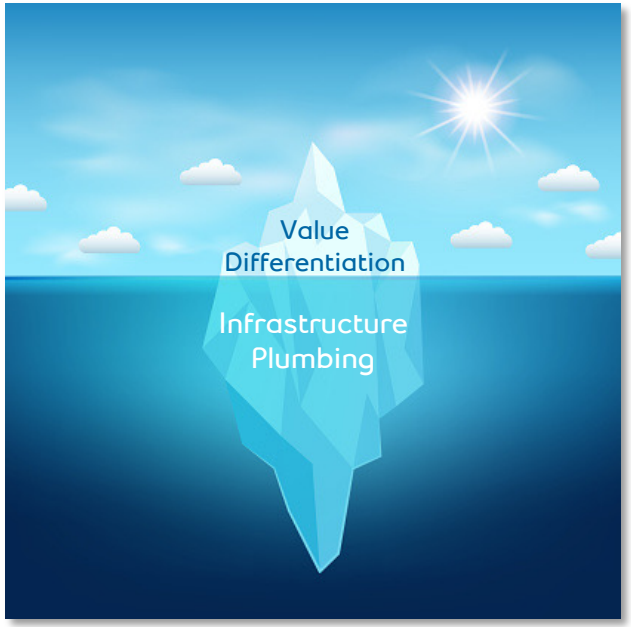
Cloudifying SRv6

Daniel Bernier | Bell Canada
Email: daniel.bernier@bell.ca

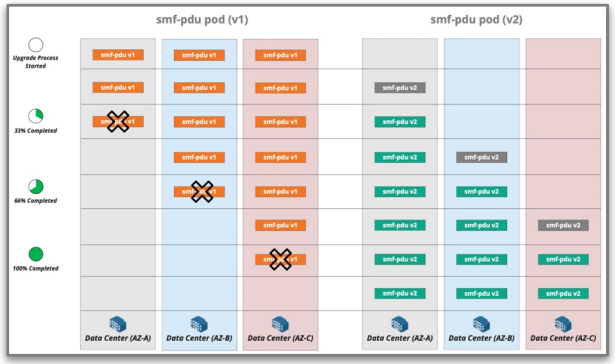
Bell

Why this Movement ?

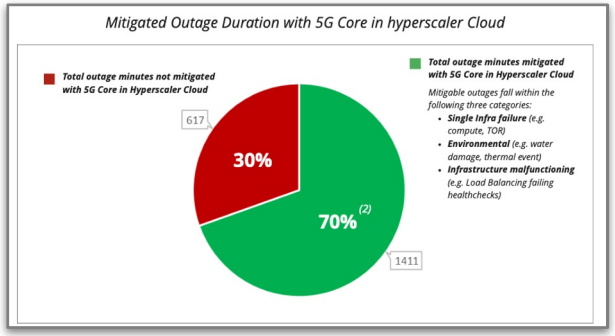
- All-in-one service providers are disaggregating into subsets of specialized providers.
- To switch efforts towards **rapid innovation** or **new product delivery** and more focus towards **research**.
- Delegate operations to focus more on Planning, Design and Architecture or Business Strategy *



- Adopt fit-for-purpose public cloud technology
- Embrace true hyper scaler practices to innovate and scale
- Driving new services at a velocity, volume and cost not previously achievable
- Shift from technology delivery to product delivery.



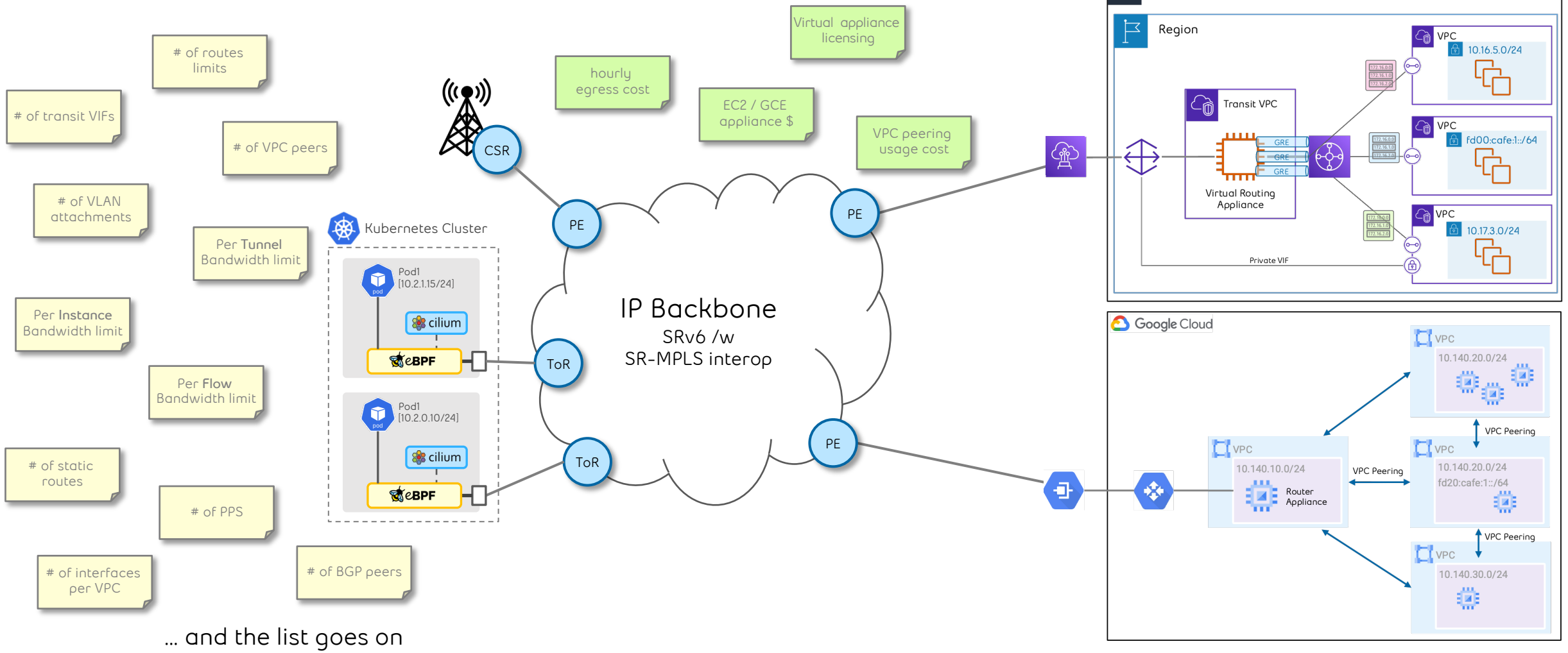
- Integrated/deployed a 5GC (start to finish) in less than a week compared to months.
- Shift from $/Bps to complete TCO viewpoint through cloud-native operations.
- An estimated 70% of mitigated outage minutes if leveraging hyper scaler clouds for 5G Core..



* https://virtualizationreview.com/articles/2019/10/21/cloud-trends.aspx

Into the realm of Limits, Quotas and FinOps

- There is no such thing as a “free meal” when thinking about cloud networking … **literally**.
- Effectively extending an operator network to the cloud is nothing short of **epic**.
- Most 3rd party cloud networking solutions usually **add even more** complexity.



