OPEN SOURCE NETWORKING DAYS

Singapore

Service Proxy, Container Networking & K8s

Hongjun Ni
Intel
Email: hongjun.ni@intel.com

Acknowledgement:
Pierre Pfister, Jerome Tollet @Cisco
John DiGiglio, Ray Kinsella @Intel

Agenda

- What is in Cloud Native Networking?
- Problem and Challenge
- Proposed Architecture
- Existing vs Proposed Solution
- Why Choosing FD.io
- Service Proxy Implementation
- Key Takeaway

What’s in Cloud Native Networking?

Control Plane

Control Plane:

- Assigns IPs (from a pool given to each workload)
- Distributes routing information (i.e. how to get to this workload)
- Distributes policy (e.g. who can connect to whom)

Data Plane

Data Plane:

For each packet to/from the workload:

- Enforces policy
- Forwards it to the right destination

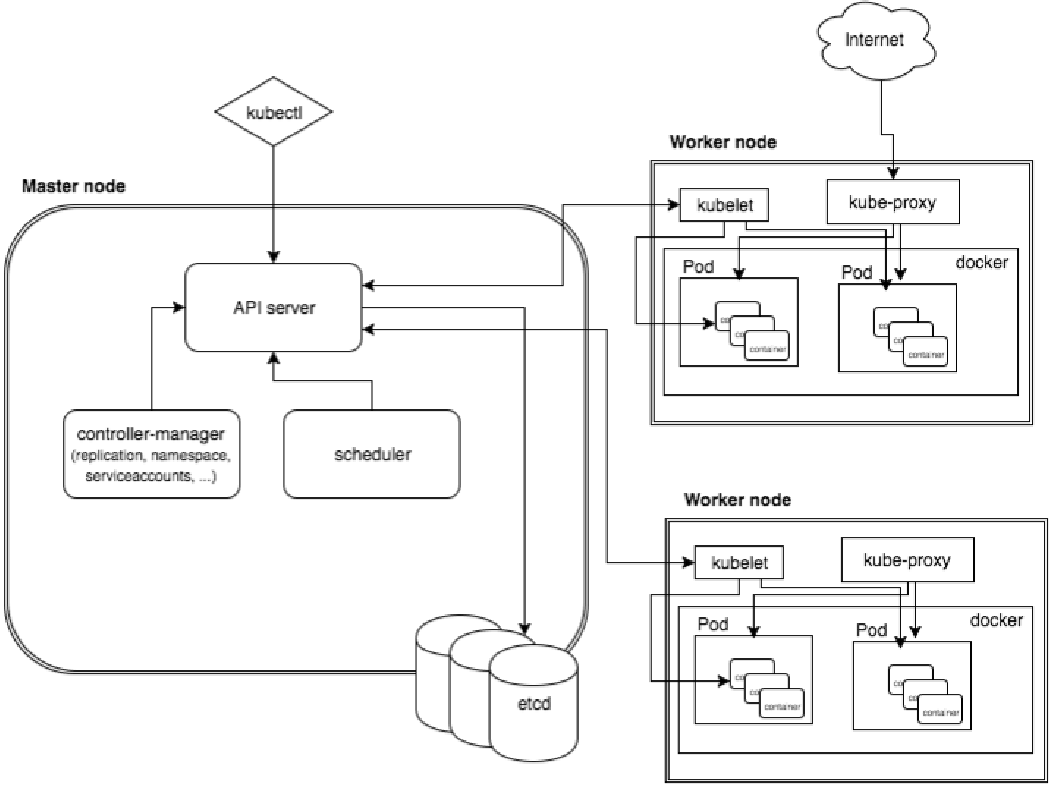
Reference: https://www.cncf.io/wp-content/uploads/2017/11/CNCF-Networking-Webinar-final-1-1.pdf

Problem in Cloud Native Networking

- Cumbersome configuration
- Container network cannot cross zones and regions
- Forwarding performance is poor
- Limited scalability
- Unwanted communication between services
- Failure recovery difficult
- Long convergence time
- Monitoring and Troubleshooting is not easy

Reference: https://www.cncf.io/wp-content/uploads/2017/11/CNCF-Networking-Webinar-final-1-1.pdf

