vCMTS as a Service: Scalable and Extensible APIs

A technical paper prepared for presentation at SCTE TechExpo24

De Fu Li
Distinguished Engineer
Comcast
Defu_Li@comcast.com

Jay Zhu
Senior Principal Software Engineer
Comcast
Jay_Zhu@comcast.com

Berkay Aygun
Software Engineer
Comcast
Berkay_Aygun@comcast.com

Daniel Rice
Vice President II
Comcast
Daniel_Rice4@comcast.com

Mody Niv
Senior Vice President
Comcast
Mody_Niv@comcast.com

Suketu Bhatt
Senior Director
Comcast
Suketu_Bhatt@cable.comcast.com

Andrey Skvirsky
Executive Director
Comcast
Andrey_Skvirsky@comcast.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

List of Tables

Title **Page Number**

1. Introduction

1.1. PacketCable Multimedia

The PacketCable (PC) MultiMedia Framework (PCMM) is a specification [1] published by CableLabs. It provides Quality of Service (QoS) for the Internet protocol (IP) multimedia communications over the data over cable service interface specifications (DOCSIS®) 1.1 specification and later versions. The first version of the PacketCable Multimedia specification was published on June 27, 2003, targeting the voice over Internet protocol (VoIP) telephony application using embedded media terminal adapter (eMTA). And it has evolved to its latest, seventh revision on November 11, 2015, and has expanded its scope to provide a service agnostic QoS and accounting framework.

1.2. Comcast IP Telephony Solution

The PC/PCMM specifications are broad in scope [1][2]. Here, we focus on a specific use case of providing VoIP telephony services, which includes both residential and multi-line commercial services. To ensure high-quality voice and video calls for both residential and commercial subscribers, Comcast developed the "PacketCable MultiMedia - Next Generation" (PCMM-NG) solution. The PCMM-NG solution bridges the QoS requirements of both video and audio calls between the proxy - call session control functions (P-CSCFs) and the cable modem termination systems (CMTSs), ensuring that the necessary bandwidth and timing requirements are implemented for all the audio, video, upstream, and downstream components of each call. Because of this bridging functionality, PCMM-NG is also referred to as the Policy Bridge solution.