… and the list goes on

What if we tried to drastically simplify cloud networking instead ?

3.2 Gaps & Challenges in Today's Communication Services

The Network 2030 initiative is a structured approach to defining the capabilities of networks and corresponding communication services for the decade following 2030. The goal is to have networks ready for the market verticals that will utilize emerging technologies in 2030. Network 2030 extrapolates from what we know about technologies and develops a vision of the new media, new services, and new infrastructure. For this, we outline a few areas of importance to address towards building this vision.

Lacking service-network interaction: Failure to find an ordered and healthy relationship between the applications and the network has been a sticking point. Connectivity is just one of the workflows involved in application logic. It is a service consumed, with reachability being the only explicit means of setting up the application behaviour in the networks. However, a number of services offered by networks are not obvious to the end user. These include reliability of the network fabric, broadcast or multicast, integrity and security of data delivered, and network level awareness of congestion, capacity, and latency. Accommodating for such capabilities directly in the networks have been much of the focus of industry for the past thirty years. *Due to lack of direct support for such services through proper interfaces to the network, application developers have been left in a limbo, designing for every conceivable possibility of network failures and outages*. Many of these services are controlled over end-to-end interfaces between the endpoints using the transport (TCP) layer which is another example of free-form evolution aiming to solve the problems of the network without sufficient assistance from the networks.

Network 2030 – A Blueprint of Technology, Applications and Market Drivers Towards the Year 2030 and Beyond

Get the Network Out of the Way

Justin Pietsch Follow

Jan 21 · 15 min read

Getting the network out of the way has been very important for me in my thinking about networks, and is an easy way to help talk about a bunch of very important concepts, decisions, and arguments. This can sound trite, obvious, or insulting. In this post I'll try to describe the concepts I'm talking about and illustrate with some of most important examples in my career.

The more that the network is noticed the worse things are for everyone. Often times, especially when the network is noticed, networking and network engineers are thought of negatively. Instead, if you think of it as a challenge it can help you focus on making a great network. You can think about your goals: how important it is to keep the network working well, to not disrupt the business, and to be able to keep up with any changes that the business needs.

Invisinets: Removing Networking from Cloud Networks

Sarah McClure, Zeke Medley*, Deepak Bansal*, Karthick Jayaraman*, Ashok Narayanan†, Jitendra Padhye*, Sylvia Ratnasamy*†, Anees Shaikh†, and Rishabh Tewari**
**UC Berkeley †Google *Microsoft*

"In other words, *we believe that the best way for tenants to think about networking is to not think about networking at all.*"

https://www.usenix.org/system/files/nsdi23-mcclure.pdf

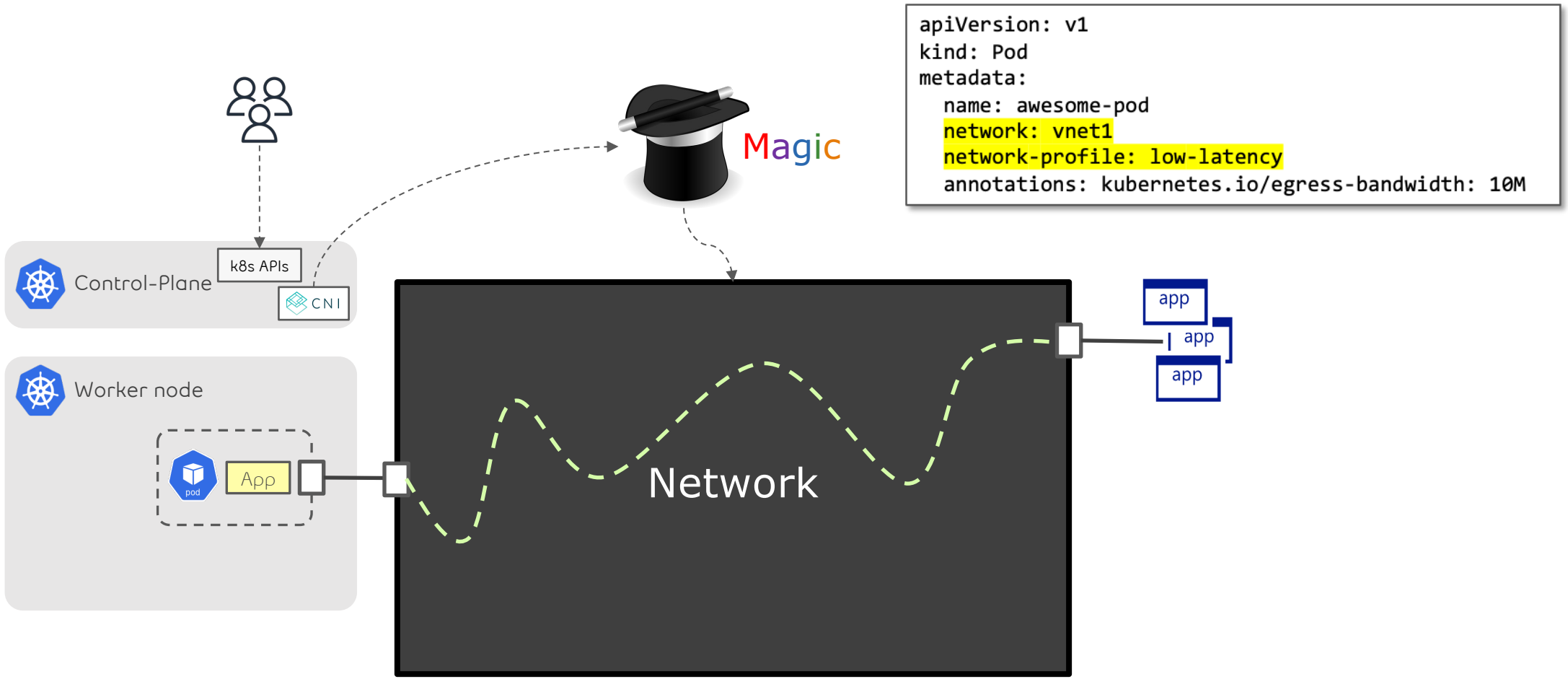
From Software Defined to Application Defined

- SDN is still fundamentally about packet delivery and connection management
  - Naturally, networking community optimizes for these concepts
- Application developers do not care about packets or connections; even the concepts are meaningless

Amin Vahdat - https://conferences.sigcomm.org/sigcomm/2021/workshop-nai.html#keynote-talk-2

