KubeCon CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024

KubeCon

CloudNativeCon

China 2024

How Does KubeEdge Build the Tunnel Which Is Secure, Trusted, and Adaptable to Edge Networks

Wei Hu | DaoCloud

Clement Richard | n-hop

Abort Us

Wei Hu
Senior Software Engineer at DaoCloud
KubeEdge Maintainer

Clement Richard
Platform Engineer at n-hop
Red Hat Certified Architect

Agenda

01 Background and motivation

02 Security Of KubeEdge Connections

03 Cloud-Edge Control Flow

04 Network Coding for Cloud Computing

05 Q&A

Background and motivation

The Challenges Of Edge Computing Network

User Story

The heterogeneous nature of the network at the edge (e.g., Internet, 5G, WIFI, and other forms) makes achieving quality of service much more difficult. Users want an edge computing platform that can support weak and insecure network environments.

Challenges

- Security of communication.
- Adaptability to weak network environments.
- High throughput, low latency and reliability.

The Requirement Of The IOV Scenario

1

System dynamic and open, the network environment is unstable and requires edge interconnection.

2

Security and privacy protection, to ensure the security of data communication and safeguard users' privacy to prevent leaks..

3

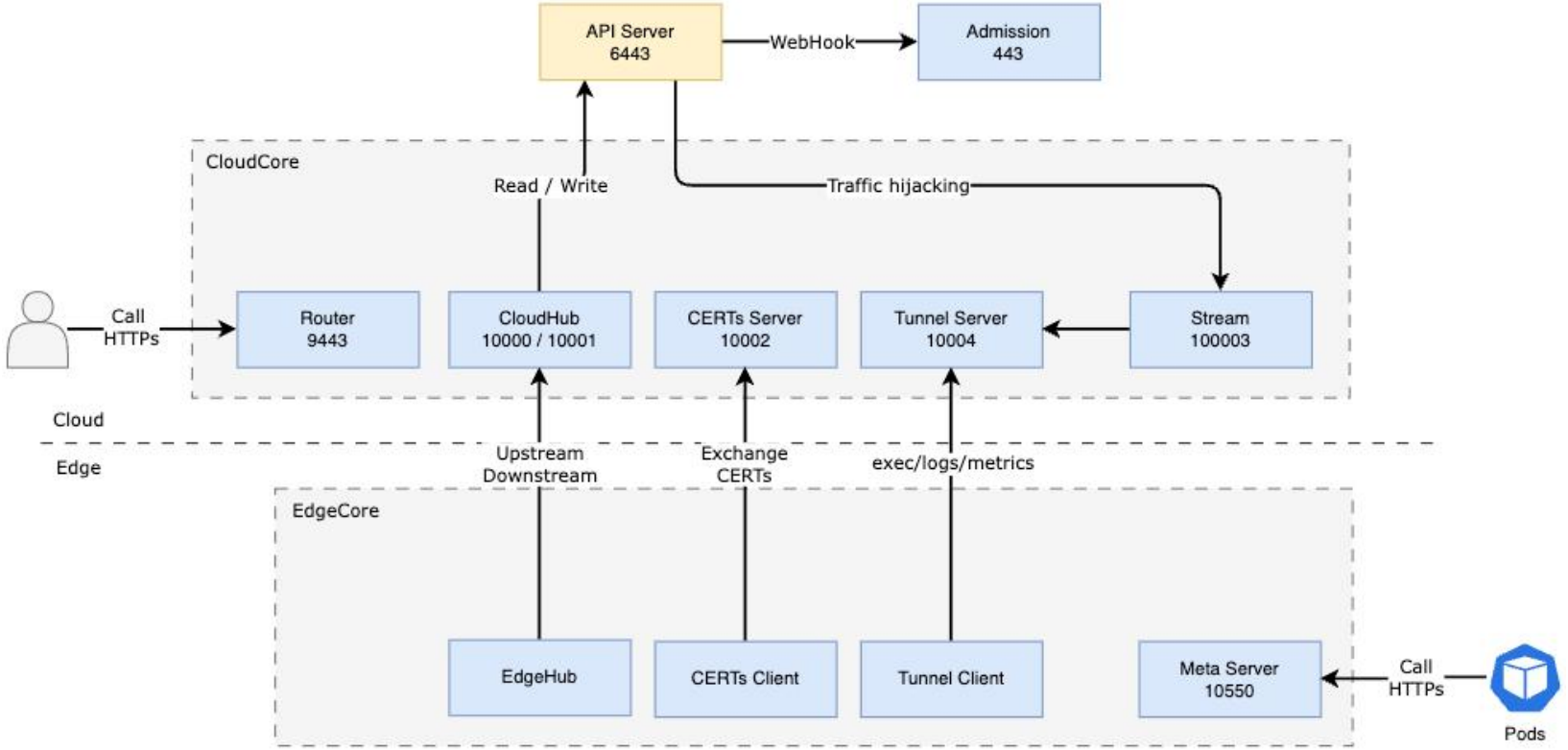
Network bandwidth and equipment costs, The corresponding compression algorithm should be adopted to reduce the bandwidth demand.

4

Management of massive devices, efforts must be made to alleviate the network load on cloud.

Security Of KubeEdge Connections

Data Flow Of KubeEdge



CloudHub

Provides WebSocket or QUIC Server.
List/Watch K8s resources for downstream.
Write k8s resource status for upstream.

Stream & Tunnel Server

Stream hijacks the stream apis from api-server to kubelet and calls Tunnel Server(WebService) to pass stream datas.

EdgeHub & Tunnel Client

Connects CloudHub for control-flow and connects TunnelServer for stream-flow.

Meta Server

Provides K8s resource APIs that can be accessed offline.