Kubernetes Architecture



Master Node

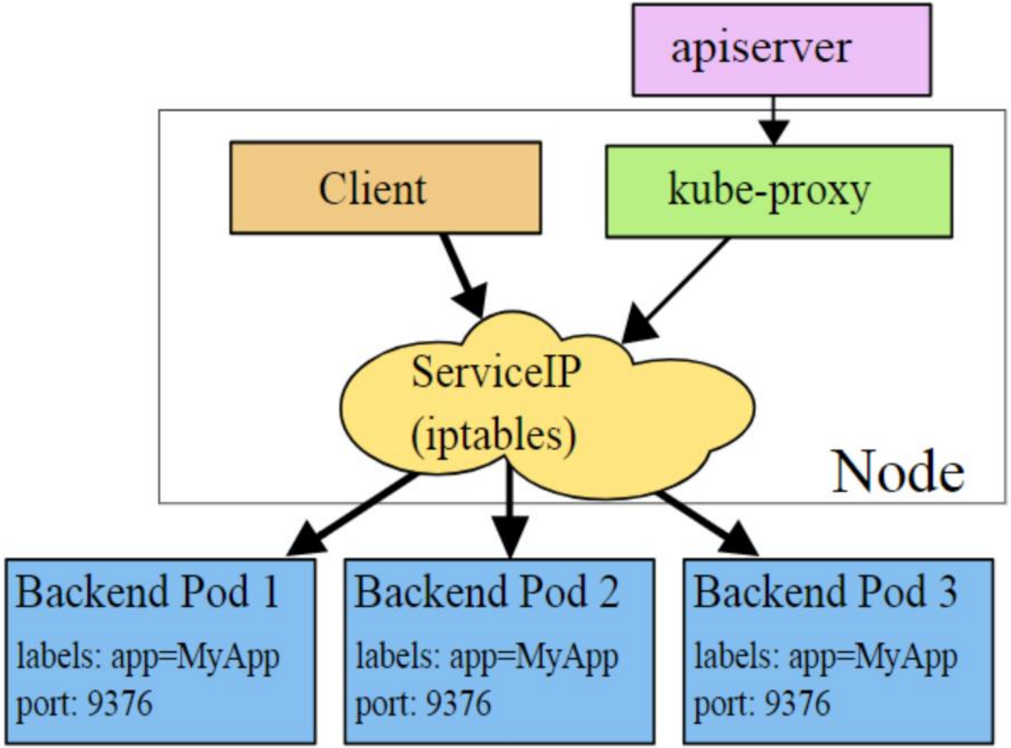
- Responsible for the management of Kubernetes cluster.
- Entry point of all administrative tasks.
- Taking care of orchestrating the worker nodes.

Worker node

- The pods are run here.
- Contains all the necessary services to manage the networking between the containers.
- Communicate with the master node.
- Assign resources to the containers scheduled.

Reference: https://x-team.com/blog/introduction-kubernetes-architecture/

Challenge With Current Solution



Linux kernel solution:

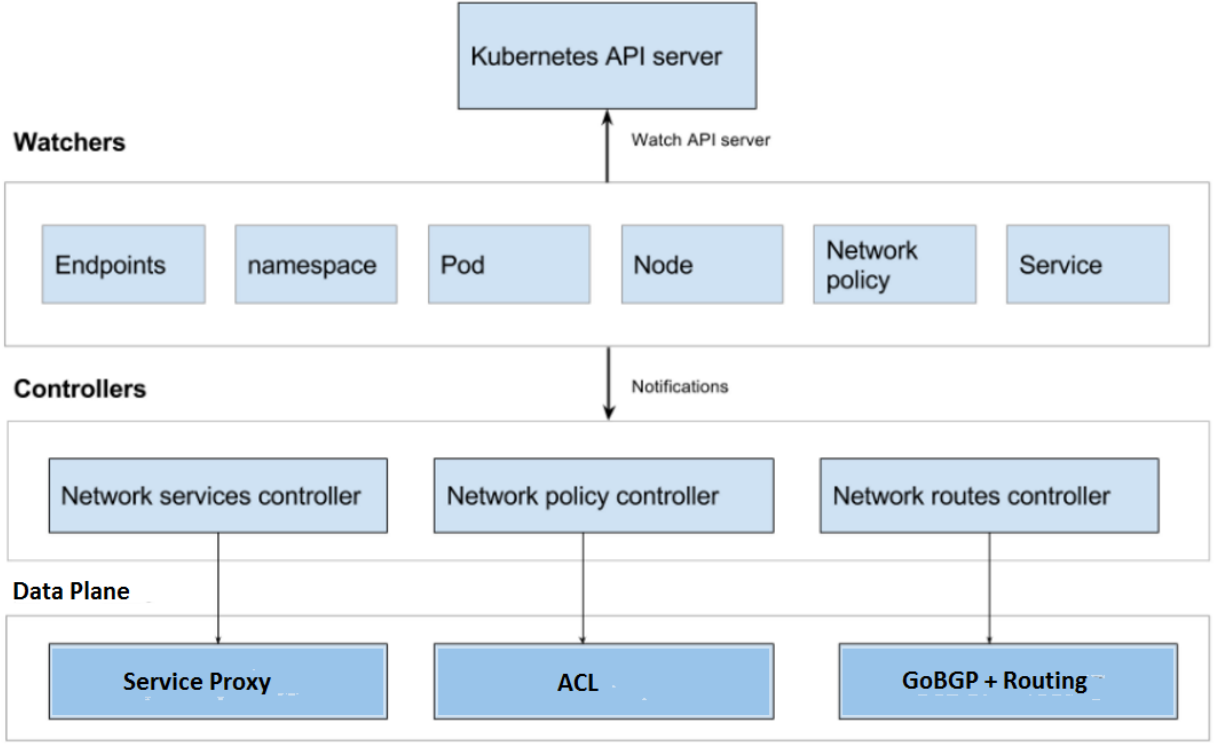
- Watches service and endpoints
- Installs iptables/IPVS rules
- Captures traffic and selects pod
- Redirects traffic to chosen pods

Problems:

- Uses load balancing on iptables/IPVS
- Uses NAT on iptables/IPVS
- Communication via VETH
- Performance degrades when service/endpoint pairs increase iptables entries.

