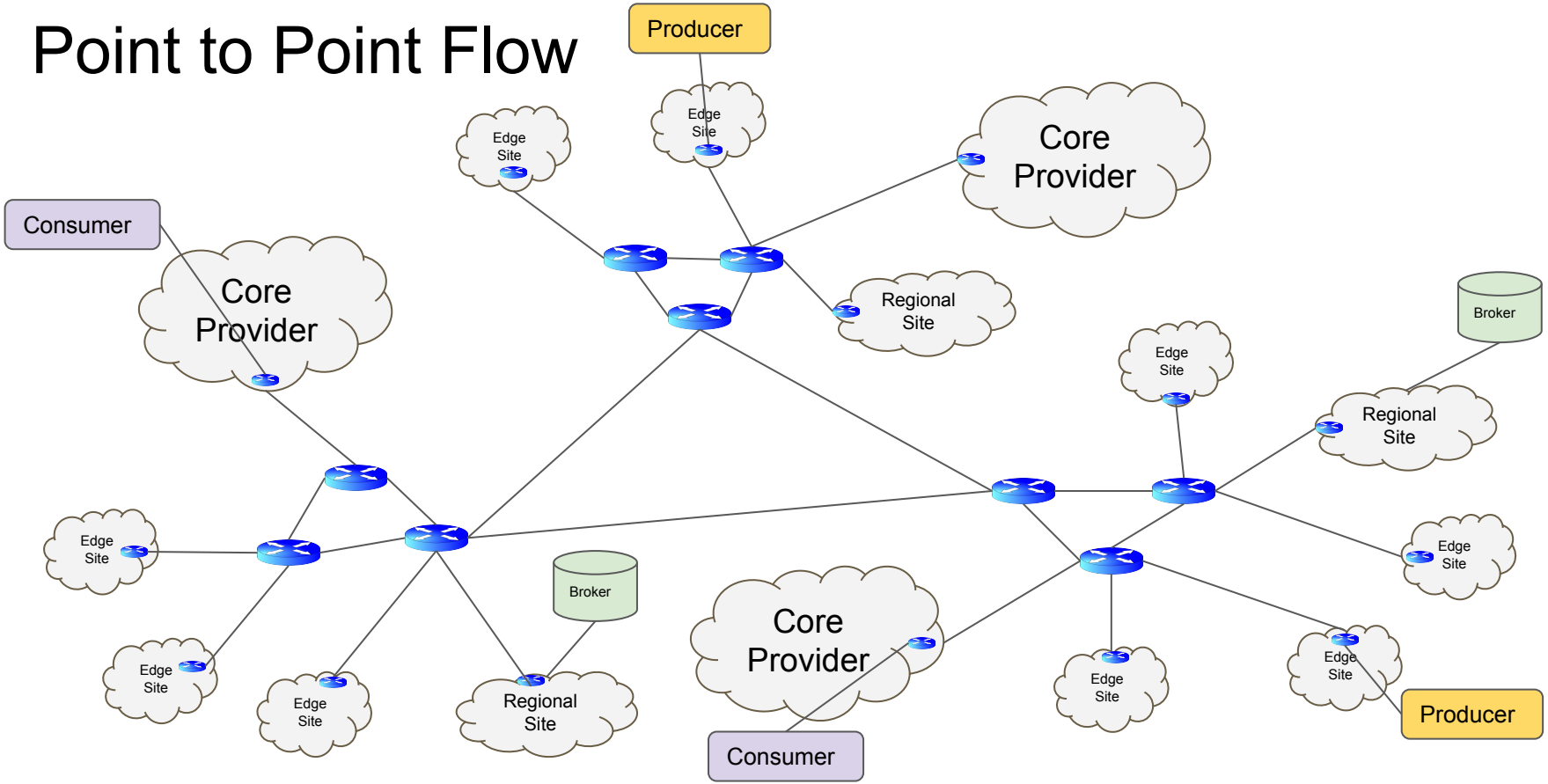


Secure multi-site communication

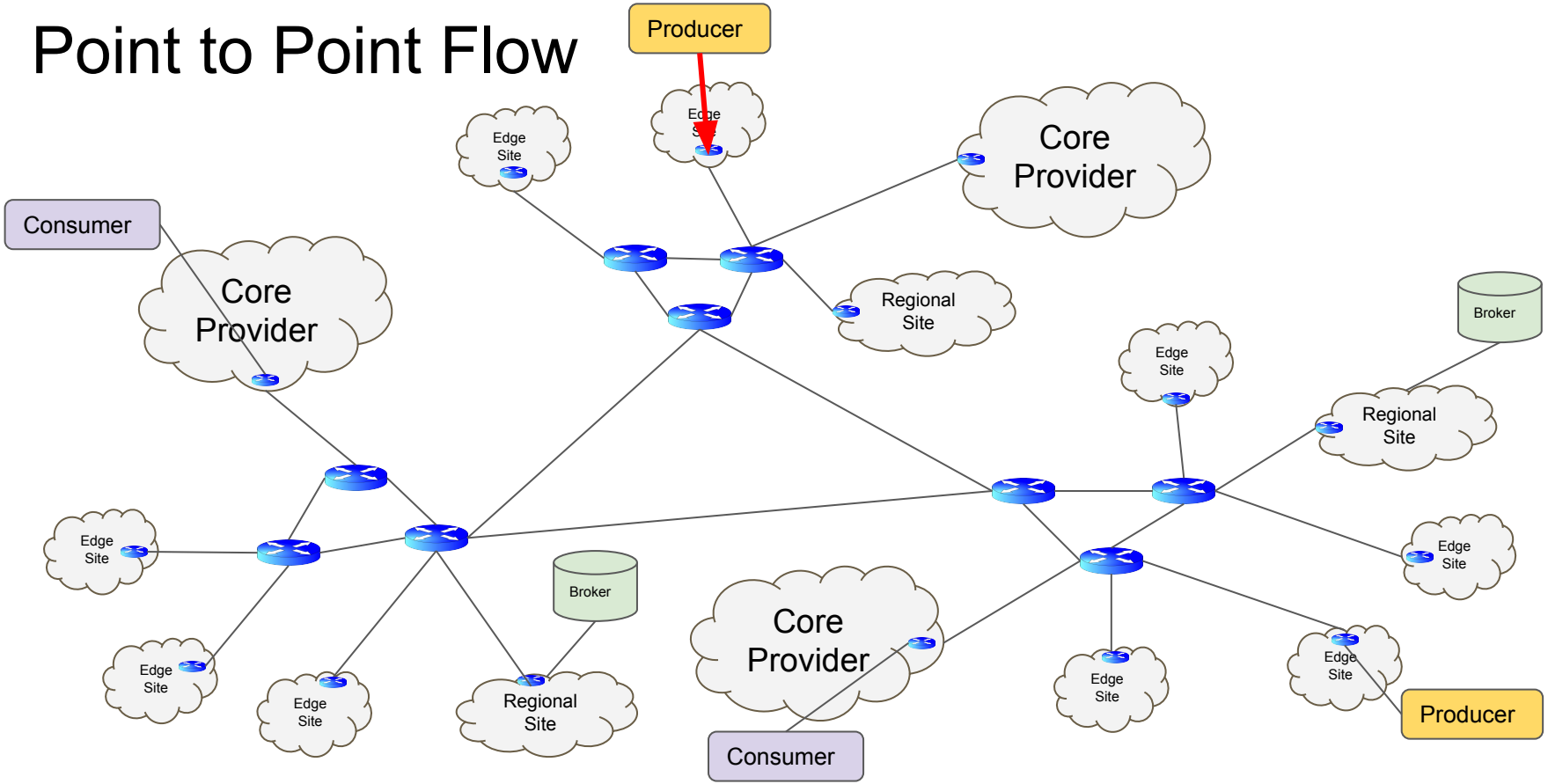


- Connection is always initiated from the Edge - No special VPN settings required
- L7 addressing - No special DNS settings required
- Edge sites can communicate without being directly connected (and without broker service)