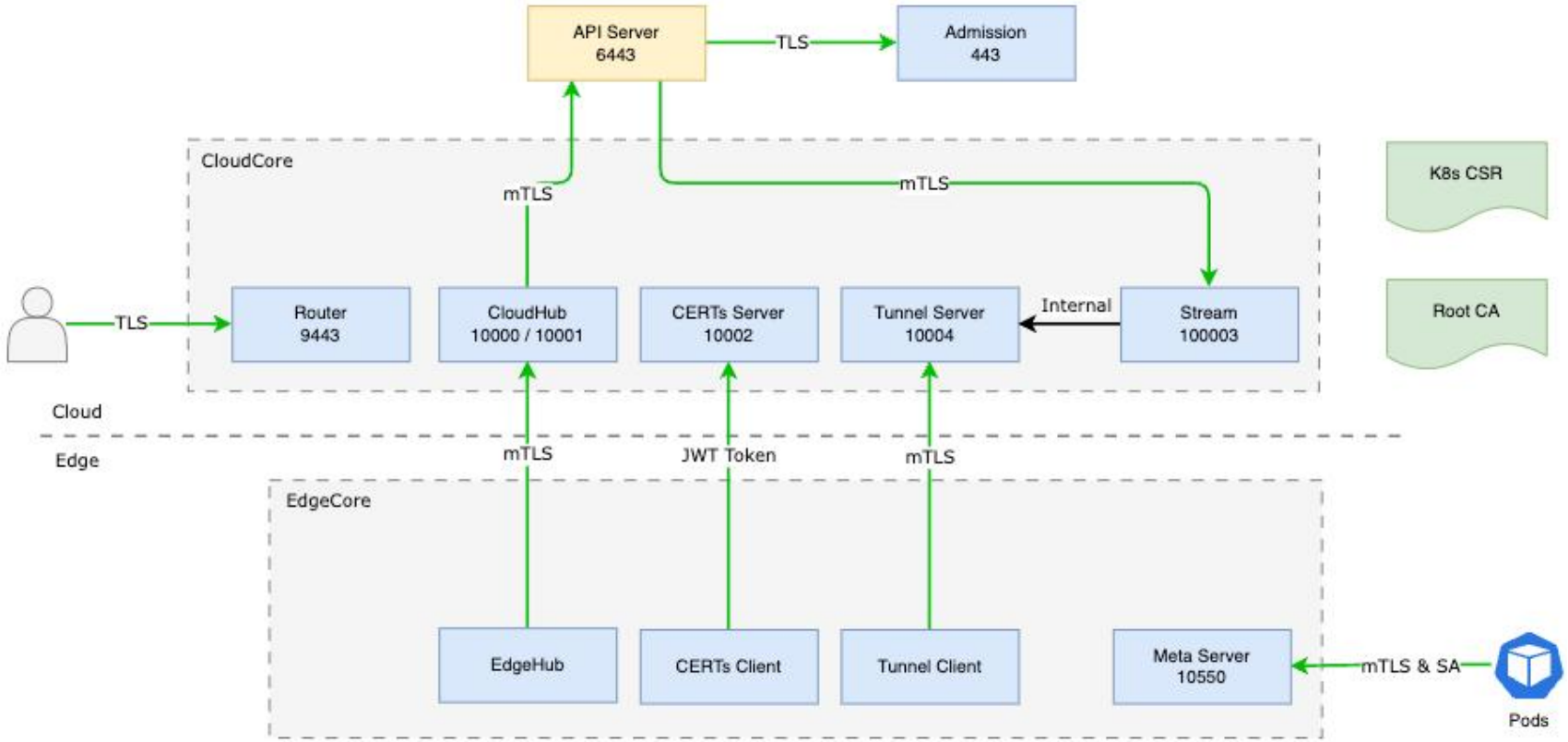
Ruter

Calling the router API enables cloud apps can be called to the edge via eventbus/servicebus.

Admission

Responsible for verifying KubeEdge CRDs.

TLS Certificate



KubeEdge supports only two methods for obtaining certificates: K8s CRS and Self signed.

All external services use tls or mtls.

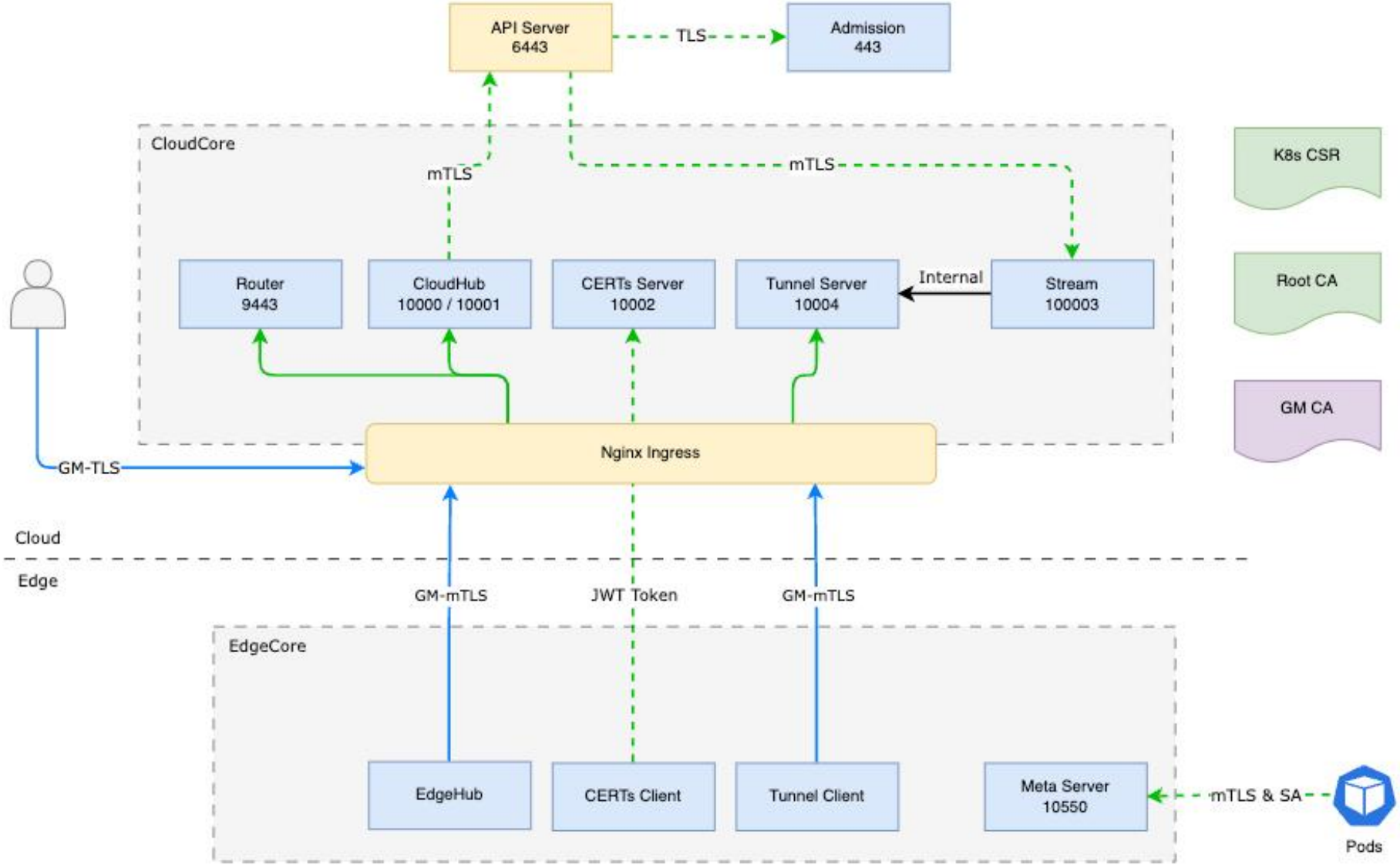
CloudHub and Stream uses C/S certificates of api-server.