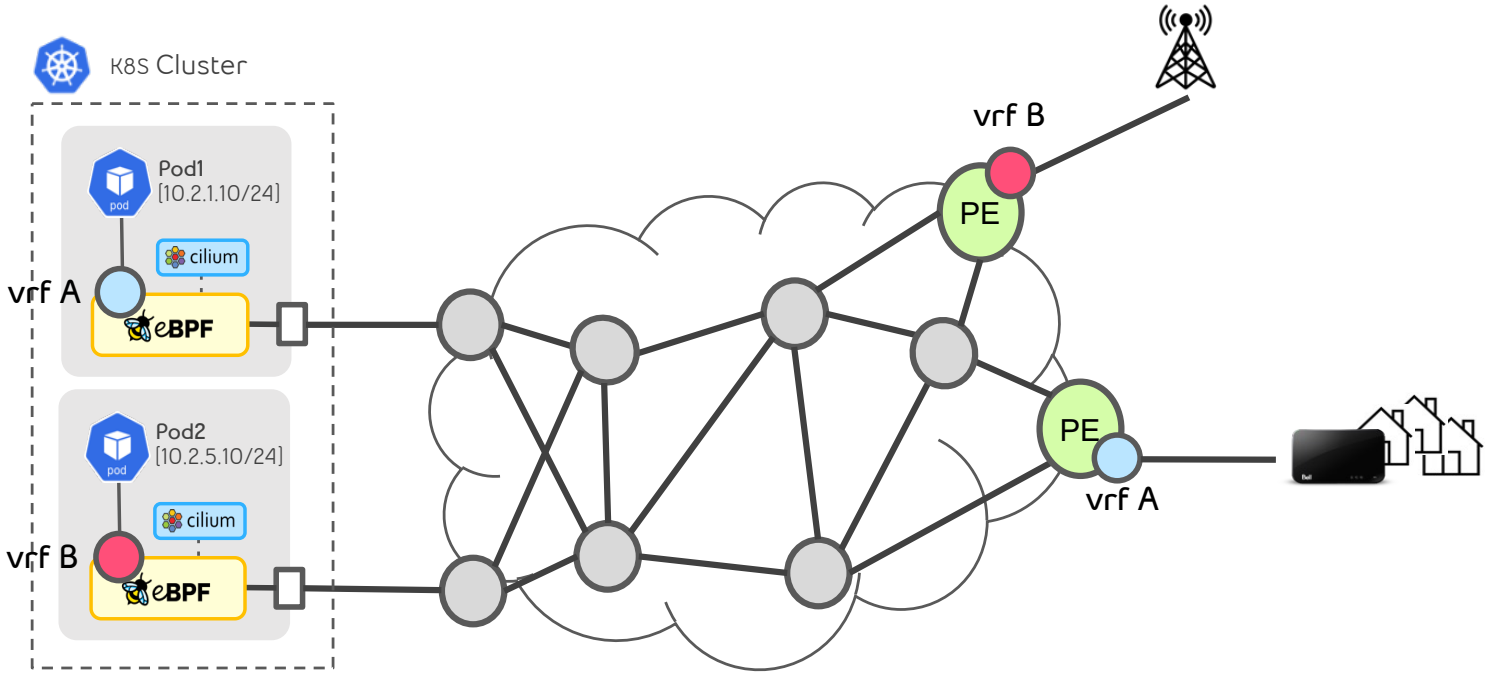
Because in a utopian world !

- This is what we aspire to … simple, declarative, developer friendly
- Evolving towards a *network spec* which represents more than L2/L3 definitions

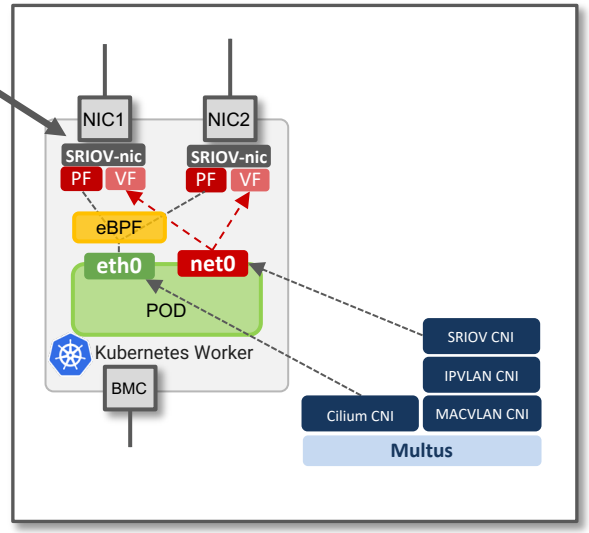


Because in a utopian world !

- Typical problem we are all facing when connecting our networks to Kubernetes
- It was there before, got thrown into the spotlight with Telco workloads
- We are all being ingenious (although quite complicated) at solving the problem



Do we really need all of this ?!

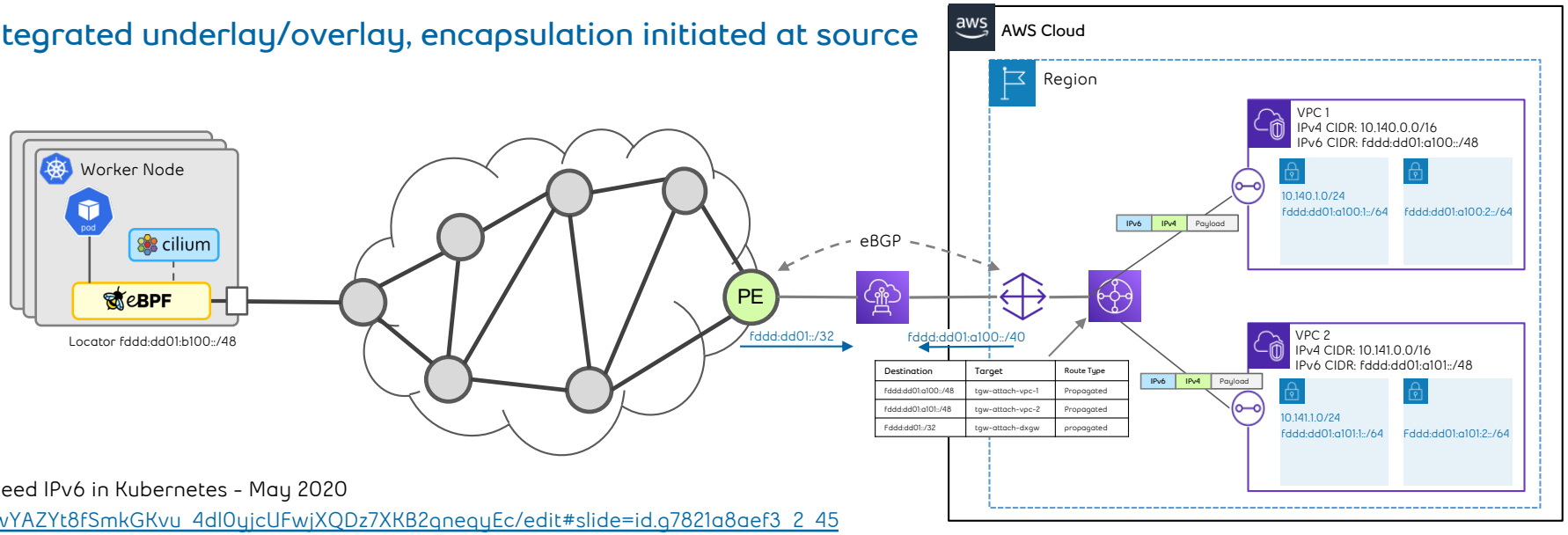


What if we were disruptive in our approach and use emerging tech to do so ?

Leveraging SRv6 to evolve multi-cloud connectivity

- Let's assume everything was IPv6 … would we need to use overlay VPNs ? *
- Unfortunately, existing technical or business constraints still require us to perform some sort of segmentation.
- SRv6 lets us achieve both end-to-end IPv6 **WITH** VPN service capabilities.
- When deployed at the upmost edge of a flow (host) SRv6 allows for **MASSIVE** cloud networking simplification.