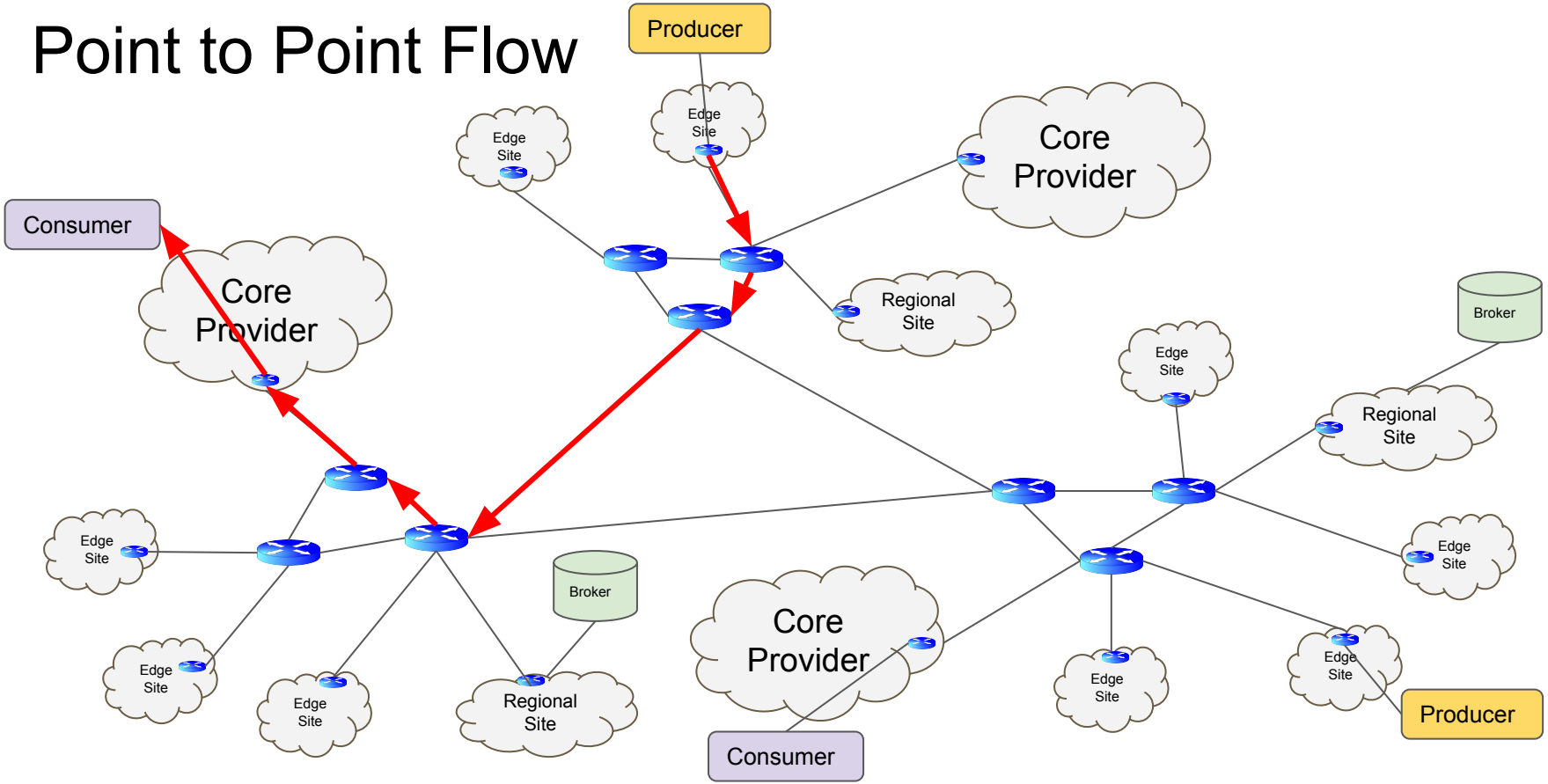
Point to Point Flow



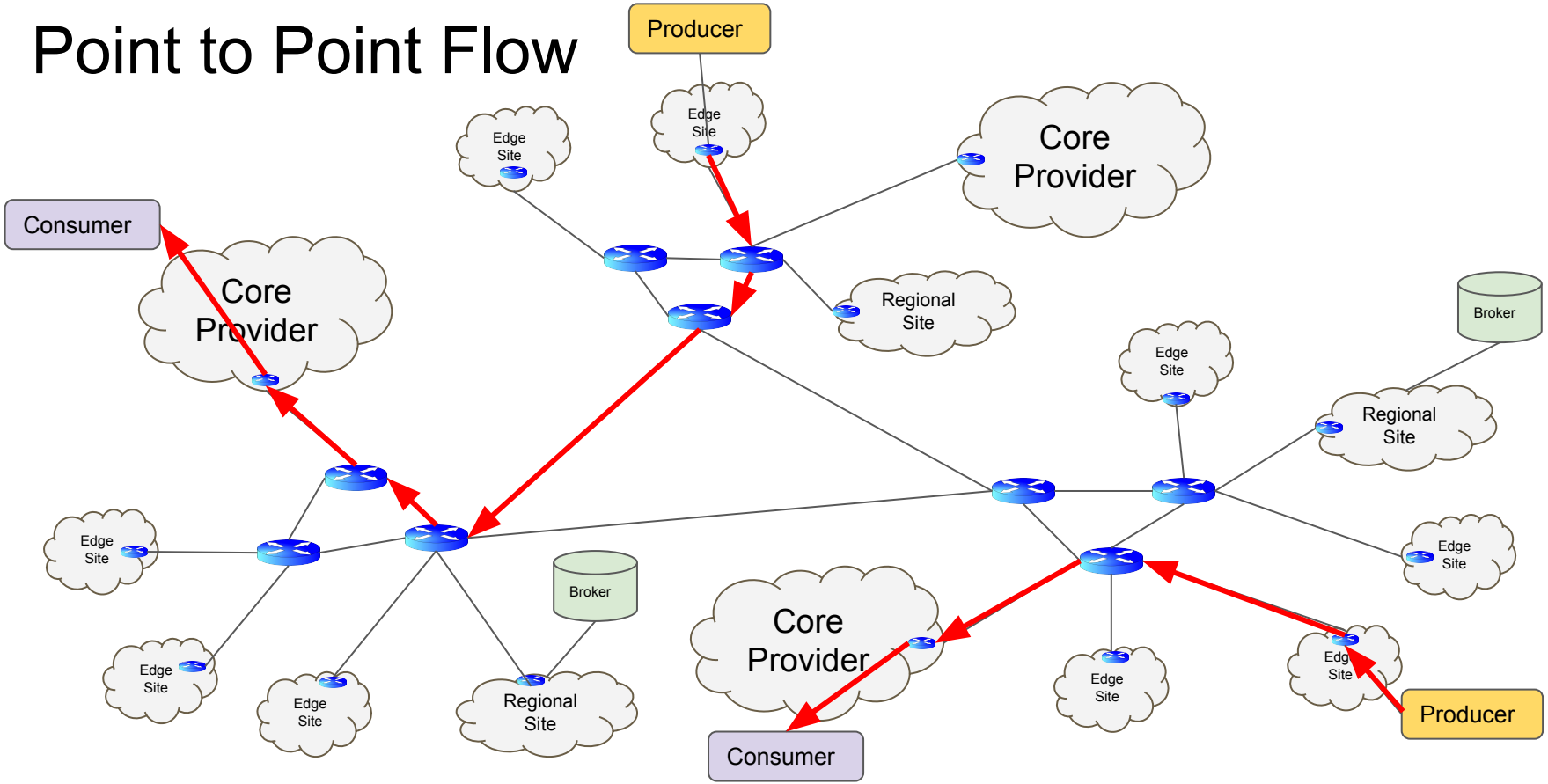
Point to Point Flow



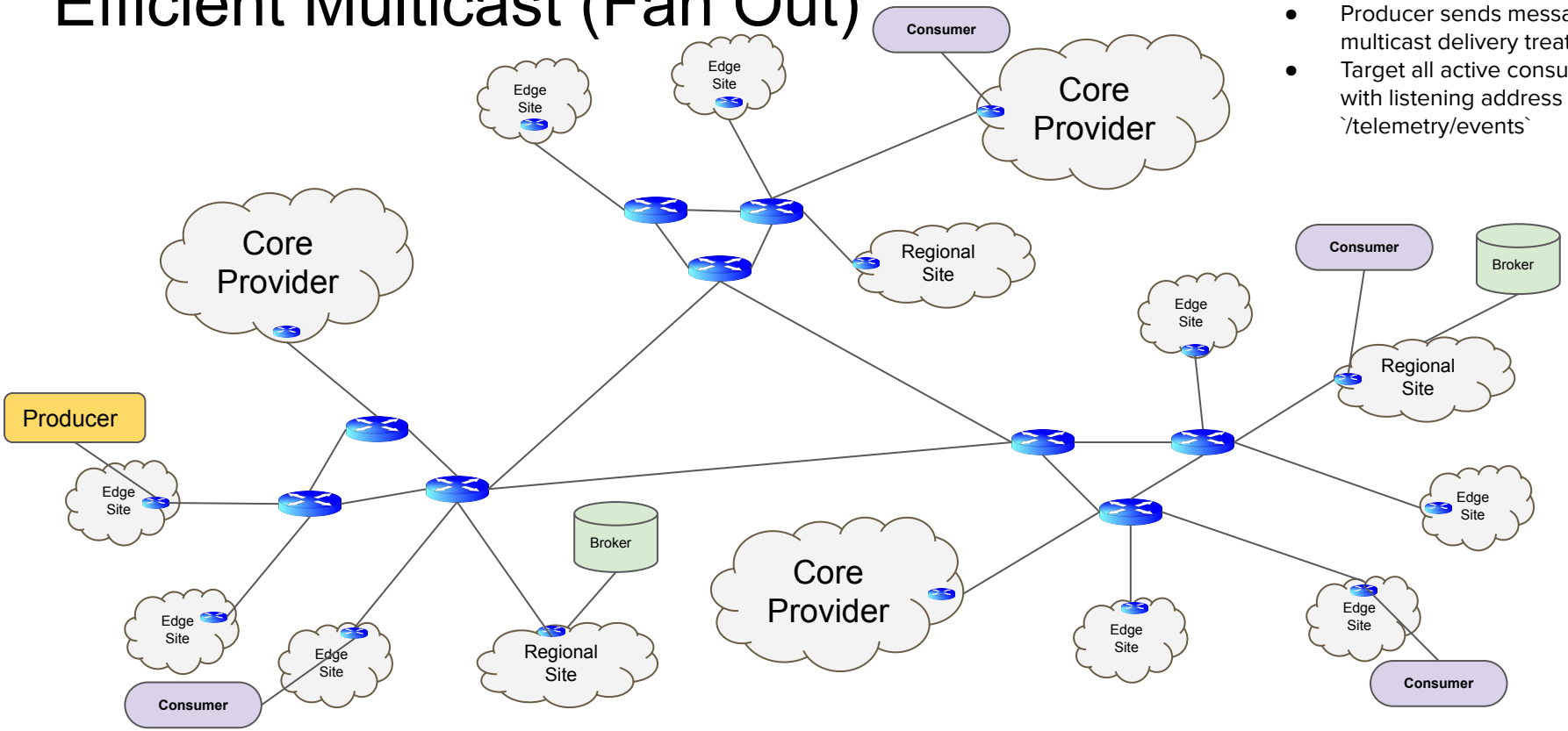
Point to Point Flow



Point to Point Flow