Router obtains a server certificate via Root CA.

EdgeCore obtains client certificates via the token to be exchanged when node joined. One certificate per node.

MetaServer obtains a server certificate via K8s CSR.

GM-TLS Certificate



Go's native tls package does not support the GM-TLS.

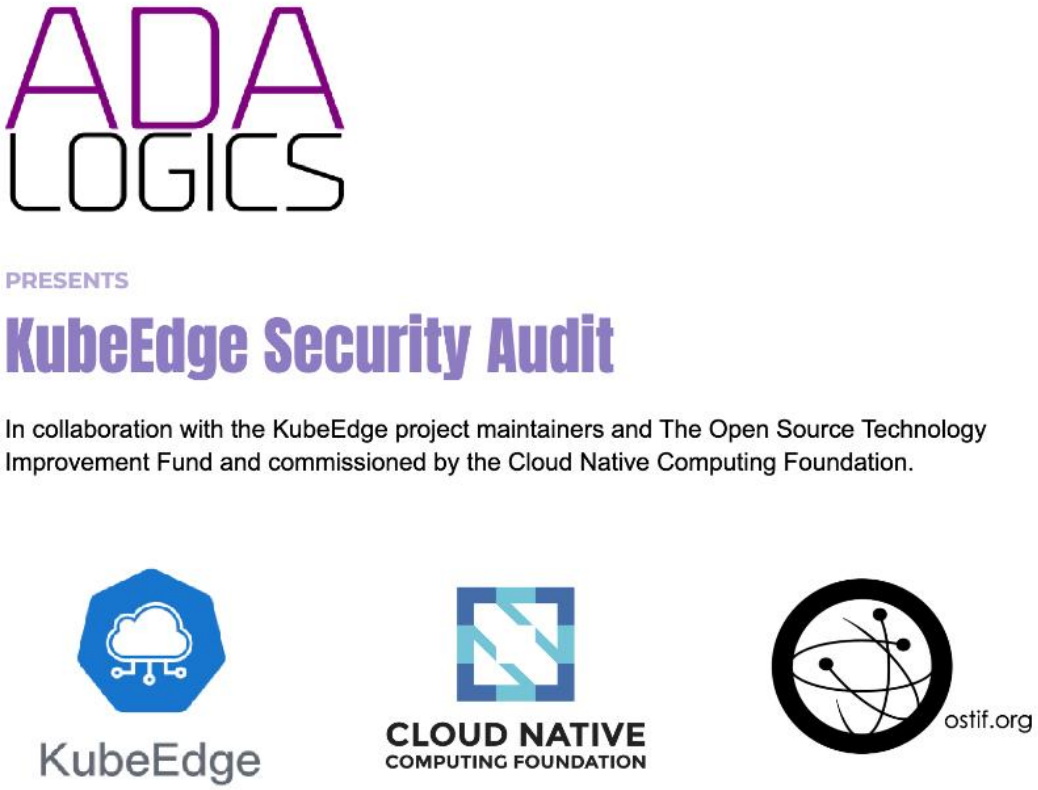
Nginx extendeds(i.e., Tengine, Tongsuo, etc.) supports GM-TLS. This's cheaper than coding.

K8s does not support GM-TLS, so we only need Router, CloudHub, TunnelServer to support it.

Clients use GM-TLS to access Nginx ingress, Nginx converts the CloudCore TLS certificate to access the CloudCore servers.

