



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

SIG Network Update

The state of networking support for AI/ML



Who are we?



KubeCon



CloudNativeCon

Europe 2026



Nadia Pinaeva

Senior Software Engineer @ Nvidia
Network Policy Maintainer



Haiyan Meng

Senior Software Engineer @ Google
Kube Agentic Networking Maintainer



David Martin

Principal Software Engineer @ Red Hat
Kube Agentic Networking Maintainer



Kellen Swain

Google
Gateway API Inference



Bowei Du

Google
SIG-NET Chair



Agenda



KubeCon



CloudNativeCon

Europe 2026



Introduction to SIG Network



AI/ML Networking Overview



Q&A



Introduction to SIG Network



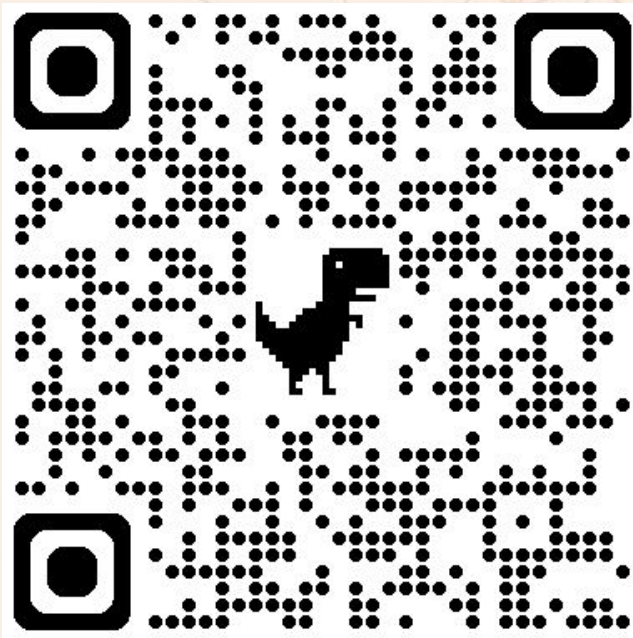
KubeCon



CloudNativeCon

Europe 2026

- SIG Network manages components, interfaces, and APIs for networking in Kubernetes
- It maintains core code such as `kube-proxy` for the Service API
- Defines APIs for other networking needs (e.g. Ingress, Gateway API)
- We have a variety of sub-projects (non-exhaustive):
 - Network Policy
 - Multi-Network
 - Gateway API + AI-related sub-sub projects (we will discuss)
- We work with other SIGs (e.g. Node, Scheduling, Multi-Cluster)
- We participate in working groups (e.g. AI Gateway WG, AI Conformance WG)



AI/ML Networking Overview



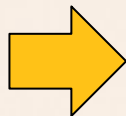
KubeCon



CloudNativeCon

Europe 2026

- Hardware
- Training
- Inference
- Agentic workloads



- Gateway API
- Gateway API Inference Extension
- Kube Agentic Networking
- AI Gateway WG
- Dynamic Resource Allocation (DRA)



DRANET is a Kubernetes Network Driver that uses Dynamic Resource Allocation (DRA) to deliver high-performance networking for demanding applications in Kubernetes.

It does **resource discovery**:

```
apiVersion: resource.k8s.io/v1
kind: ResourceSlice
metadata: ...
spec:
  devices:
    - basic:
        attributes:
          dra.net/cloudNetwork:
            string: dra-1-vpc
          dra.net/encapsulation:
            string: ether
          dra.net/ifName:
            string: gpu7rdma0
```

```
dra.net/ipv4:
  string: 10.0.8.8
dra.net/mac:
  string: 9a:41:2e:4f:86:16
dra.net/mtu:
  int: 8896
dra.net/numaNode:
  int: 1
dra.net/pciAddressBus:
  string: c8
dra.net/pciAddressDevice:
  string: "00"
dra.net/pciAddressDomain:
  string: "0000"
```

```
dra.net/pciAddressFunction:
  string: "0"
dra.net/pciVendor:
  string: Mellanox Technologies
dra.net/rdma:
  bool: true
dra.net/state:
  string: up
```



DRANET: RDMA handling



KubeCon



CloudNativeCon

Europe 2026

DRANET manages three primary types of RDMA-related components for Pods:

1. **RDMA Character Devices:** These are user-space interfaces (e.g., `/dev/infiniband/uverbsN`, `/dev/infiniband/rdma_cm`) that user applications interact with to set up RDMA resources.
2. **RDMA Network Devices:** This refers to the standard Linux network interface (e.g., `eth0`, `gpu3rdma0`, `ib0`) that receives an IP address and participates in the conventional TCP/IP networking stack, often built on top of an HCA.
3. **RDMA Link Devices:** This is the physical hardware adapter card (Host Channel Adapter - HCA), often named like `mlx5_0`, `mlx4_0`, or `hfi1_0`, which performs the core RDMA operations.