Reference: https://kubernetes.io/docs/concepts/services-networking/service

Proposed Architecture

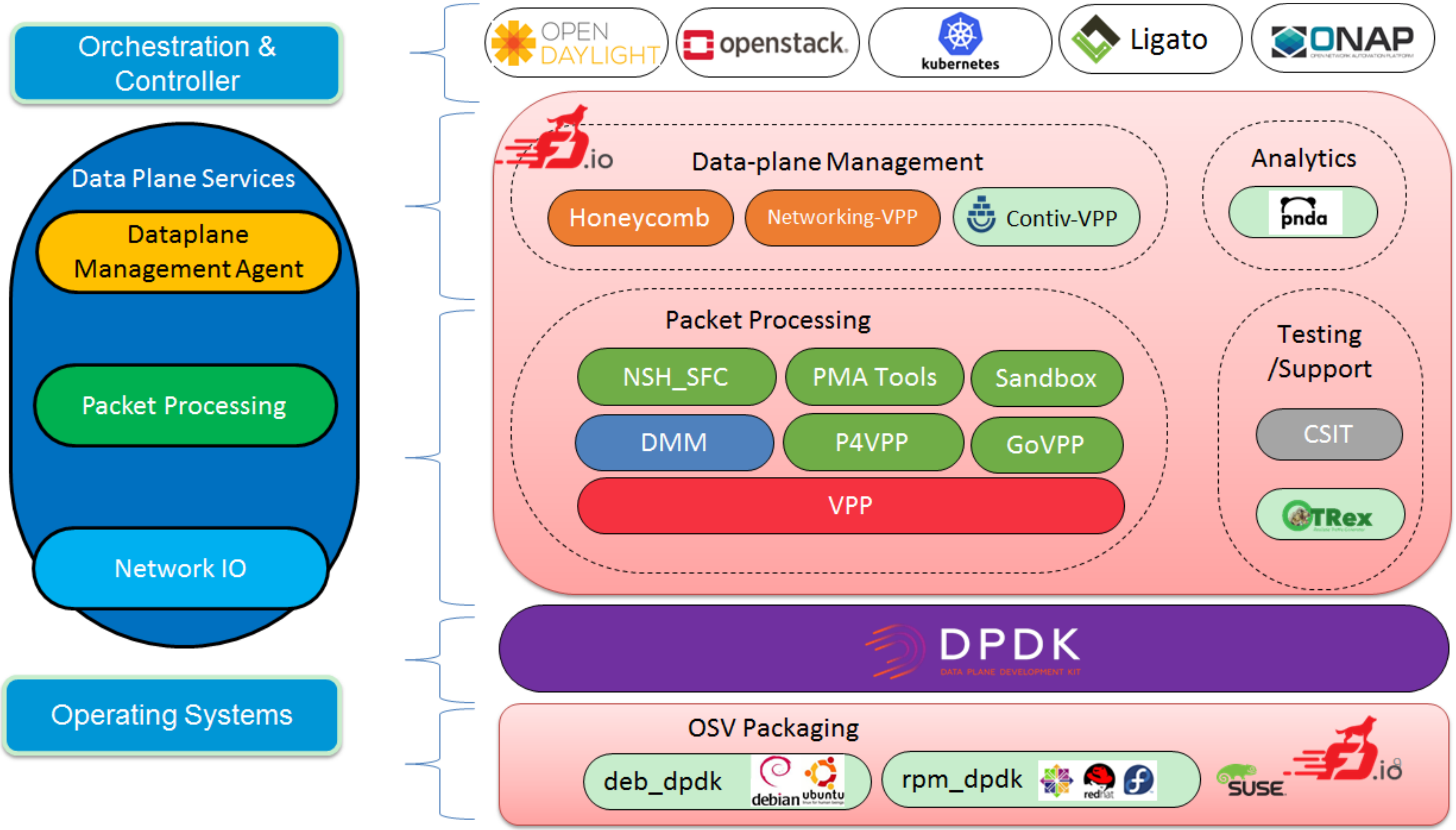


- Running on VPP and DPDK
- Policy based on VPP ACL
- Integrate with GoBGP or FRR
- Routing based on VPP FIB

Existing vs Proposed Solution

	Existing Solution	Proposed Solution
Solution Stack	Linux Kernel Stack	User Space Project, VPP & DPDK
Policy Enforcement	Iptables + ipset	VPP ACL
Node Load Balancing	Iptables, IPVS	VPP kube-proxy
Connection Tracking	Iptables, IPVS	VPP kube-proxy
DNAT and SNAT	Iptables, IPVS	VPP kube-proxy
Communication between Host and Container	Via VETH	Via vhost-user or memif
External Load Balancer	Via CSP’ load balancer	Via VPP load balancer
Performance	Limited	Very high
Scaling	Limited	Very well

Why Choosing FD.io?