Security Audit Of KubeEdge

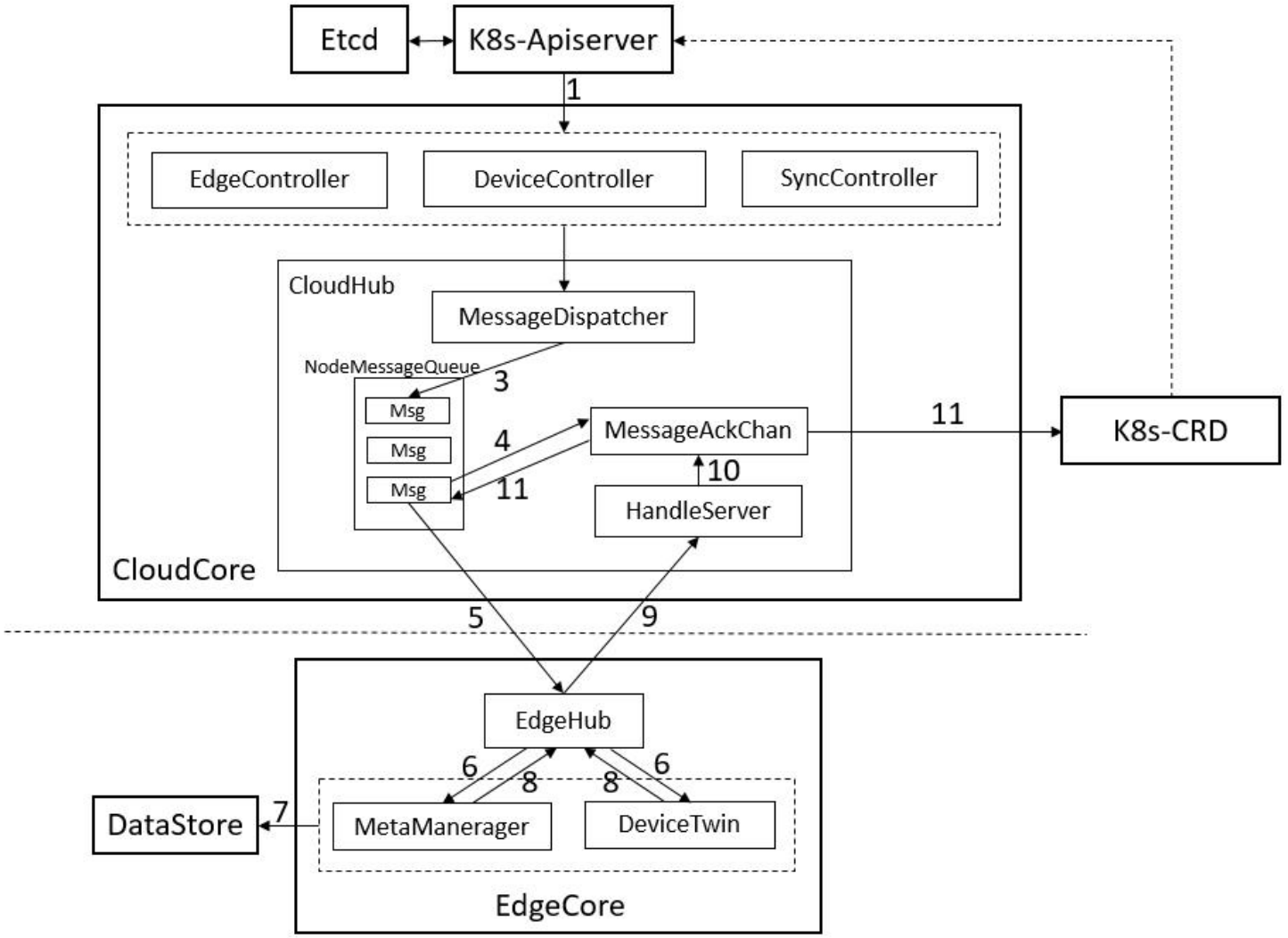
A third party security audit of KubeEdge has been completed in July 2022.



- Security Audit
- Threat model and security protection analysis paper

Cloud-Edge Control Flow

HUB Tunnel



CloudCore uses list/watch APIServer to alleviate the request load pressure and enable the cluster to join more edge nodes.

A dedicated tunnel between CloudHub and EdgeHub that processes control plane data by message queues.

The QUIC protocol of Hub tunnel helps to optimize network communication in weak network environments.

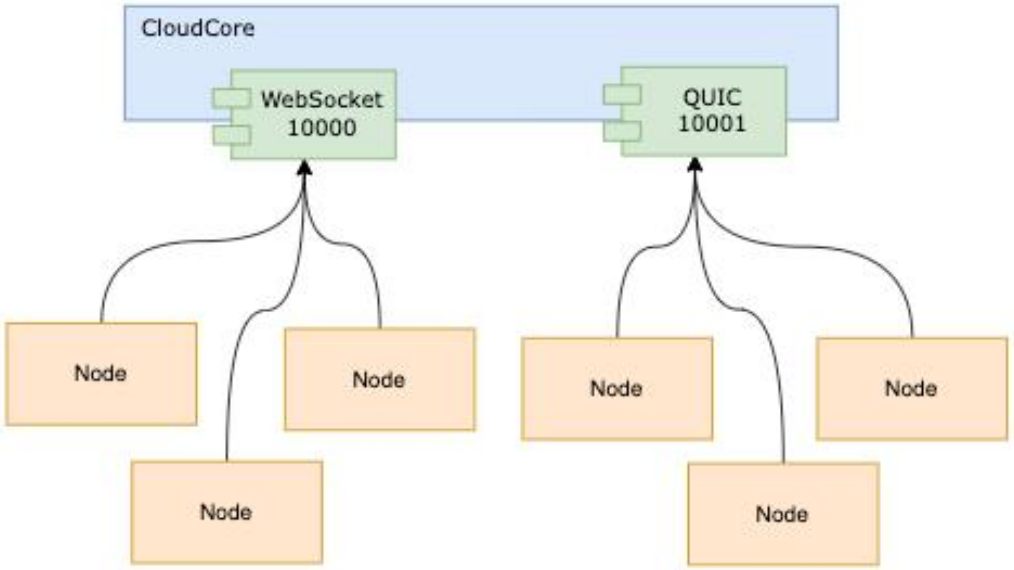
The ACK mechanism can be used to ensure the reachability of messages.

The data authz function can be enabled so that a node can only read/write K8s resources within the current node.

Roadmap:

Merge messages for a time segment and compress messages.