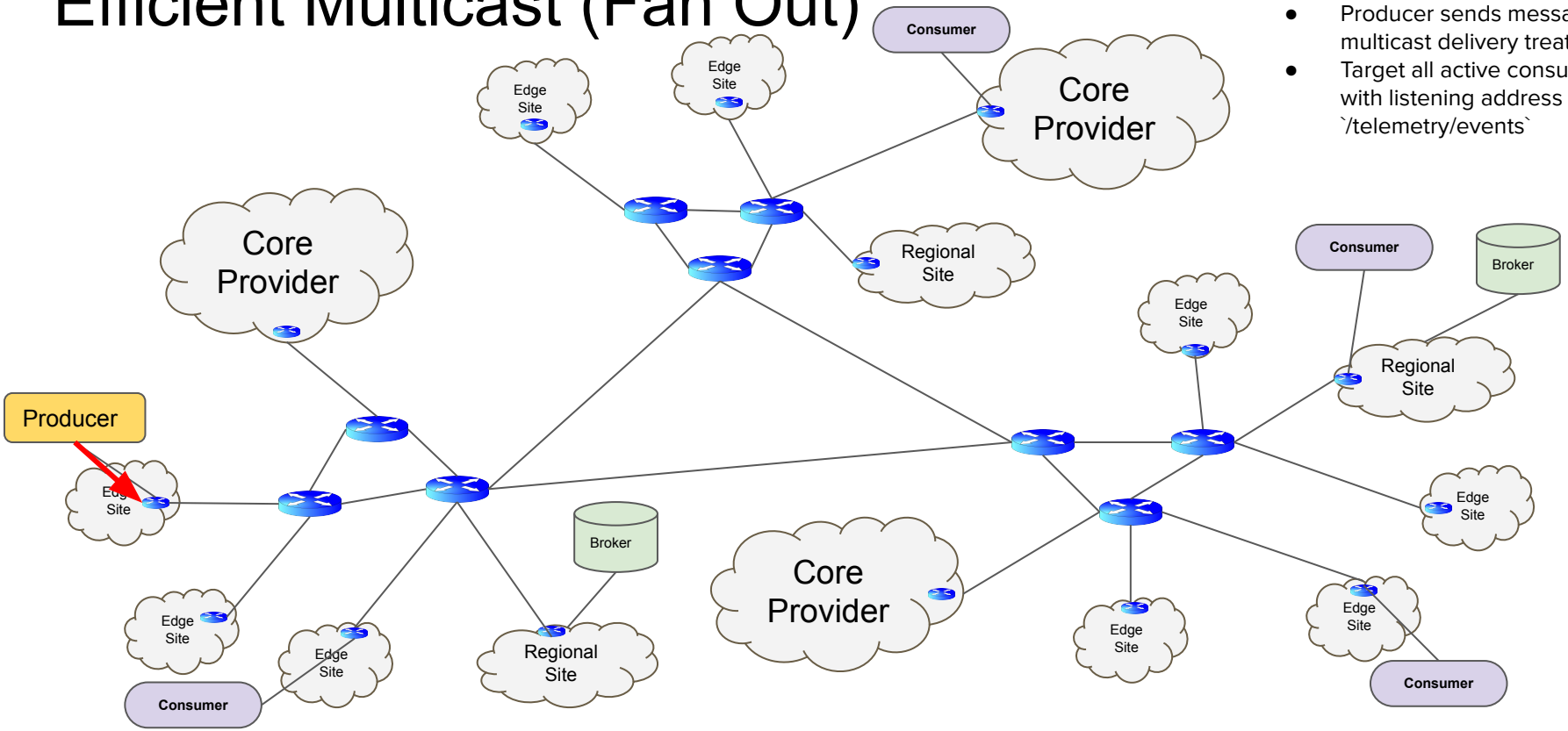


Efficient Multicast (Fan Out)



- Producer sends message with multicast delivery treatment
- Target all active consumers with listening address (e.g. `/telemetry/events`