SIMPLICITY	Hyperscaler networks only needs to carry locator address space, no added complexity
COST EFFICIENCY	Reduced need for add-on cloud features or extending quota limits.
PERFORMANCE	Even without flow-label, SRv6 allows optimized flow performance (1 flow per uDT/DX or END function)
OBSERVABILITY	Integrated underlay/overlay, encapsulation initiated at source



* Ref: Nicola Leiva, Three reasons why we need IPv6 in Kubernetes - May 2020
https://docs.google.com/presentation/d/1wYAZYt8fSmkGKvu_4dl0yjcUFwjXQDz7XKB2qneqyEc/edit#slide=id.g7821a8aef3_2_45

Enabling SRv6 on Telco private clouds

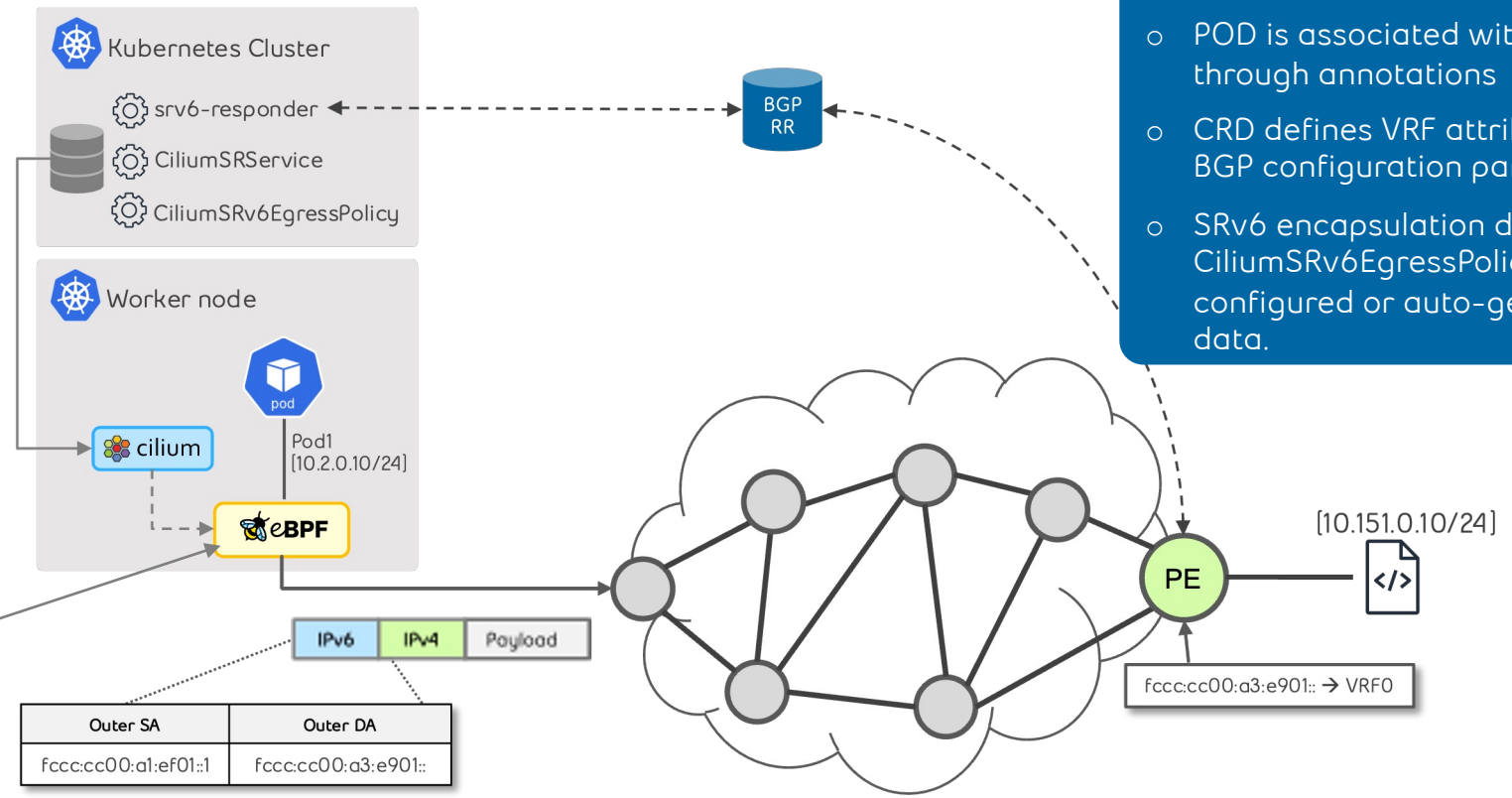
```
ipv6:
  enabled: true
srv6:
  enabled: true
  encapMode: reduced
bgpControlPlane:
  enabled: true
```

```
apiVersion: "cilium.io/v2alpha1"
kind: CiliumBGPPeeringPolicy
metadata:
  name: pe0
spec:
  nodeSelector:
     matchLabels:
     kubernetes.io/hostname: srv6-responder
  virtualRouters:
  - localASN: 64577
     exportPodCIDR: true
     mapSRv6VRFs: true
     neighbors:
     - peerAddress: "fccc:cc00:1:1"
     peerASN: 64577
```

```
apiVersion: cilium.io/v2alpha1
metadata:kind: CiliumSRv6VRF
name: vrf0
spec:
  vrfID: 1
  importRouteTarget: "64577:661"
  exportRouteTarget: "64577:661"
  rules:
  - selectors:
    - podSelector:
        matchLabels:
        vrf: vrf0
    destinationCIDRs:
    - 0.0.0.0/0
```

```
apiVersion: cilium.io/v2alpha1
kind: CiliumSRv6EgressPolicy
metadata:
  creationTimestamp: "2022-05-16T02:03:15Z"
  generation: 1
  name: bgp-control-plane-…
  resourceVersion: "791"
  uid: f8519c91-d6d3-4cc8-941d-e5efd69e4317
spec:
  destinationCIDRs:
  - 10.151.0.10/24
  destinationSID: 'fccc:cc00:a3:e901::'
  vrfID: 1
```