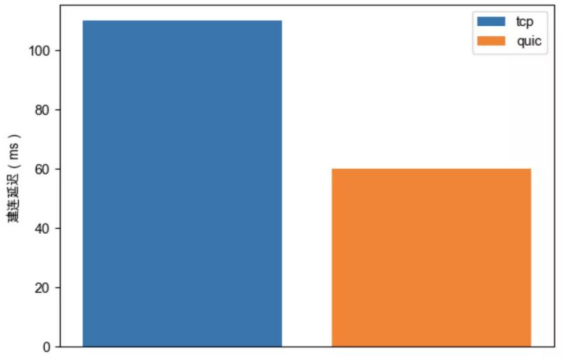
QUIC Potocol

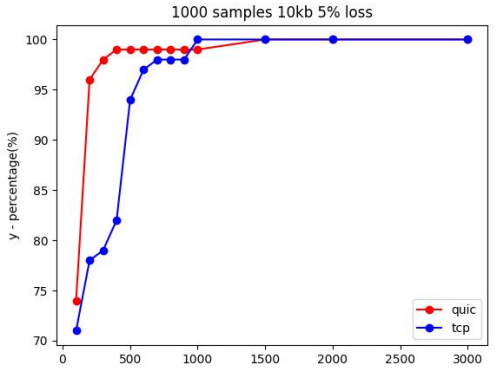


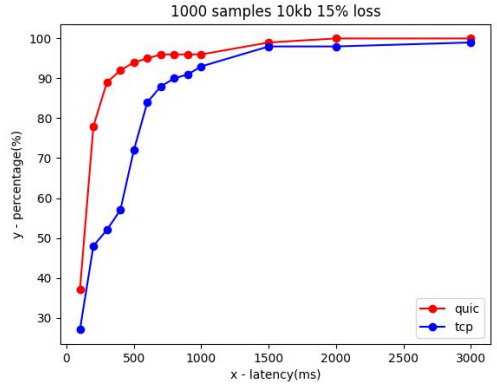
CloudCore can enable both WebSocket and QUIC tunnels, or selectively enable either one as required.

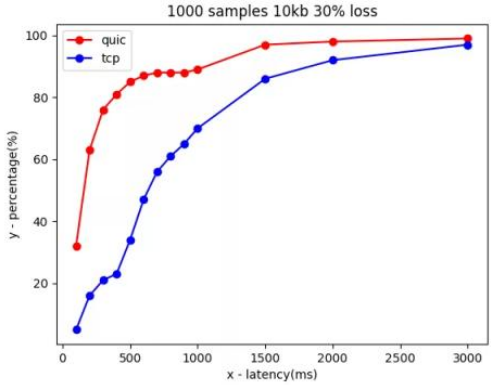
Key features of QUIC

- Reduced connection establishment time - 0 round trips.
- Improved congestion control feedback.
- Multiplexing without head of line blocking.
- Connection migration.
- Transport extensibility.
- Optional unreliable delivery.

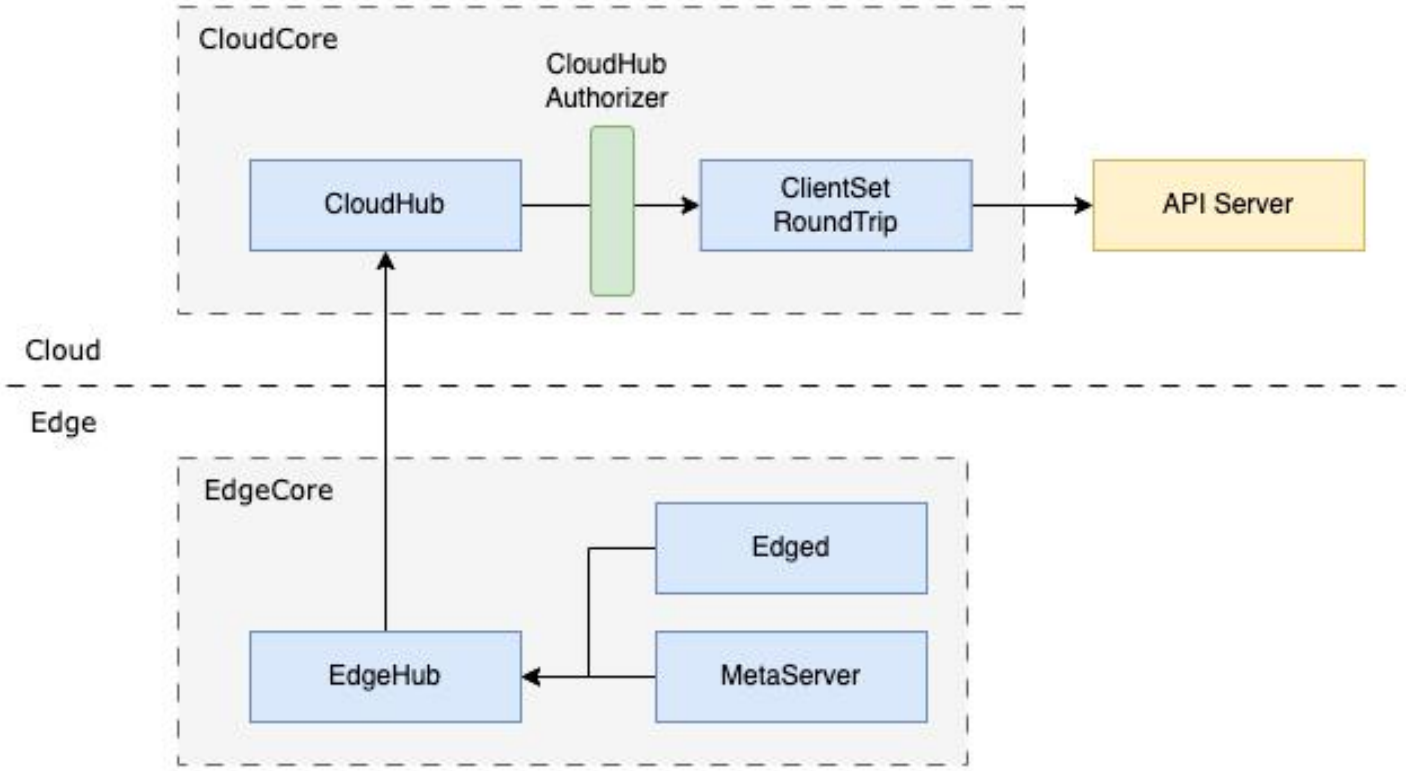








Authorization of HUB



Node Authorization of K8s

Read operations:

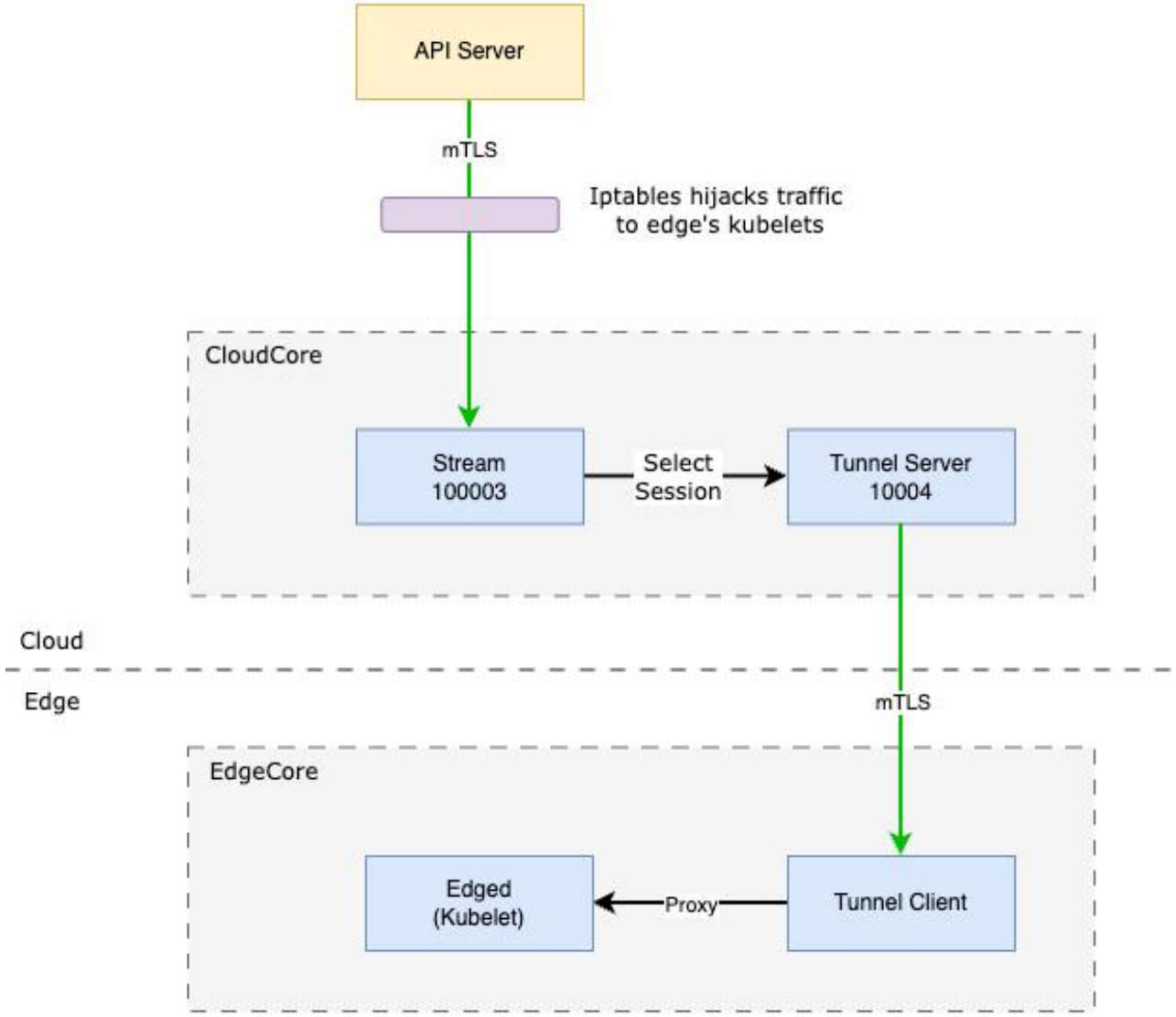
- services
- endpoints
- nodes
- pods
- secrets, configmaps, persistent volume claims and persistent volumes related to pods bound to the kubelet's node

Write operations:

- nodes and node status (enable the NodeRestriction admission plugin to limit a kubelet to modify its own node)
- pods and pod status (enable the NodeRestriction admission plugin to limit a kubelet to modify pods bound to itself)
- events

KubeEdge has customized its own authorizer to filters the read and writer operations of its CRDs.
And define a RoundTrip to injects the credential request headers(i.e., **system:nodes** group and a username of **system:node:<nodename>**) before requesting the client-go.

Stream Tunnel



KubeEdge provides an iptables rule in the control node where the API Server resides to hijack its traffic to the edge node.

Tunnel Server caches client sessions for the Stream module to use.

External traffic is secured with mTLS(i.e., API Server -> Stream and Tunnel Server -> Tunnel Client).

Tunnel Client uses K8s SpdyRoundTripper to proxy the Kubelet stream APIs.

Roadmap:

Use eBPF to hijack API server traffic to kubelet.