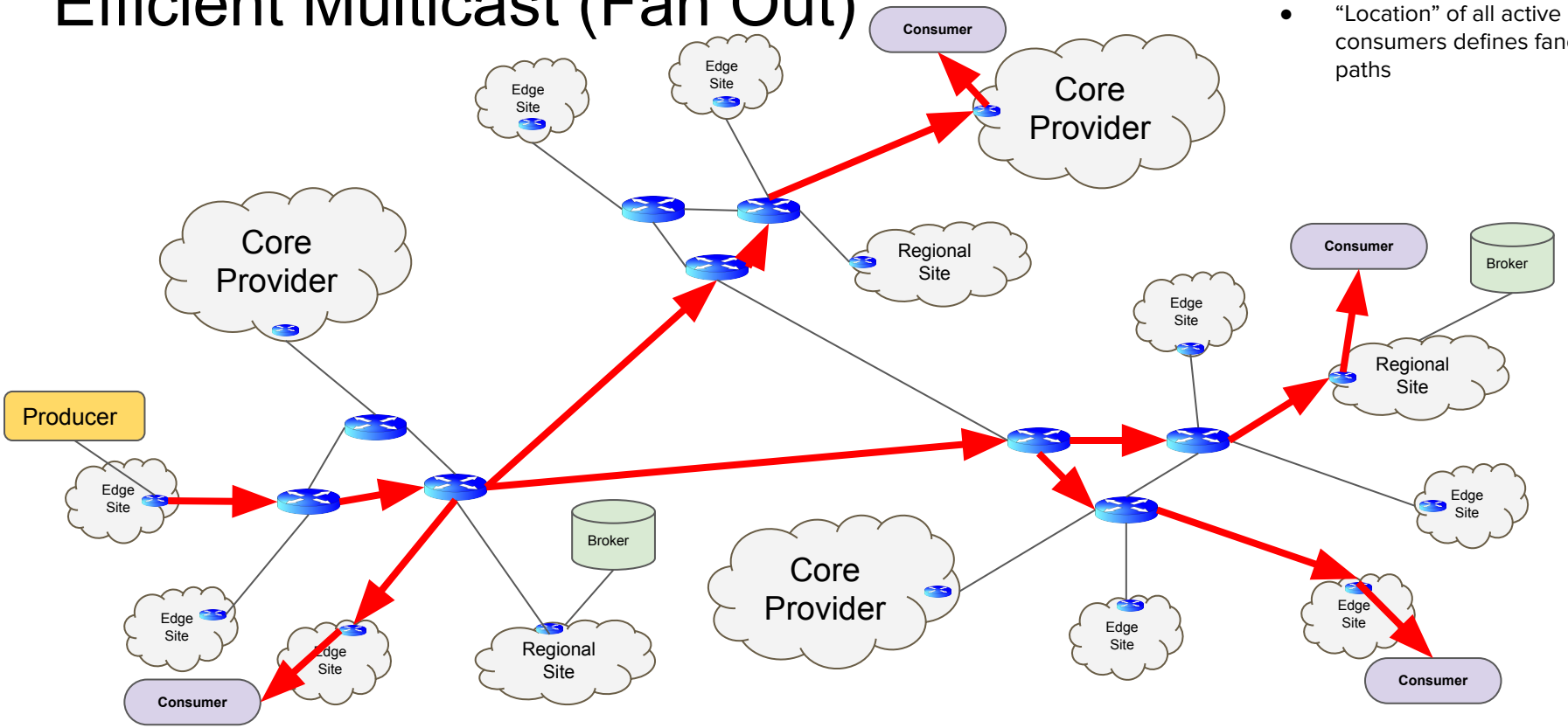
Efficient Multicast (Fan Out)



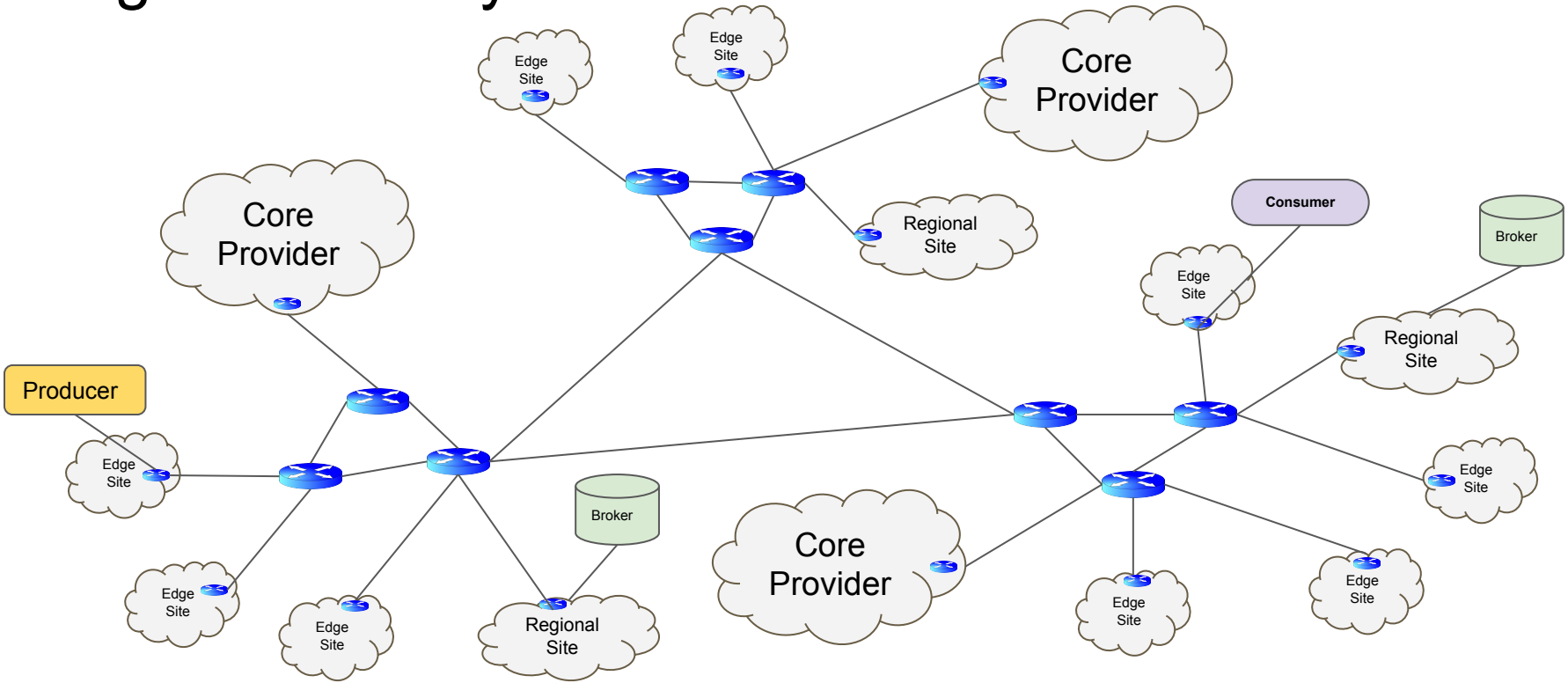
- Producer sends message with multicast delivery treatment
- Target all active consumers with listening address (e.g. `/telemetry/events`

Efficient Multicast (Fan Out)