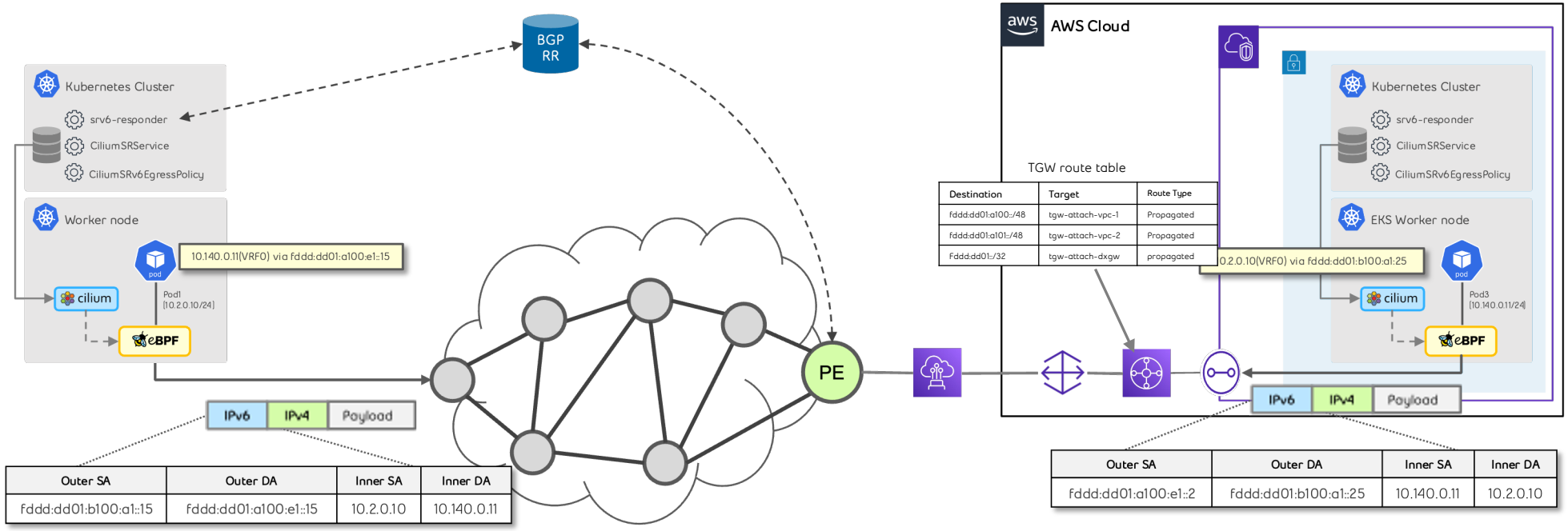
- Leveraging widely adopted Cilium project in use by all major Kubernetes releases
- Collaboration effort with Isovalent to augment Cilium CNI to support SRv6 encapsulation.
- Currently in **private-preview for release 1.13** of **Cilium Enterprise Edition**.
- Successful interop testing done with IOS-XR, FRR, GoBGP, JUNOS and ArcOS
- Aligned with multi-network KEP in sig-network https://github.com/kubernetes/enhancements/pull/3700



- POD has a single interface with only a default route.
- POD is associated with one or more VRFs through annotations
- CRD defines VRF attributes (RT, name, etc.) and BGP configuration parameters, etc.)
- SRv6 encapsulation done in eBFP through a CiliumSRv6EgressPolicy which can be statically configured or auto-generated through BGP NLRI data.

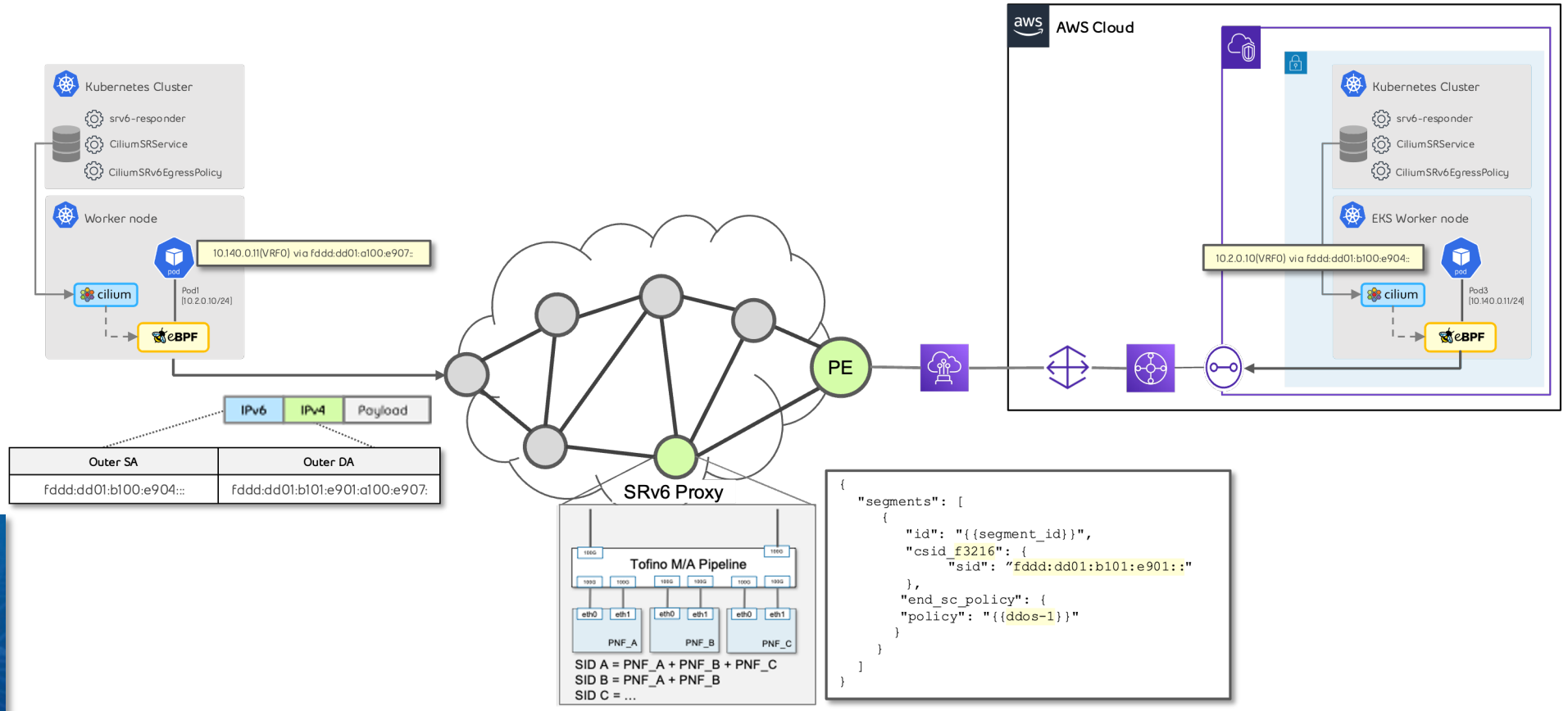
Extending SRv6 from Private to Public Cloud

- Work in progress to extend SRv6 over hyperscaler cloud infrastructure.
- Only possible with H.Encaps.Red or with SRH with f128 encoding due to ULA addressing limitations with hyperscalers.
- Collaborating extensively with AWS to enable Cilium Enterprise Edition with EKS Kubernetes release.
- EC2 or Google Compute Engine instances can either use eBPF for standalone workloads or leverage the right kernel implementations (kernel 6.1 for CSID support).
- Does not preclude deploying virtual routing appliances for non cloud-native workloads.