Orchestration & Controller

Data Plane Services

Dataplane Management Agent

Packet Processing

Network IO

Operating Systems

FD.io

Data-plane Management

Honeycomb

Networking-VPP

Contiv-VPP

Analytics

Packet Processing

NSH_SFC

PMA Tools

Sandbox

DMM

P4VPP

GoVPP

VPP

Testing /Support

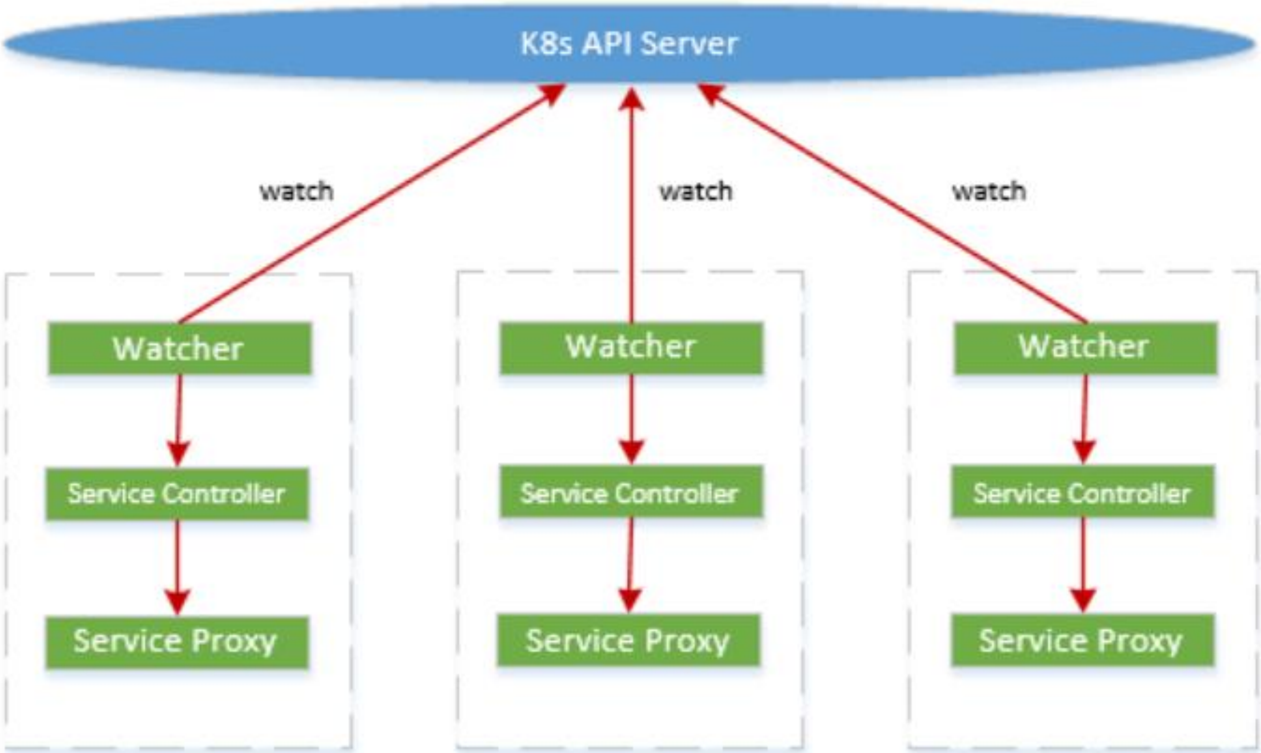
CSIT

OSV Packaging

deb_dpdk

rpm_dpdk

Service Proxy Architecture

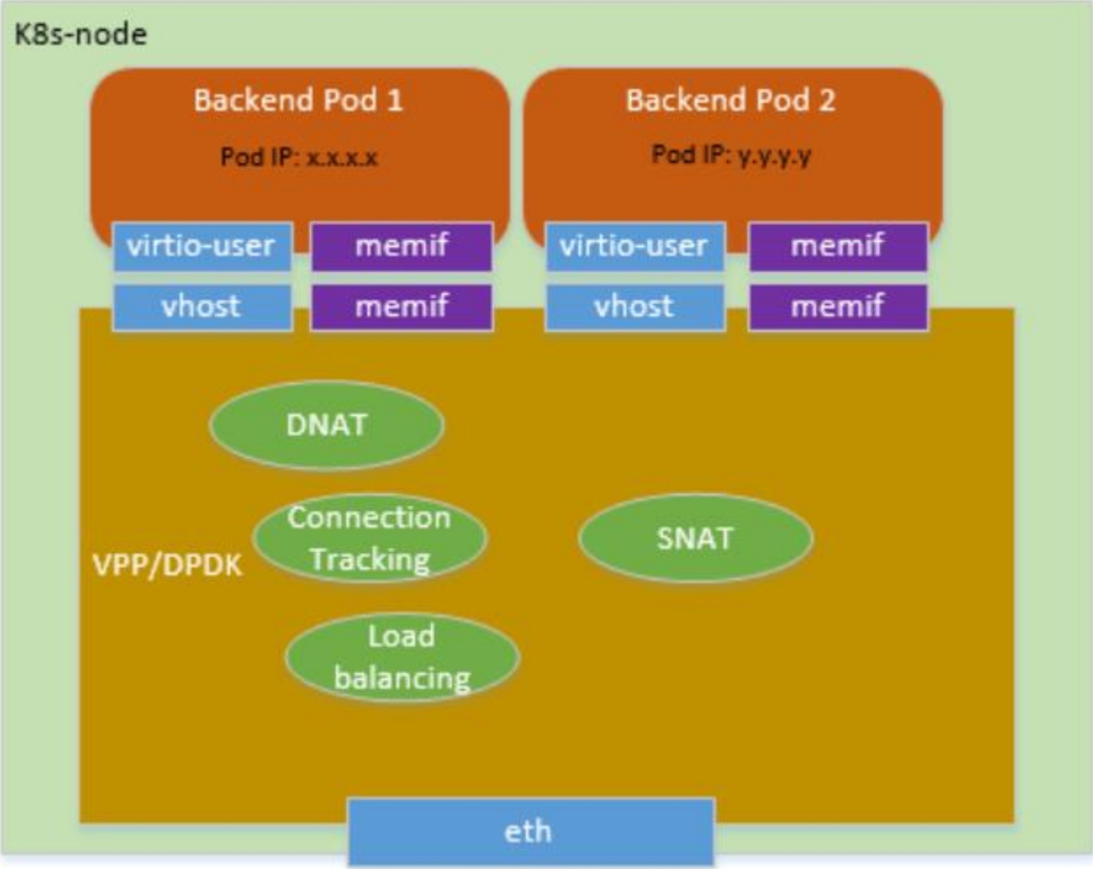


Services Controller:

1). Reads the services and endpoints information from K8s API server

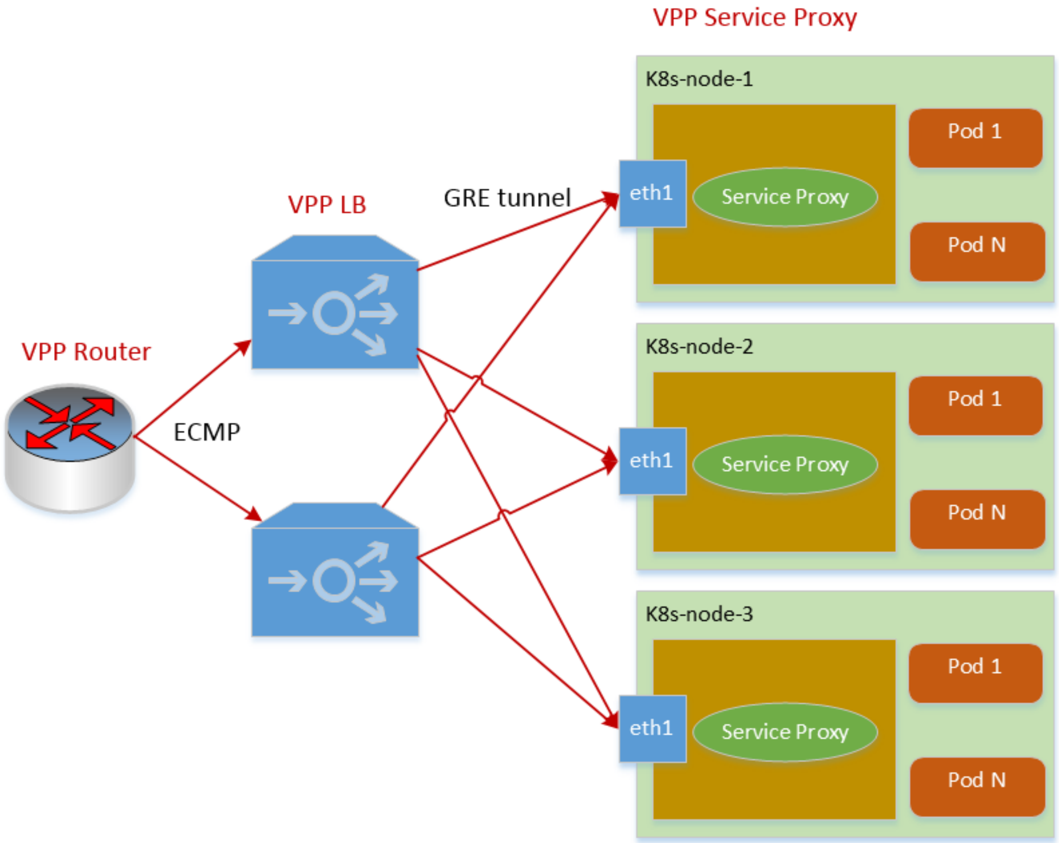
2). Configures Service Proxy on each cluster node.