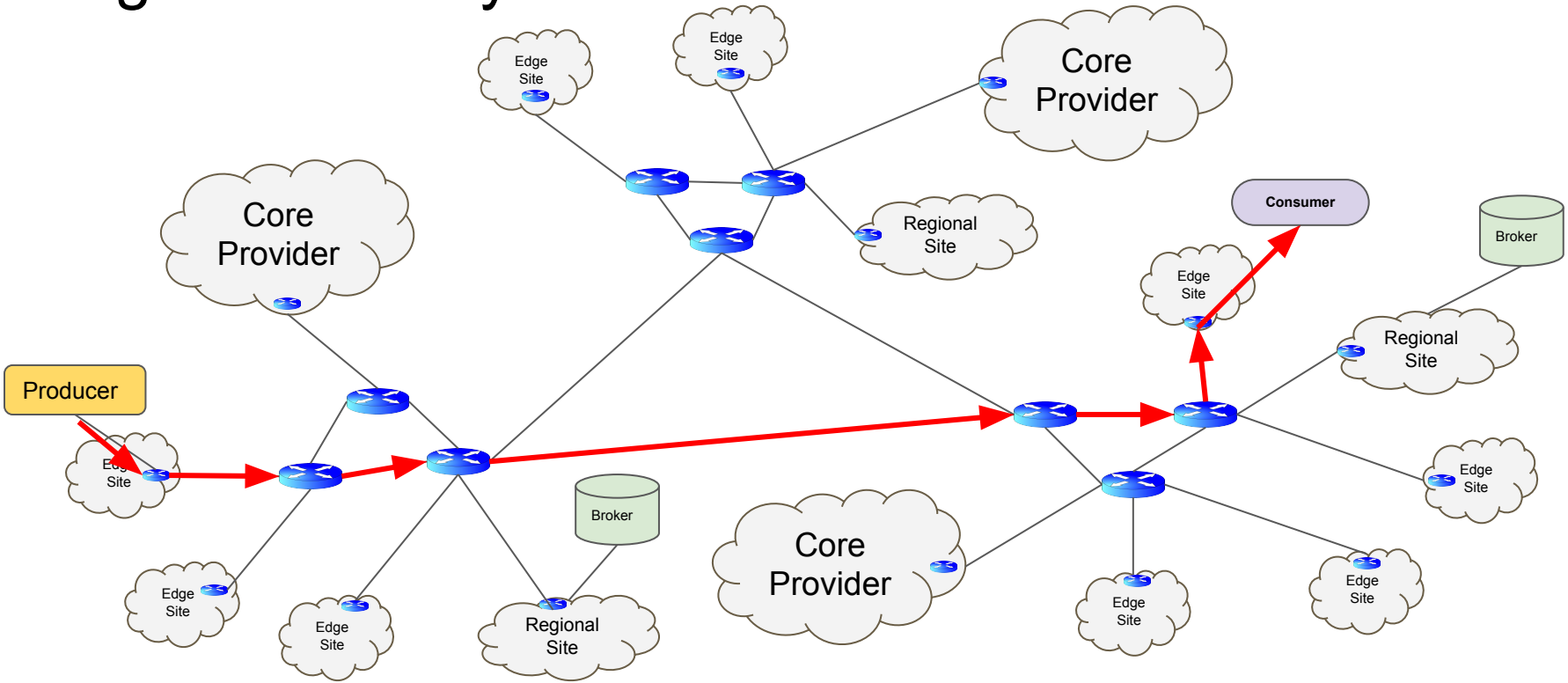
- “Location” of all active consumers defines fanout paths