Network Coding for Cloud Computing

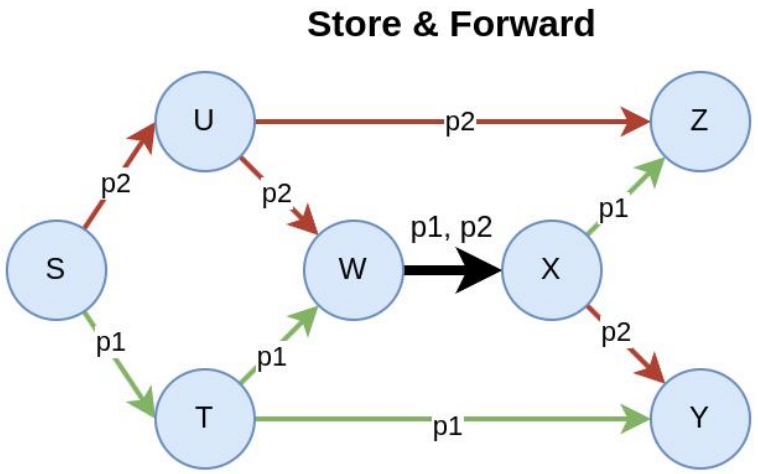
n-hop

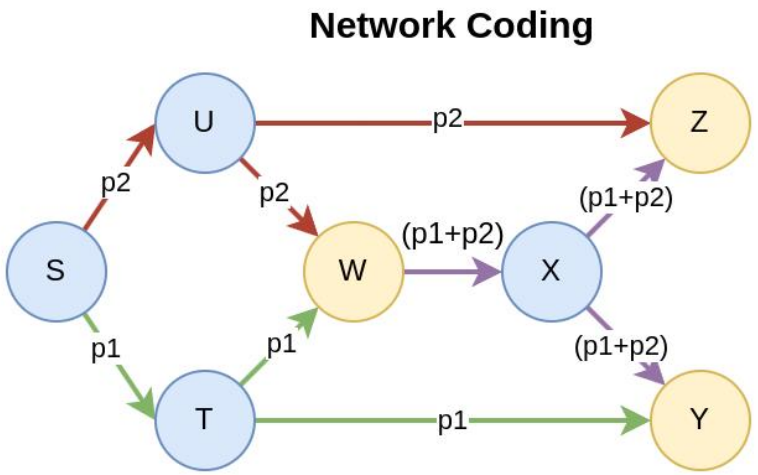
Prof. Shenghao Yang

Prof. Raymond Yeung

- n-hop technologies is a tech start-up focusing on research, development and commercialization of our invention: “Network Coding”.
- Our founders are pioneers of the field of Network Coding
- Network coding is a paradigm shift in the field of data communication and networking. Using network coding, we can improve the efficiency, reliability, and throughput of data transmission.
- The company currently focuses on implementing the technology in Cloud Infrastructure (*kubernetes, kube-edge …*)

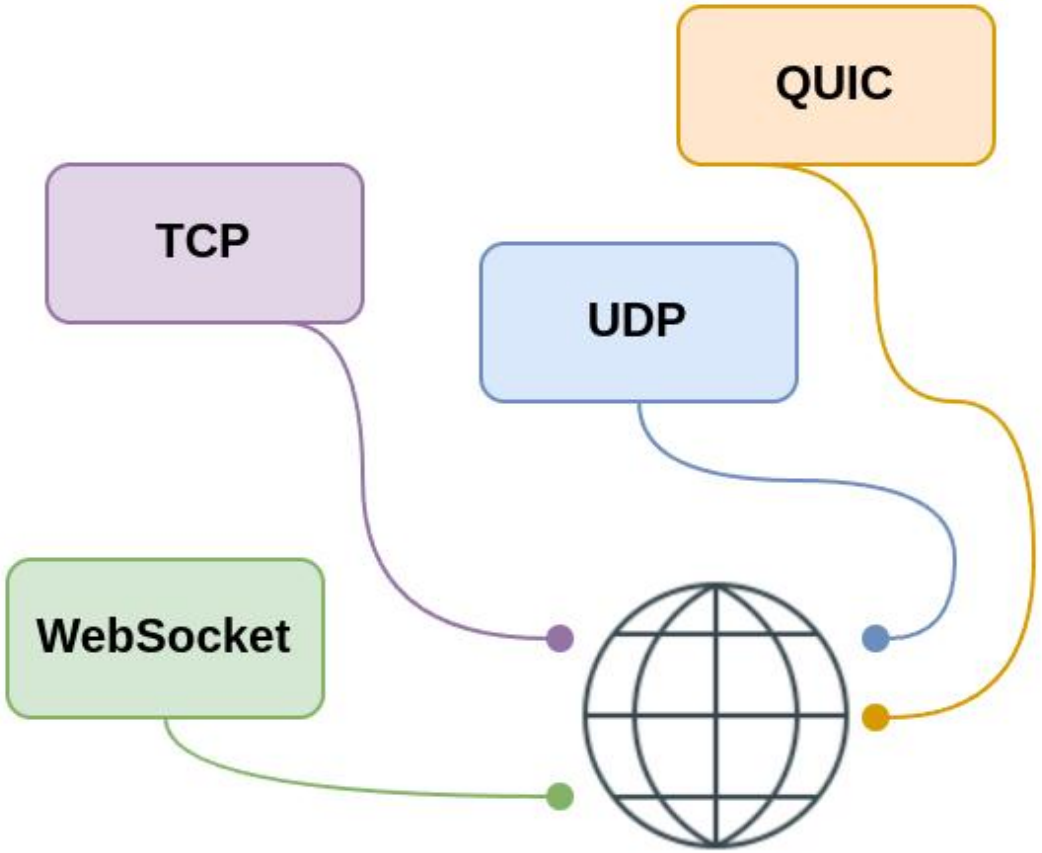
Network Coding introduction





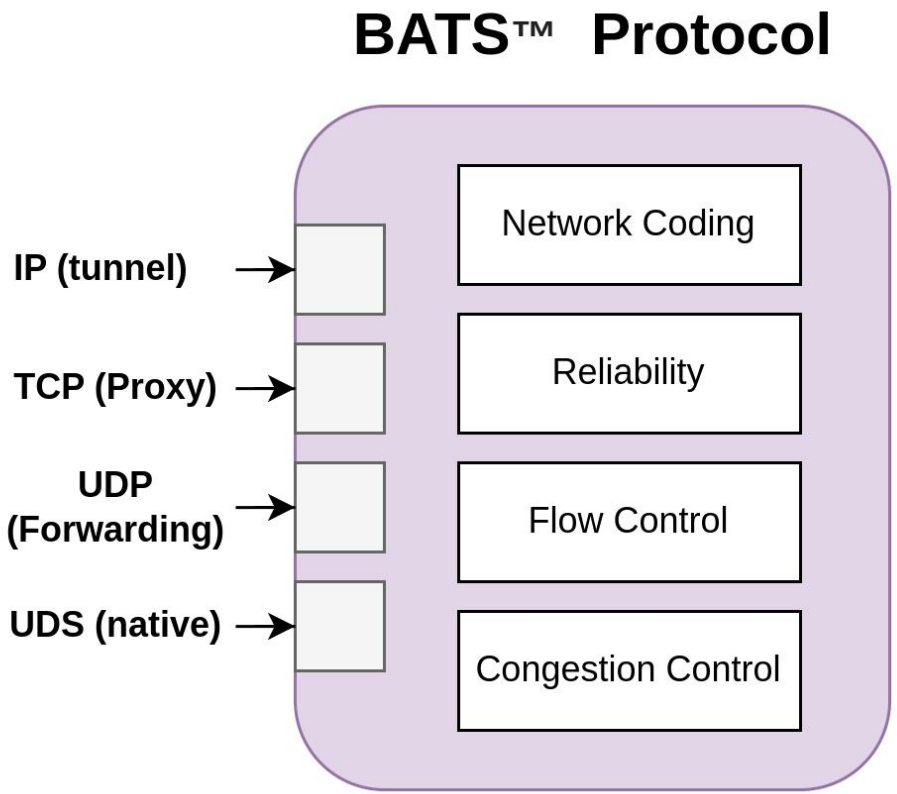
- Network coding is a technique that offers better reliability, efficiency and latency tradeoff than the existing networking protocols at the cost of computation
- Invented in late 90's, but it took some time for computing and coding techniques to enable practical usage
- Through experience, we realize the largest gain is in Network Coding-aware protocols, and we have developed one called BATS™