Service Proxy Implementation



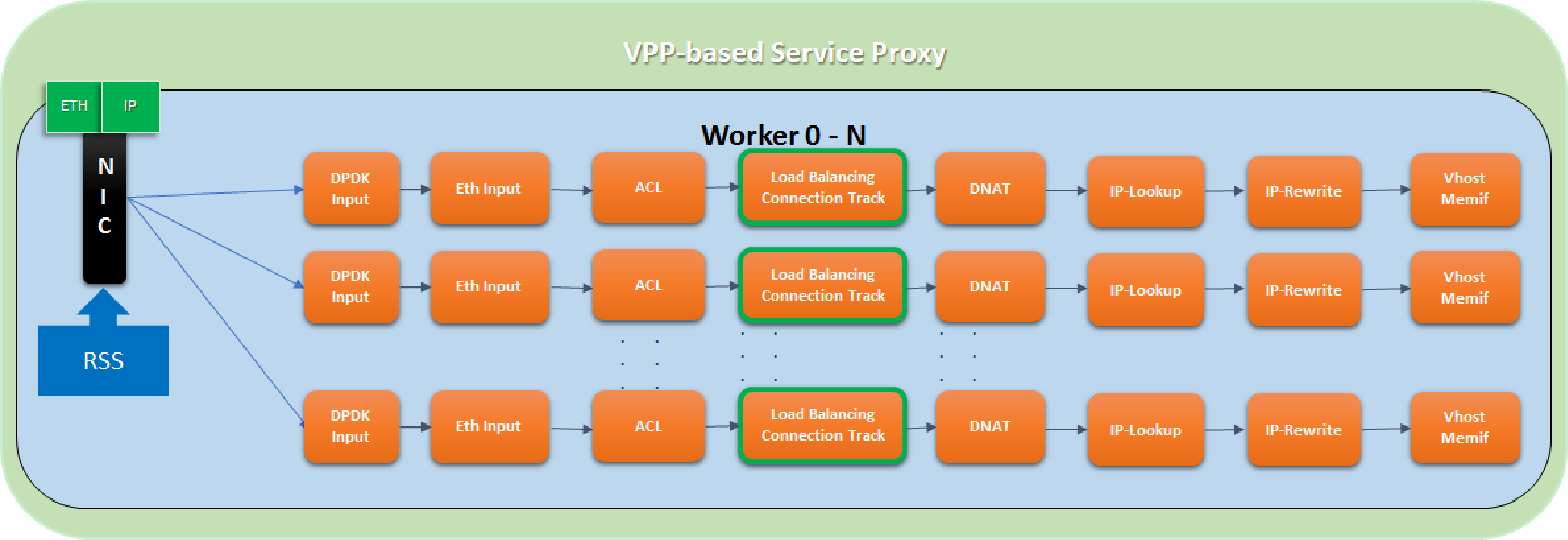
- Distributes traffic evenly
- Supports two interface types
  - ➢ vhost and virtio-user
  - ➢ memif
- Load Balancing optimized based on SSE4.2 to improve performance.
- Supports three service types:
  - ➢ Cluster IP
  - ➢ NodePort
  - ➢ External Load Balancer

Integrates External Load Balancer



- Router, Load Balancer and Service Proxy are supported on VPP.
- On Router, will enable ECMP feature.
- VPP Load Balancer distributes traffic and encapsulates packets via GRE tunnel.
- On K8s node, it removes GRE tunnel and goes through Service Proxy to distribute traffic to chosen pod.

Multithread Support



- RSS enables traffic associated with one connection to a given thread.
- Load balancing and connection track redirects traffic to a chosen pod.

Key Takeaway

- A solution offering high performance K8s Service Proxy.
- Implementation ready for K8s container networking.
- Load Balancing distributes traffic to pods almost evenly.
- Connection tracking supports connection persistence.
- Consistent hashing ensures resilience to pod changes.
- External Load Balancer in support of node-level scaling.
- Multithread support for pod-level scaling.

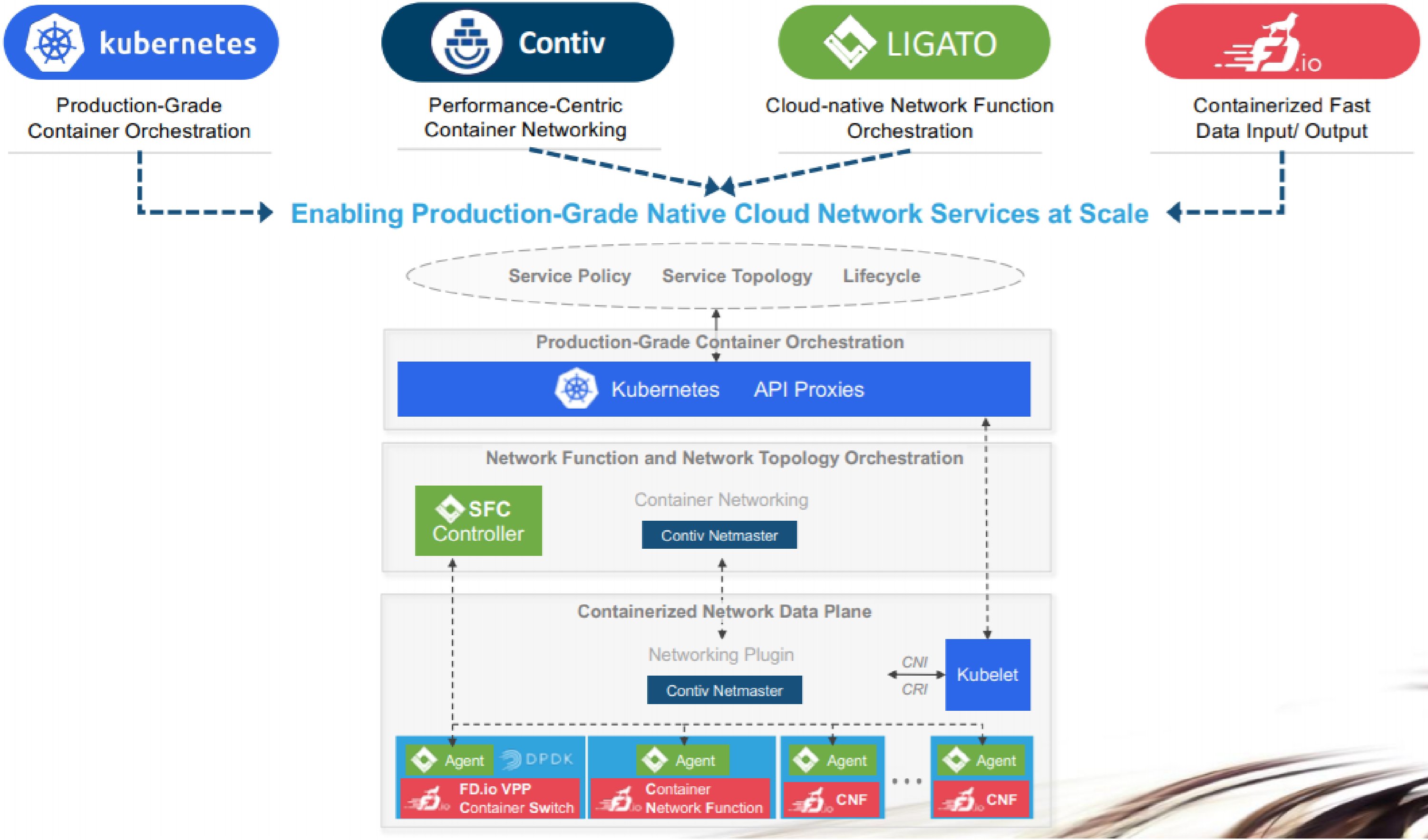
Thank you !