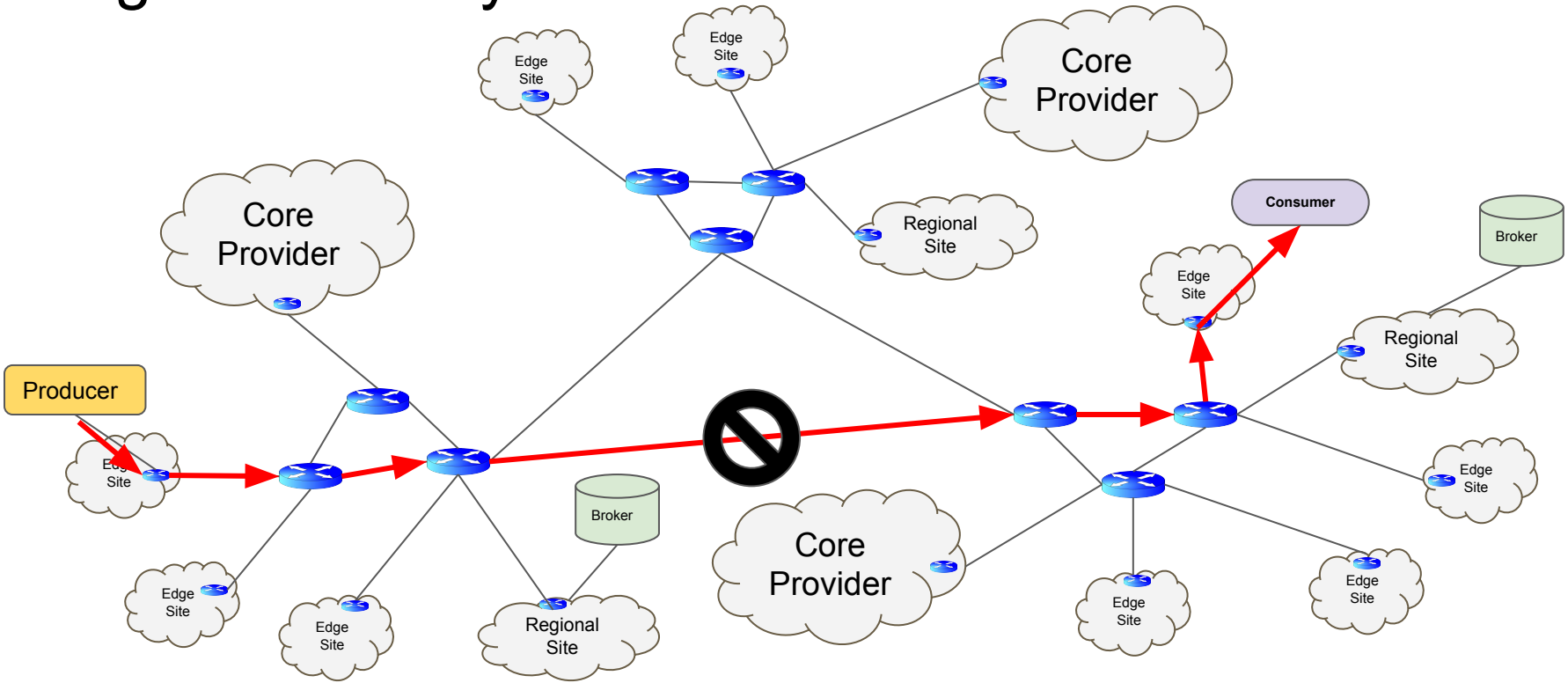
High Availability



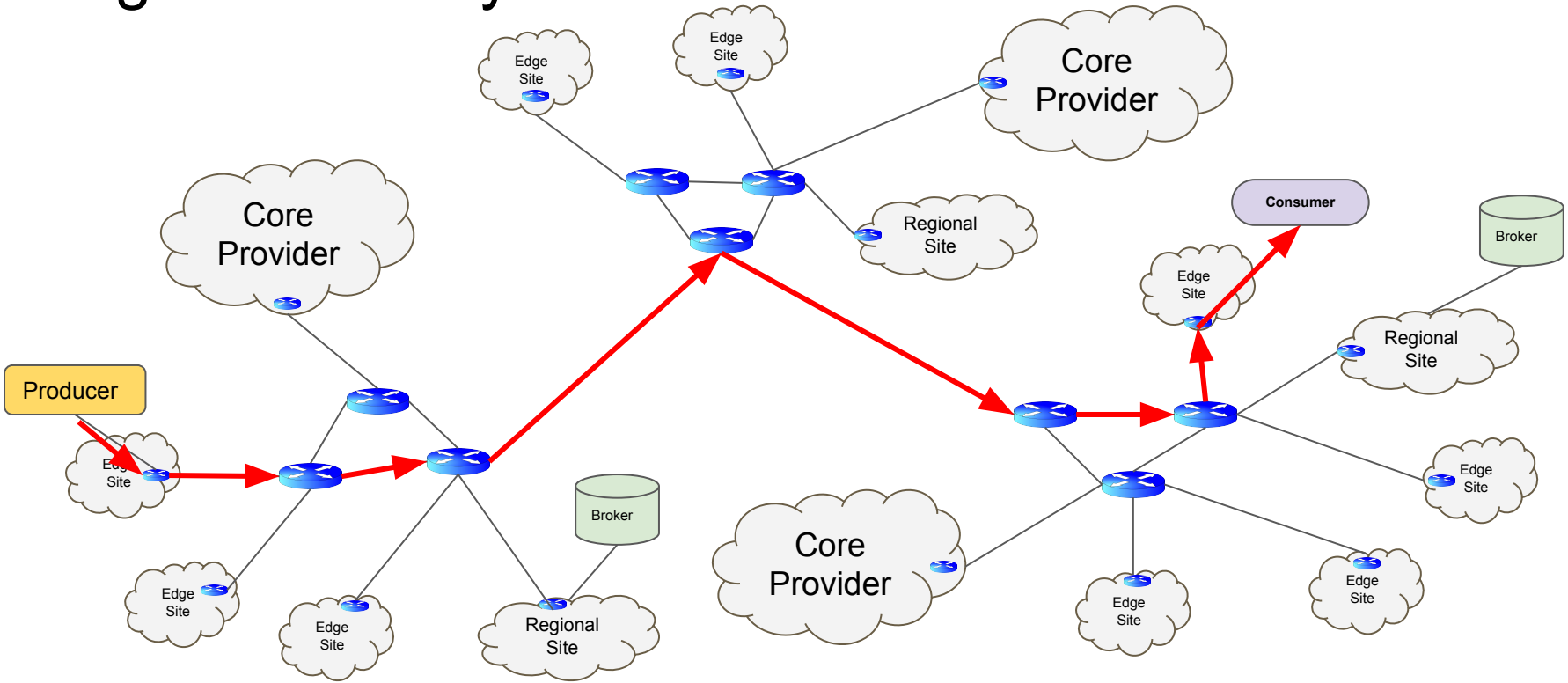
High Availability



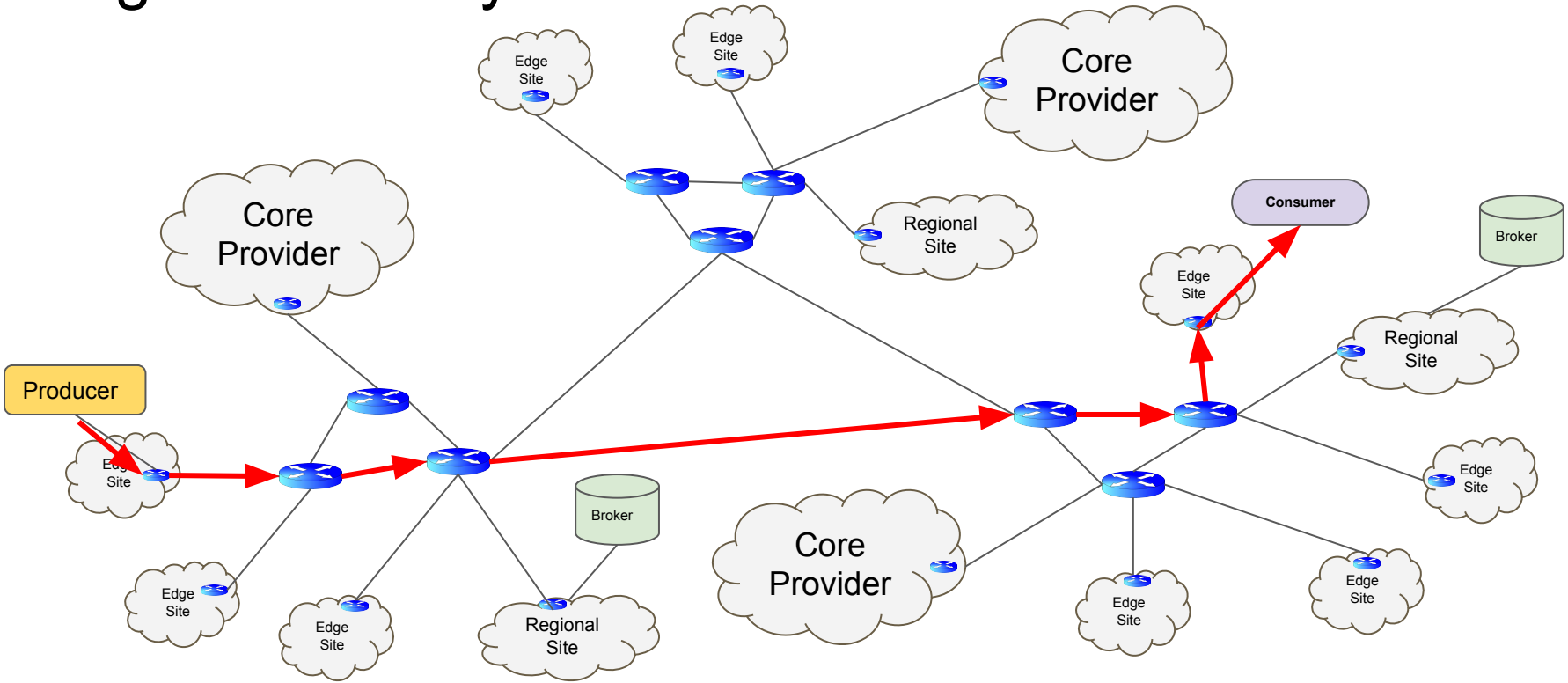
High Availability



High Availability



High Availability



Live data plane demo

Material used in demo is here

- https://github.com/interconnectedcloud/qdr-operator
- https://github.com/grs/addressing-demo

Security

Edge Security Challenges

Security at the edge cannot be an afterthought… it must be designed into our edge infrastructure from the start

Edge Security Challenges

Let’s start by exposing the security challenges of the edge

Edge Security Challenges

When we have a complete picture of our security problems,
we can provide more complete solutions

Edge Environment Differences

- Diverse networks
- No guarantees of continuous power
- Intermittent connectivity
- Direct physical access to hardware
- Heterogeneous hardware
- Non-TCP/IP communication
- Multiple vendors in a single solution
- Need to handle security in “offline mode”
- Low latency locally, higher latency to cloud

Edge Security Challenges - Overview

- Trusting edge hardware
- Trusting connected devices
- Operating systems
- Network concerns
- Edge microservices

Trusting Edge Hardware

- Physical security is not guaranteed at the edge

Trusting Edge Hardware

- Physical security is not guaranteed at the edge
- Hardware root of trust is a starting point

Trusting Edge Hardware

- Physical security is not guaranteed at the edge
- Hardware root of trust is a starting point
- Trustworthy condition of hardware (location, resource availability, etc.)

Trusting Edge Hardware

- Physical security is not guaranteed at the edge
- Hardware root of trust is a starting point
- Trustworthy condition of hardware (location, resource availability, etc.)
- Trusting attached devices (USB, serial, SATA, etc.)

Trusting Edge Hardware

- Physical security is not guaranteed at the edge