It supports 2 modes:

- In **shared** mode, the RDMA subsystem allows multiple network namespaces to access the same RDMA link devices.
- In **exclusive** mode, the RDMA subsystem enforces stricter isolation, allowing an RDMA device to be assigned to only one network namespace at a time.



DRANET: topology awareness



KubeCon



CloudNativeCon

Europe 2026

DRANET allows attribute matching across resources.

This provides topology awareness across devices, but only on the same node.

To select GPU and NIC that share the same PCIe root use **ResourceClaimTemplate** with constraints

```
apiVersion: resource.k8s.io/v1
kind: ResourceClaimTemplate
metadata:
  name: 2-gpu-nic-aligned
spec:
  devices:
    requests:
      - name: gpu
        exactly:
          deviceClassName: gpu.nvidia.com
      - name: nic
        exactly:
          deviceClassName: dranet
        selectors:
          - cel:
              expression: device.attributes["dra.net"].rdma == true
  constraints:
    - matchAttribute: "resource.kubernetes.io/pcieRoot"
```



DRA and cross-node topology



KubeCon



CloudNativeCon

Europe 2026

DRANET solves the topology awareness problem **within the node**, but there is currently no way to request **cross-node** networking constraints.

SIG-Scheduling is working on this as part of the PodGroup scheduling usually referred to as TopologyAwareScheduling.

Some concepts and ideas:

- Use performance-based metrics for networking
 - Instead of same-rack/same-rail requests, ask for specific performance metrics
- Performance domain is a set of nodes/devices that all meet the same performance requirements
- Multi-plugin system for different vendors to tell which nodes have high-performance networking between them
- Potential extensions for device-based instead of node-based scheduling
- Currently already support node label-based scheduling