Q & A

Email : hongjun.ni@intel.com

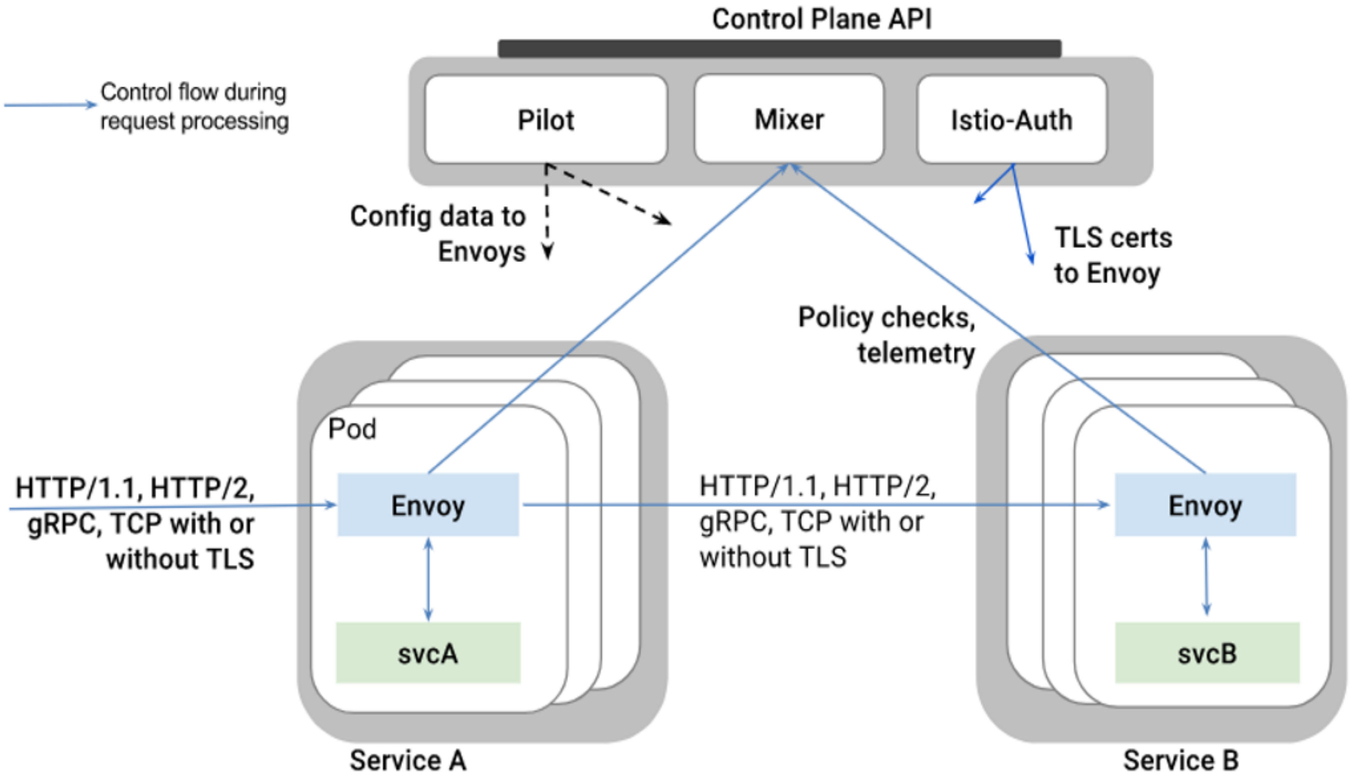
Backup Slides

Ligato & Contiv-VPP



kubernetes
Production-Grade Container Orchestration
Contiv
Performance-Centric Container Networking
LIGATO
Cloud-native Network Function Orchestration
.io
Containerized Fast Data Input/ Output
Enabling Production-Grade Native Cloud Network Services at Scale
Service Policy
Service Topology
Lifecycle
Production-Grade Container Orchestration
Kubernetes
API Proxies
Network Function and Network Topology Orchestration
SFC Controller
Container Networking
Contiv Netmaster
Containerized Network Data Plane
Networking Plugin