Protocols Challenges Overview



- **TCP:**
  head of line blocking, tail latency, ossification
- **WebSocket:**
  Rely on TCP, same issues
- **UDP:**
  unreliable, no congestion control
- **QUIC:**
  No network coding implementation to enhance reliability and congestion and flow control are not designed for network coding
- …

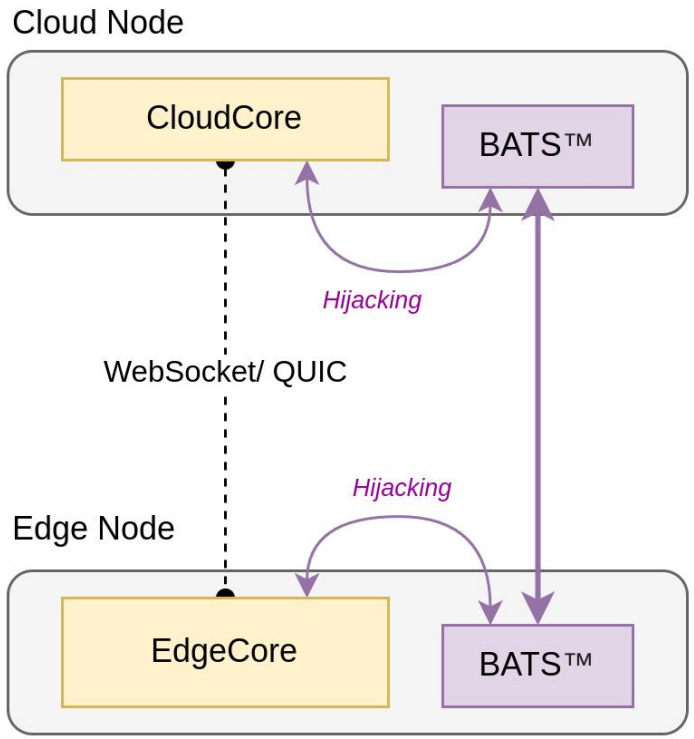
BATS™ Protocol



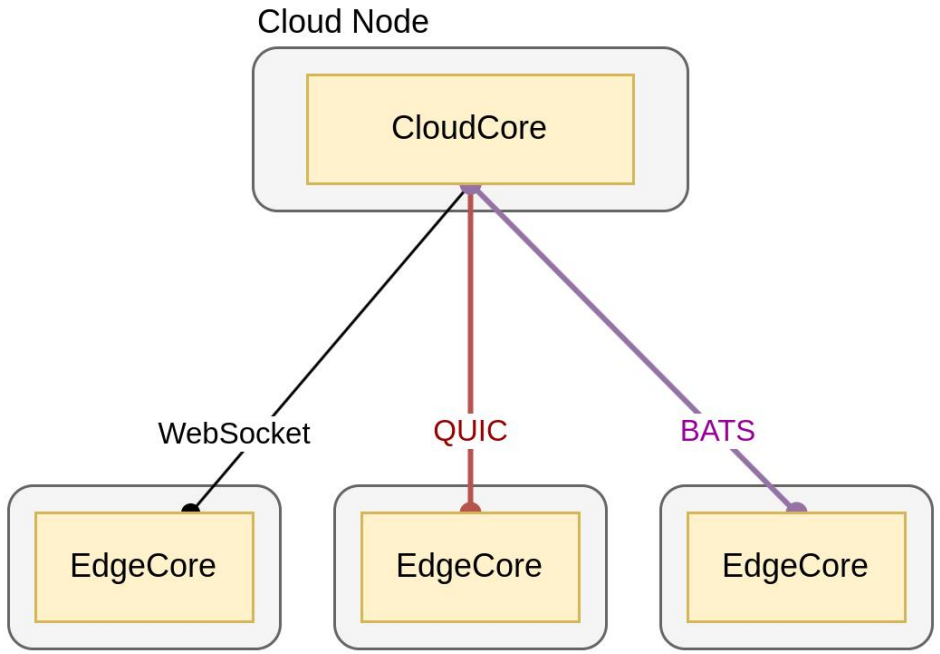
- Similarities to QUIC:
  - UDP as transport
  - User Space
  - Not Ossified
- Features:
  - UDS socket
  - IP Tunneling *(TUN interface)*
  - TCP Proxy over BATS™ transport
  - UDP Port Forwarding
  - Innovative Network Coding algorithm
  - Network Coding aware Congestion and flows Control

Network Coding in KubeEdge

QUIC or Websocket over BATS™ protocol



Network coding protocol as transport



KubeCon | CloudNativeCon

THE LINUX FOUNDATION OPEN SOURCE SUMMIT

AI_dev Open Source GenAI & ML Summit

China 2024

Thanks & Questions

Join Us

- Contribute to KubeEdge(welcome to star)
  https://github.com/kubeedge/kubeedge
- KubeEdge community Slack
  https://kubeedge.io/docs/community/slack
- n-hop website
  https://n-hop.com