Gateway API

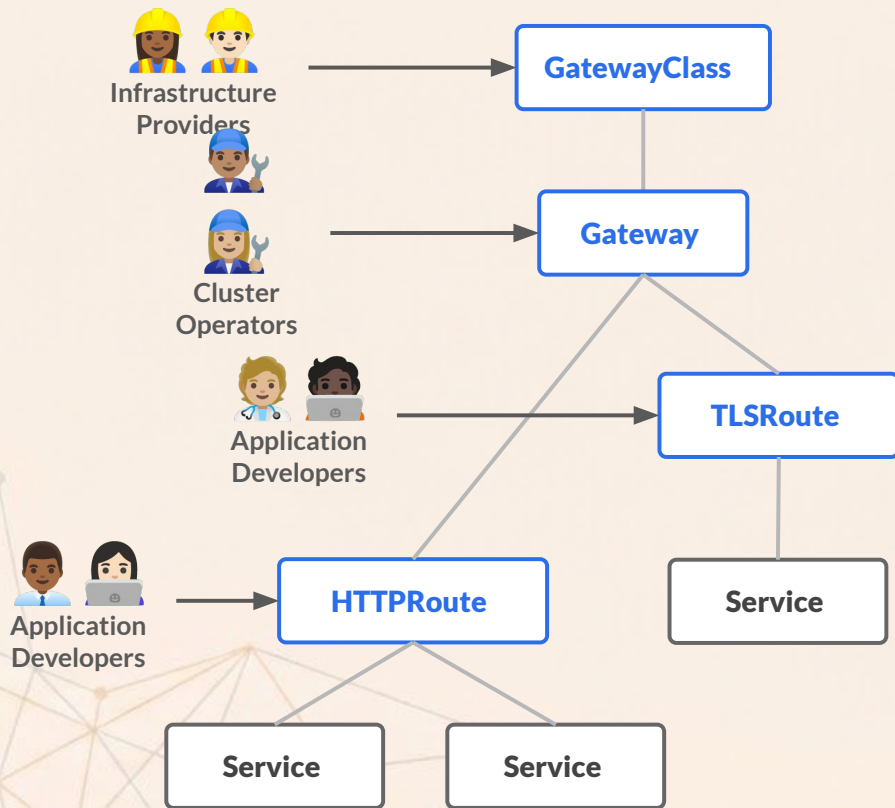


KubeCon



CloudNativeCon

Europe 2026



- Role-oriented
- Portable
- Expressive
- **Extensible**



Gateway API - Extensibility

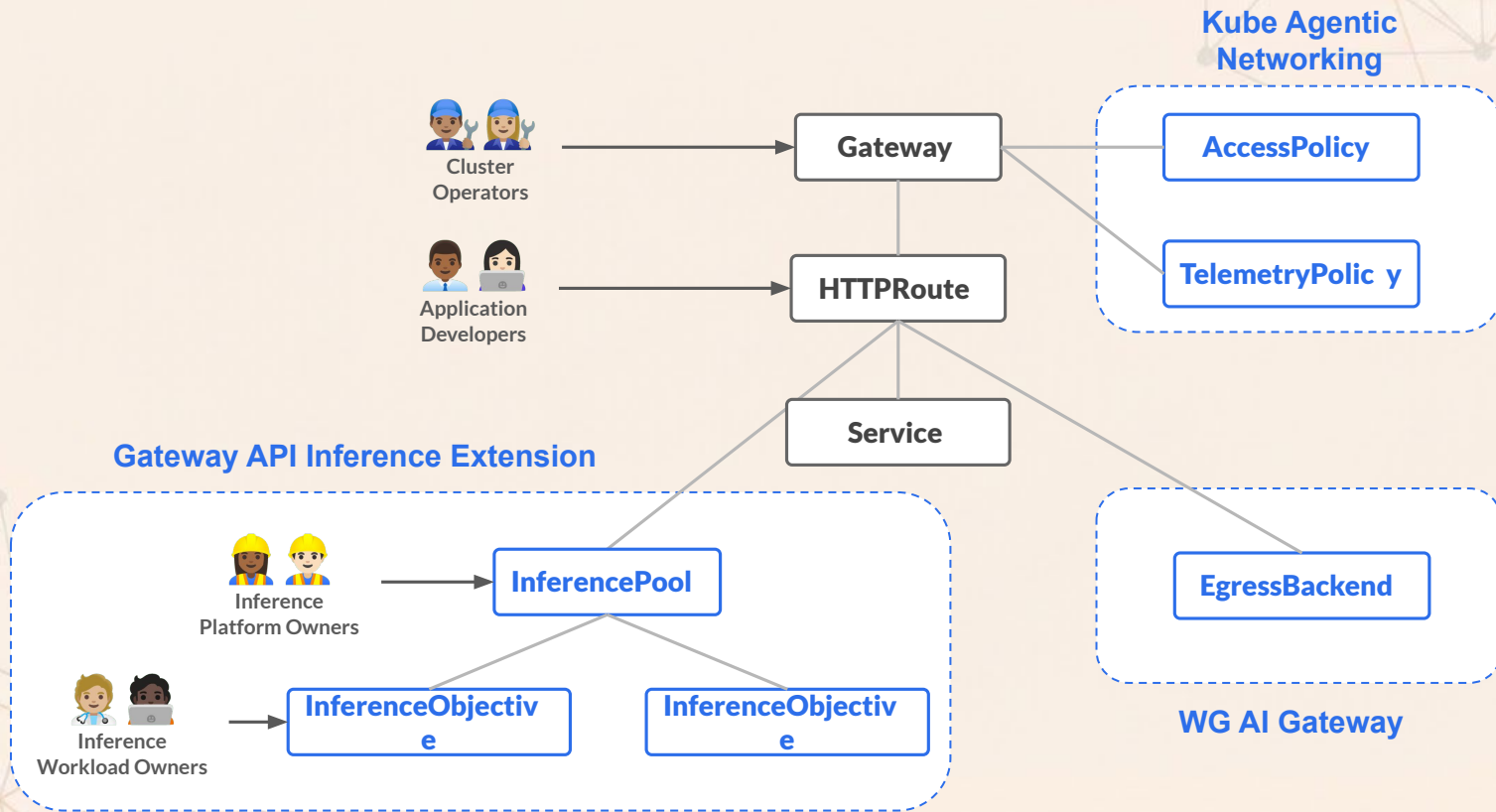


KubeCon



CloudNativeCon

Europe 2026



Gateway API Inference Extension



KubeCon



CloudNativeCon

Europe 2026

Rationale



Inference requests are not uniform

Unlike service-to-service calls, inference requests vary considerably and are expensive.