- Hardware root of trust is a starting point
- Trustworthy condition of hardware (location, resource availability, etc.)
- Trusting attached devices (USB, serial, SATA, etc.)
- Reacting to indication of compromise

Trusting Edge Hardware

- Physical security is not guaranteed at the edge
- Hardware root of trust is a starting point
- Trustworthy condition of hardware (location, resource availability, etc.)
- Trusting attached devices (USB, serial, SATA, etc.)
- Reacting to indication of compromise
- Authenticity of hardware

Trusting Connected Devices

- Verifying devices and detecting corruption

Trusting Connected Devices

- Verifying devices and detecting corruption
- Protecting data and commands

Trusting Connected Devices

- Verifying devices and detecting corruption
- Protecting data and commands
- Device management

Operating System

- BIOS and secure boot

Operating System

- BIOS and secure boot
- Running processes and binary attestation

Operating System

- BIOS and secure boot
- Running processes and binary attestation
- False sense of trust using fixed identities

Operating System

- BIOS and secure boot
- Running processes and binary attestation
- False sense of trust using fixed identities
- Component firmware vulnerabilities

Operating System

- BIOS and secure boot
- Running processes and binary attestation
- False sense of trust using fixed identities
- Component firmware vulnerabilities
- Security updates of the operating system

Operating System

- BIOS and secure boot
- Running processes and binary attestation
- False sense of trust using fixed identities
- Component firmware vulnerabilities
- Security updates of the operating system
- Audit trail and log files

Network Concerns

- Open ports

Network Concerns

- Open ports
- Fixed VPNs

Network Concerns

- Open ports
- Fixed VPNs
- Network access control

Network Concerns

- Open ports
- Fixed VPNs
- Network access control
- Identity verification of control plane