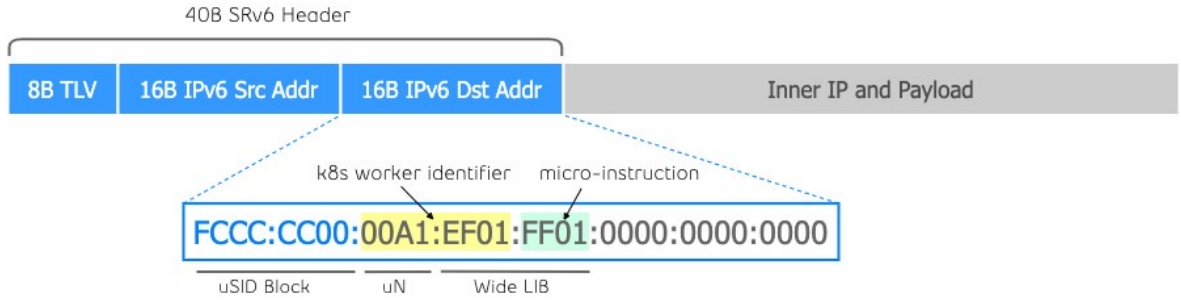
Inserting inline value-add network services

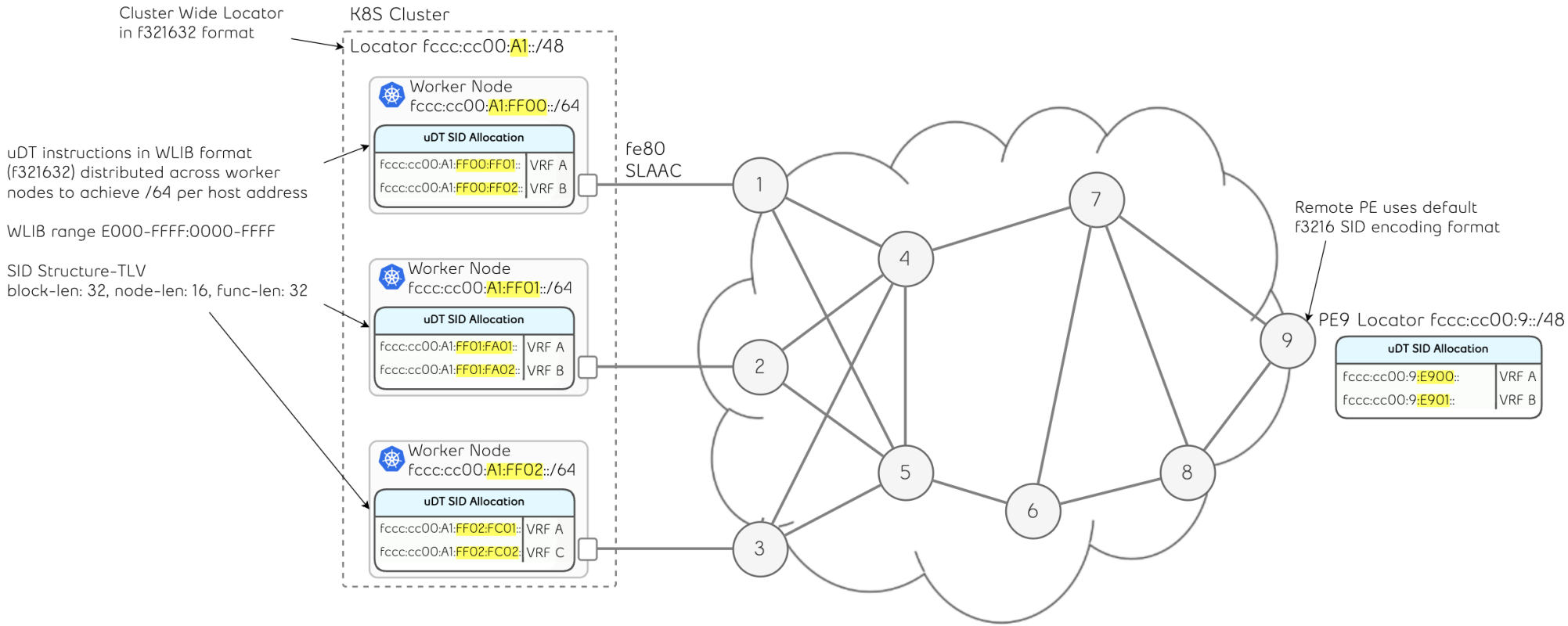
- Coming full circle with the work presented last year, leveraging in-network SRv6 proxy function.
- Can easily insert logically inline services through SRv6 service programming policies.
- Services can be inserted at any location in the network and steered using SRTE explicit paths.
- Paths can be configured via standard NETCONF or API driven through CRDs, P4RT(gRPC) or PCE.



SRv6 WLIB applied to Kubernetes

- How can we map uSID addressing schemes with K8S
- End goal is to align uSID to standard IPv6 host designs
- For this we leverage the Wide LIB capabilities
- Aligns really well with Cilium “cluster-pool” concept





Why Cilium ? – A Trial and Error Approach

Rethinking kubernetes networking with SRv6 and Contiv-VPP

Ahmed Abdelsalam, Cisco Systems; Daniel Bernier, Bell Canada; Rastislav Szabo, Filip Gschwandtner, Pantheon.tech; Miroslaw Walukiewicz, Intel

Extending Kubernetes Networking to make use of Segment Routing over IPv6 (SRv6)

Francesco Lombardo, Stefano Salsano, Ahmed Abdelsalam, Daniel Bernier, Clarence Filsfils

AWS picks Cilium for Networking & Security on EKS Anywhere

Sep 09, 2021 Isovalent

Today, we're introducing GKE Dataplane V2, an opinionated dataplane that harnesses the power of eBPF and Cilium, an open source project that makes the Linux kernel Kubernetes-aware using eBPF. Now in beta, we're also using Dataplane V2 to bring Kubernetes Network Policy logging to Google Kubernetes Engine (GKE).

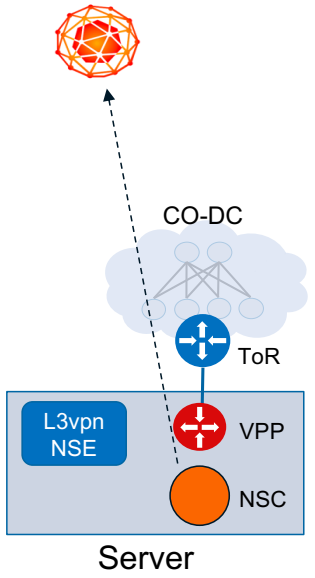
Public preview: Azure CNI Powered by Cilium

Published date: October 26, 2022

Azure CNI powered by Cilium provides native support for the next generation Cilium eBPF dataplane in AKS clusters running Azure CNI. It offers Pod networking, basic Kubernetes Network Policies, and high-performance service load balancing. The eBPF dataplane is available in both VNet mode and Overlay mode of Azure CNI.

Learn more.