Different goals affect UX

Interactive workloads require low TTFT and TPOT

Traditional load balancing falls short

Network and application load balancing (e.g., RR, least loaded) fail to account for diversity. Body access needed for scheduling.

Functionality



Inference aware scheduling

Inspect both request and serving endpoint for optimal routing (cost/performance based on KV prefix).

Flow control

Serving priorities to address expensive serving and limited preemption

Model rollouts

Canary, blue/green etc via model rewrite

Ecosystem Integration



Optimizes self-hosting models on Kubernetes

Integrates with Gateway API

Seamlessly integrates with Gateway API as a new backend type (InferencePool).

Extensible APIs

Designed for diverse implementations. Front end handles AAA, sets category



Kube Agentic Networking

<https://kube-agentic-networking.sigs.k8s.io>



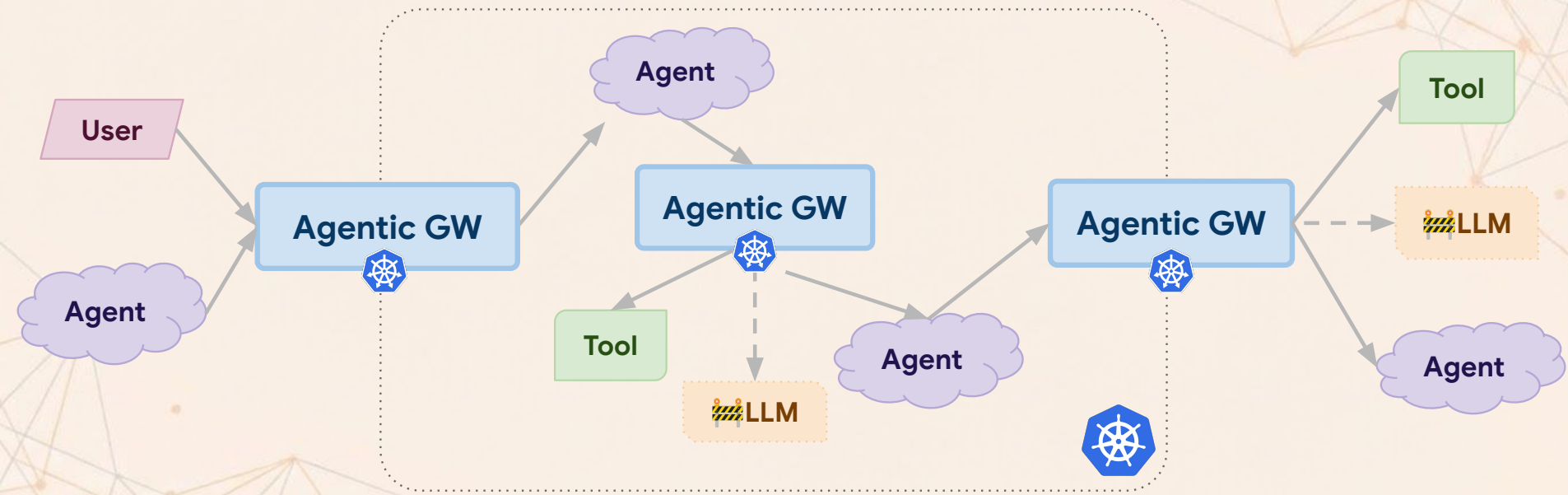
KubeCon



CloudNativeCon

Europe 2026

Provide standardized APIs for secure, governed communication for the next generation of AI agents.



 Potential Future Scope



Kube Agentic Networking



KubeCon



CloudNativeCon

Europe 2026

Core Capabilities



Standardized APIs

Secure agent-to-agent and agent-to-tool communication across cluster boundaries.

Adaptable APIs

Designed around user-facing goals, rather than specific AI protocols.

Security & Governance



Authorization Policies

Control exactly which agents access what tools.

AI Safety Integration

Support external authn/authz, and policy enforcement.