Network Concerns

- Open ports
- Fixed VPNs
- Network access control
- Identity verification of control plane
- Attacks of transport layer

Network Concerns

- Open ports
- Fixed VPNs
- Network access control
- Identity verification of control plane
- Attacks of transport layer
- Denial-of-thing attacks

Edge Microservices

- Purity of images

Edge Microservices

- Purity of images
- Secure delivery of secrets

Edge Microservices

- Purity of images
- Secure delivery of secrets
- Unauthorized microservices

Edge Microservices

- Purity of images
- Secure delivery of secrets
- Unauthorized microservices
- Controlled access to resources

Edge Microservices

- Purity of images
- Secure delivery of secrets
- Unauthorized microservices
- Controlled access to resources
- Guaranteed remote shutdown

Edge Microservices

- Purity of images
- Secure delivery of secrets
- Unauthorized microservices
- Controlled access to resources
- Guaranteed remote shutdown
- Matching microservices to edge hardware

Edge Microservices

- Purity of images
- Secure delivery of secrets
- Unauthorized microservices
- Controlled access to resources
- Guaranteed remote shutdown
- Matching microservices to edge hardware
- Unauthorized outbound

Edge Security

Security at the edge will require a multitude of approaches with effort from many

Edge Security

Get involved now, question the status quo, and remember that the edge is a very different place than the cloud

Questions?

How to get involved with the Working Group, learn more...

Regular Work Group Meeting:

USA WG Meeting Wednesday 9am PT, every 4 weeks, next on June 19

APAC WG meeting Wednesday 5 UTC every 4 weeks, next on June 5

- Meeting notes and agenda

Link to join the group

- https://groups.google.com/forum/#!forum/kubernetes-wg-iot-edge

Link to join Slack

- https://kubernetes.slack.com/messages/wg-iot-edge

White Paper

- http://bit.ly/iot-edge-whitepaper