Azure Kubernetes Service (AKS) Features

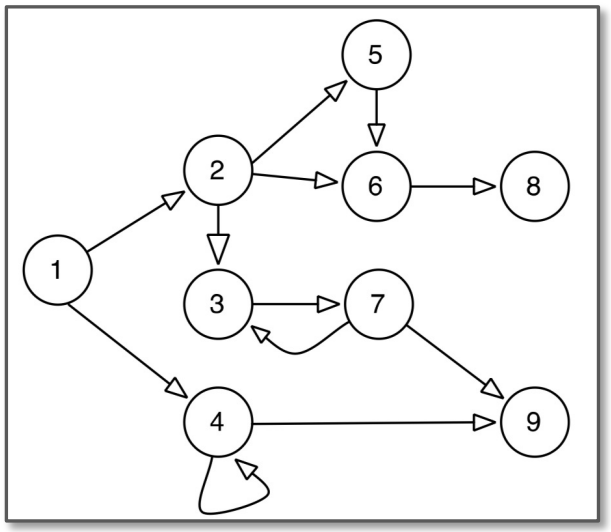


And Yes … I did try to make it work with NSM in 2019

```
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: l3vpnnse-client
  template:
    metadata:
      labels:
        app: l3vpnnse-client
    spec:
      containers:
        - image: 192.168.80.240:4000/tools/centos-tool-box:latest
          name: debug
          securityContext:
            privileged: true
          imagePullPolicy: Always
          command:
            - /bin/bash
            - -xec
metadata:
  name: l3vpnnse-client
  namespace: default
  annotations:
    ns.networkservicemesh.io:
l3vpnnse/vrf_red?vpn=5070:65033,l3vpnnse/vrf_yellow?vpn=5070:65044,
l3vpnnse/vrf_green?vpn=5070:65024
```

Another Telco Challenge – “Simple” service insertion

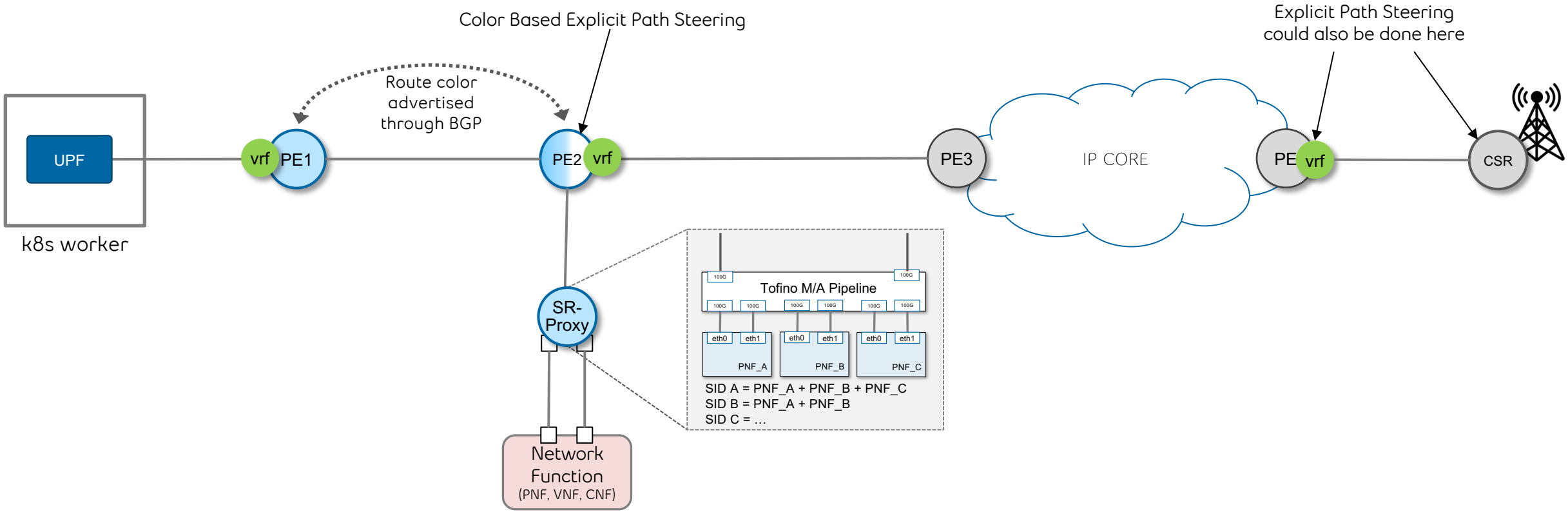
- Real Deployable Network Service Composition is Hard !
- Network Plumbing Complexity (if it works, don’t break it)
  - Traffic Selection Criteria
  - Topological Dependencies
  - Transport Dependence
  - MacGyver“esque” networking tricks (ToS manipulation, PBR, route leaks, EVPN sprawl, etc.)
- For this we leverage the Wide LIB capabilities
- Service Deployment Complexity
  - Guarantee of Service Ordering → Symmetric Traffic Flows
  - Service Scalability → Individual+Combined Services, Service Health Meatering
  - Multi-vendor Services → Dependency on service vendors



And “We Don’t Talk About … SD-WAN”

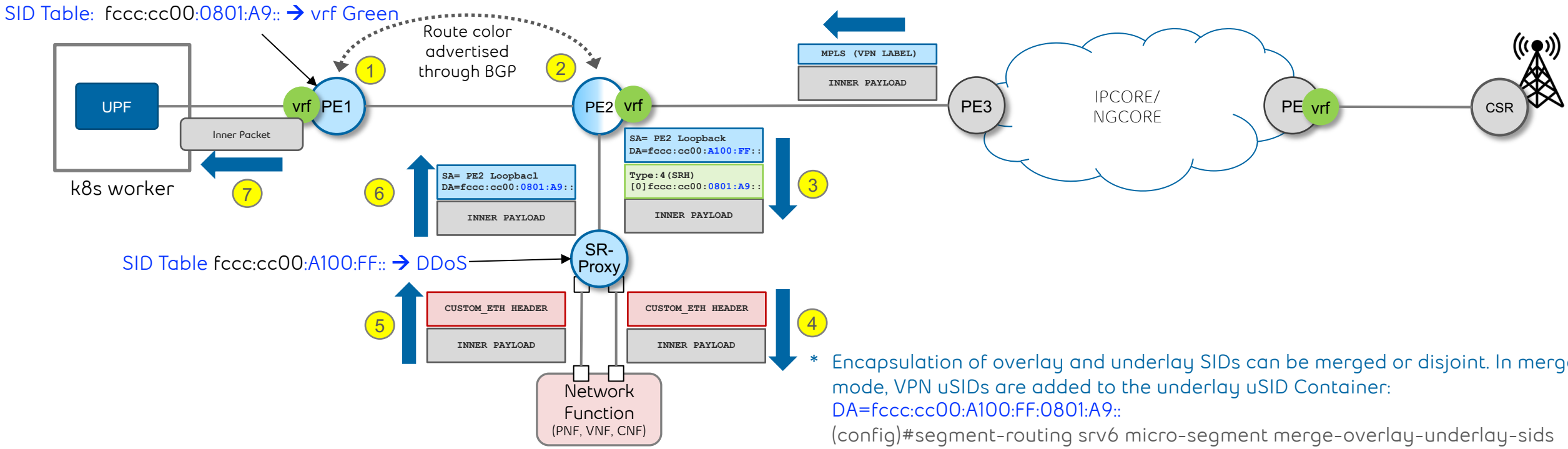
Stateless Dynamic Inline Service Insertion (1)

- Leverage SRv6 stateless traffic steering to dynamically insert services
- SR-Proxy function (Cybermapper) attaches *Network Functions* to traffic path
- SR-Proxy can create complex ”chains” agnostic to underlay network
- Can be used with on-prem CSP based workloads
- Release 1 supports route coloring, based path steering



Stateless Dynamic Inline Service Insertion (2)