Auditable Traffic Management

Rate limiting, traffic auditing.

Ecosystem Integration



Gateway API

Seamlessly integrates with Gateway API and major AI protocols.

Extensible APIs

Designed for diverse implementations (service meshes, gateways, and future architectures).



Kube Agentic Networking - Use Cases



KubeCon



CloudNativeCon

Europe 2026

Persona

What persona needs

AI engineer

- Assign identities to agents
- Audit agent actions
- Understand why an agent gets denials on tool calls

Platform engineer

- Default-deny agent traffic
- Control access to tools (MCP servers)
- Observe platform-wide failures or denials

Security engineer

- Pre-request filtering to avoid attacks like prompt injection
- Post-response filtering to prevent data breaches



Kube Agentic Networking



KubeCon



CloudNativeCon

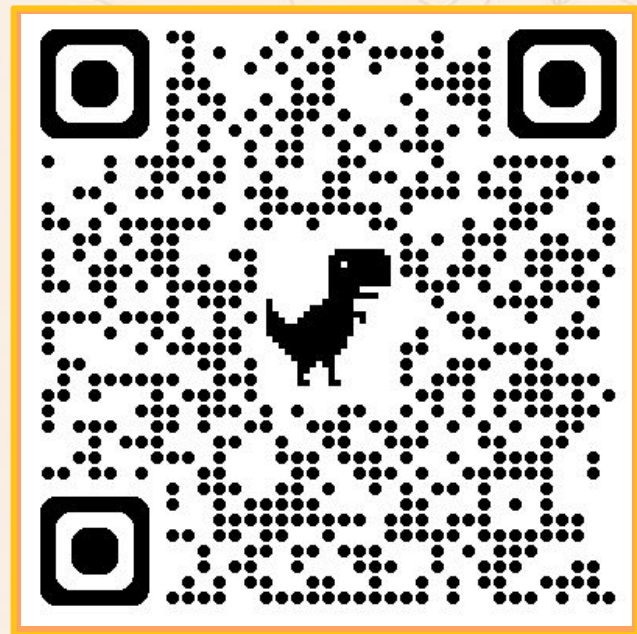
Europe 2026

Ongoing work

- XBackend
- XAccessPolicy: targeting Gateway or XBackend
- Observability standards & Telemetry policy

Future work

- Support denylist
- Authorization policies for communication between agents and agents, between agents and LLMs
- Governance of ingress AI traffic
- Rate limiting
- Guardrails, collaborating through AI Gateway working group



Learn more about Agentic Networking!

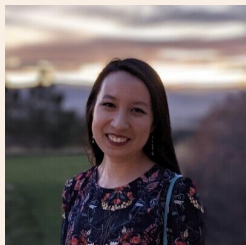


KubeCon



CloudNativeCon

Europe 2026



Evaline Ju

Senior Software Engineer @ IBM
Research



Haiyan Meng

Senior Software Engineer @ Google
Kube Agentic Networking Maintainer

**Agentic Networking:
Securing AI Agents on Kubernetes**
Wednesday March 25, 2026
17:30 - 18:00 CET
Elicium 1



AI Gateway WG

<https://github.com/kubernetes-sigs/wg-ai-gateway>



KubeCon



CloudNativeCon

Europe 2026

- SIG Network/SIG Multi-Cluster sponsored WG
- Standardizing how gateways handle AI traffic
 - Builds on the **Gateway API** specification
- **Payload Processing** - processing full HTTP request/response bodies
 - enables prompt guardrails, semantic routing/caching, RAG augmentation
- **Egress Gateways** - routing traffic to external inference services (other clusters, cloud AI APIs) with the same policy controls as ingress





KubeCon



CloudNativeCon

Europe 2026

Discussion





KubeCon



CloudNativeCon

Europe 2026



Feedback Link for this talk

Who are we?



KubeCon



CloudNativeCon

Europe 2026



Nadia Pinaeva

Senior Software Engineer @ Nvidia
Network Policy Maintainer



Haiyan Meng

Senior Software Engineer @ Google
Kube Agentic Networking Maintainer



David Martin

Principal Software Engineer @ Red Hat
Kube Agentic Networking Maintainer



Etai Lev Ran

STSM @ IBM
Gateway API Inference
Extension Contributor



Bowei Du

Senior Staff @Google
SIG-NET Chair