Contiv Netmaster
CNI
CRI
Kubelet
Agent
DPDK
FD.io VPP Container Switch
Agent
Container Network Function
Agent
CNF
...
Agent
CNF

Istio & Envoy

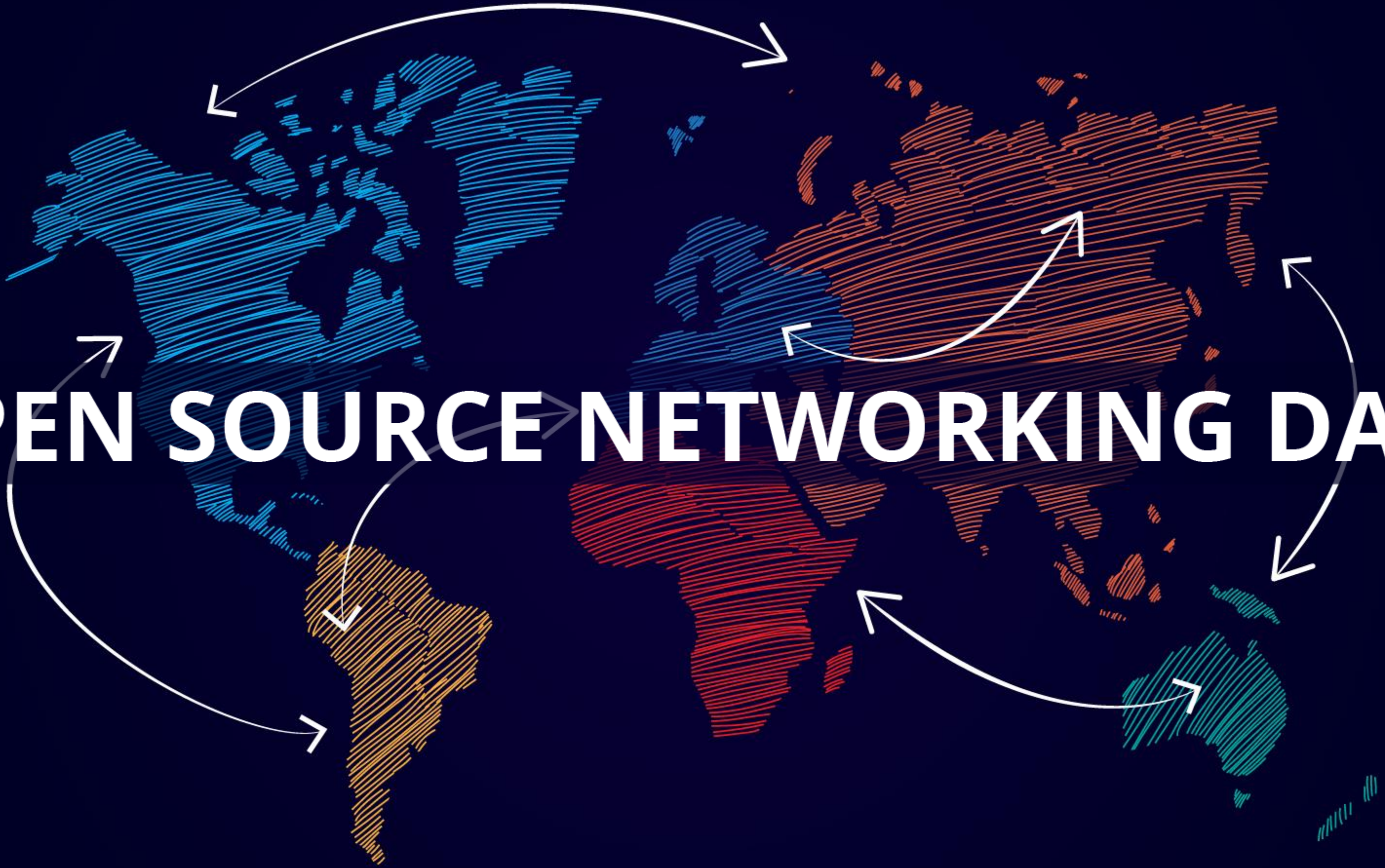


An Istio service mesh is logically split into a data plane and a control plane.

- The data plane is composed of a set of intelligent proxies (Envoy) deployed as sidecars that mediate and control all network communication between microservices.

- The control plane is responsible for managing and configuring proxies to route traffic, as well as enforcing policies at runtime.

Reference: https://istio.io/docs/concepts/what-is-istio/



OPEN SOURCE NETWORKING DAYS