1. MEC ToR colors targeted prefixes and advertise through BGP
2. MEC Leaf creates explicit-path SRTE policy based on color
3. Ingress traffic on MEC Leaf gets encapsulated* and forwarded to SR-Proxy
4. SR-Proxy performs END.SC operation and creates a custom ethernet header to send to network function
5. Network function returns packet with custom header
6. SR-Proxy performs PSP operation https://www.rfc-editor.org/rfc/rfc8986.html#section-4.16.1 and replaces outer header DA with next instructions
7. Egress ToR performs USD operation https://www.rfc-editor.org/rfc/rfc8986.html#section-4.16.2 (decap and lookup of inner packet)

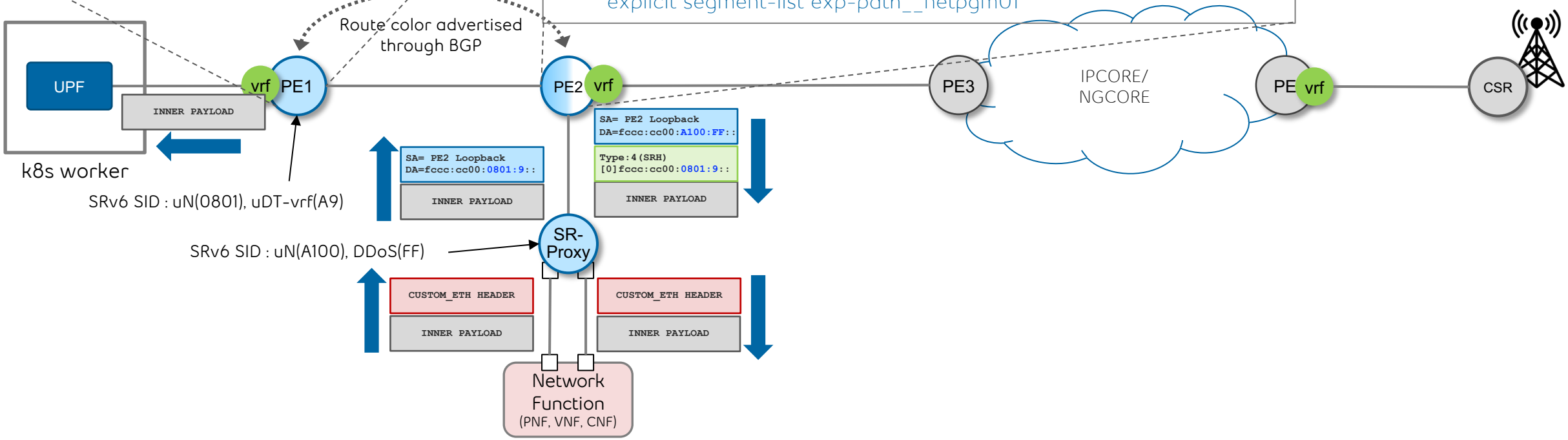


* Encapsulation of overlay and underlay SIDs can be merged or disjoint. In merge mode, VPN uSIDs are added to the underlay uSID Container:
DA=fccc:cc00:A100:FF:0801:A9::
(config)#segment-routing srv6 micro-segment merge-overlay-underlay-sids

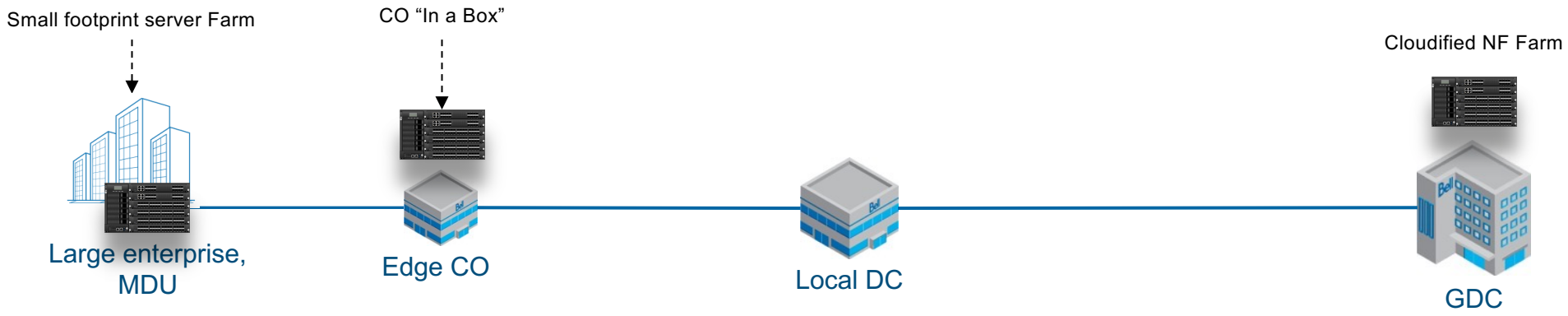
Stateless Dynamic Inline Service Insertion (3)

```
extcommunity-set opaque color_10
 10
end-set
!
route-policy set_color
  if destination in (UPF_Prefix) then
    set extcommunity color color_10
  endif
end-policy
!
router bgp 577
  neighbor <MEC_LEAF>
  route-policy set_color out
```

```
segment-routing
  traffic-eng
   srv6
    locator Locator0 binding-sid dynamic behavior ub6-insert-reduced
   !
   segment-list exp-path__netpgm01
    index 10 srv6 usid fccc:cc00:A100:FF::/64
   !
   policy exp-path-pol1
    srv6
    !
    color 10 end-point ipv6 <MEC_TOR>
    candidate-paths
     preference 100
      explicit segment-list exp-path__netpgm01
```

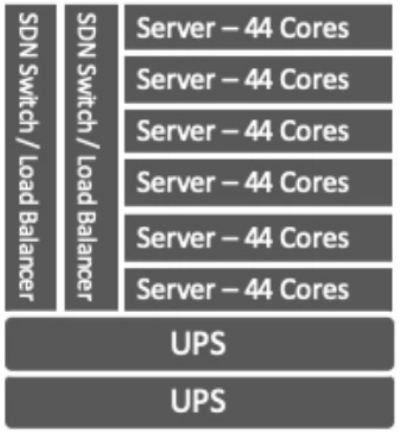


The Idea behind Lanner htca-6600



The Hypothesis

- Reduced facility requirements
- Reduced overall stack cost
- CO “in a Box”
  - Integrated networking
  - Integrated compute
  - Integrated “packet broker”



First Observations

In Conclusion

- Host based end-to-end SRv6 is currently possible in DC/Private cloud deployments
- With lots of new opportunities and innovations around the corner
- **Optimal efficiency in any cloud comes with uSID**
  - Data plane efficiency (HW Proxy, even eBPF complexity)
  - Massive scale and simplified operations
  - Overall simplicity
- However, some work is required with hyperscaler infrastructure for to achieve a truly unified uSID forwarding plane.

- Support for IPv4 protocol in next-header after IPv6 encap
- Ability to filter IPv4 prefixes out of VPC peering policies.
- Support for customer defined ULA addressing space (ie not forced fd20::/20)
- Support for BYOIPv6 address space including ULA

- Support for ULA addressing space (roadmap)
- Support for BYOIPv6 ULA address spaces (roadmap)
- Support for flexible CIDR ranges for IPv6
- Support of hashing based on IPv6 flow labels.

Thank You