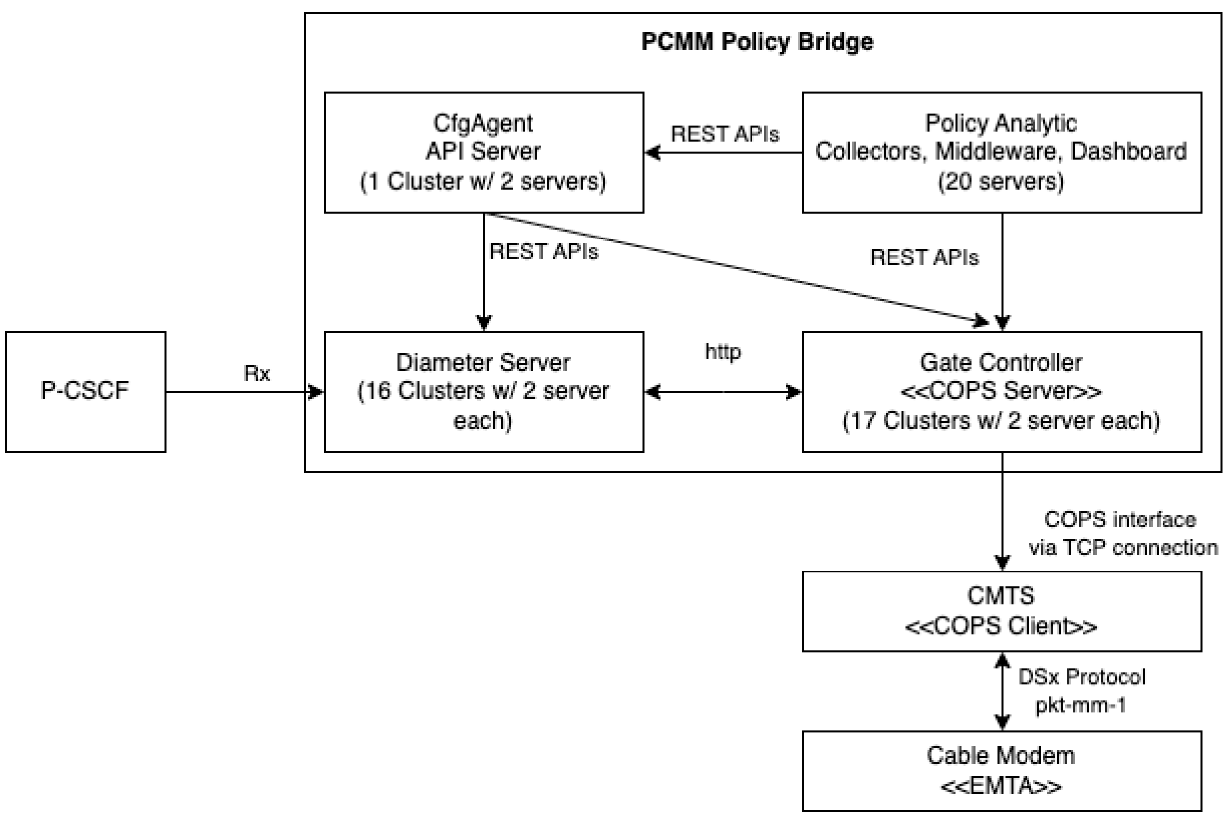

Figure 1 – Logical View of the PCMM Policy Bridge Solution

In Figure 1 – Logical View of the PCMM Policy Bridge Solution, the key PCMM interfaces are Rx and common open policy service (COPS). The Rx interface is used for session-based policy set-up information exchange between the P-CSCF, and the gate controller (GC) mediated by the Diameter servers. The Diameter servers are one of the key components of the PCMM-NG as the Rx interface uses the Diameter protocol and the Diameter servers determine the QoS settings of a call by analyzing the attribute-value pairs (AVPs) of the AA-Request (AAR) message and inform the GC of the required QoS settings over the hypertext transfer protocol (HTTP) interface. The GC then informs the CMTS serving the subscriber via the COPS connection, as specified by the Internet engineering task force (IETF). The Diameter servers also provide facilities to the PCMM-NG to communicate with the commercial and residential P-CSCFs of the Comcast network through the Diameter Rx interface, as defined by the 3rd generation partnership project (3GPP) for IP multimedia subsystem (IMS) networks.

Within the complex, the gate controller as defined by the PCMM specification is a COPS policy decision point (PDP) entity; it is also commonly referred to as the COPS server. The CMTS is the COPS policy enforcement point (PEP) entity. The dynamic QoS service flow is managed via the dynamic service add (DSA), change (DSC), and delete (DSD) DOCSIS media access control (MAC) management messages (MMMs) as defined in the MAC and upper layer protocols interface (MULPI) specification [3].

1.3. vCMTS Adaptation and Challenges

The legacy CMTS (L-CMTS) are CMTS products that were built upon custom hardware by equipment vendors. In contrast, it is well-known in the industry that the virtualized CMTS (vCMTS) refers to various software application workloads utilizing M-CMTS core functions according to the remote physical layer (R-PHY) and distribute access architecture (DAA) specifications. The vCMTS workloads can be deployed as containerized network functions (CNFs) on a cloud native platform or as virtualized network functions (VNFs) on virtual machines (VMs).

The need of vCMTS PCMM adaptation is driven by our rapid upgrades from L-CMTS to vCMTS over the years. During this transition period, the PCMM support for both L-CMTS and vCMTS platforms is required. A simple solution to this is to ensure that the vCMTS supports the COPS interface such that from the gate controller's point of view, vCMTS is compatible as an L-CMTS. However, the primary challenge, which came after implementing the COPS support, was about sizing and scaling, as tunning the partitioning strategy while balancing the system resource consumption exposed limitations in the software which we improved afterwards.

Part of the challenge is related to how the vCMTS's computing function is distributed and partitioned. Under the hood, the vCMTS decomposes its functionalities into micro-services, providing unmatched flexibility, reliability, and scalability in operations. Each micro-service is a containerized application and can be replicated for scaling. An instance of such containerized application is referred to as a "pod" in the compute cluster managed by Kubernetes (K8s). And the PacketCable client is one of the micro-services and is responsible for COPS PEP as well as interfacing the gate controller and multiple vCMTS control and data plane (CD) pods.

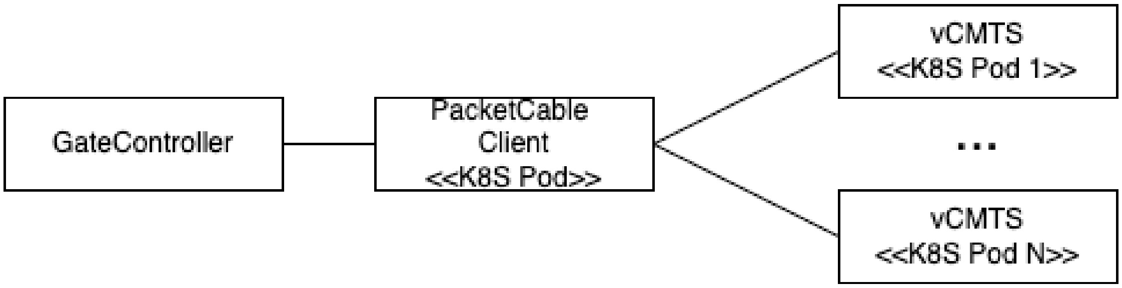

Figure 2 – A PacketCable Client Pod Serving Multiple vCMTS Pods

Several variables should be considered when choosing the right number of vCMTS CD pods to be served by each PacketCable client pod:

- Multiple vCMTS CD pods are managed by one vCMTS management (vCMTS-M) cluster.
- Multiple vCMTS-M clusters are hosted by one rack of servers.
- Multiple interconnected server racks share the same converged interconnect network (CIN).

Given these variables, tradeoffs are found between the resource allocation and the blast radius. The larger the scale, the higher the impact when there are issues. Meanwhile, with a larger scale, the resources such as the IP addresses and K8s pods, can be shared more efficiently. In a summary, we have designed and implemented a solution to support PCMM in the vCMTS, but there are observable, reasonable, and achievable improvements to be done in the future.

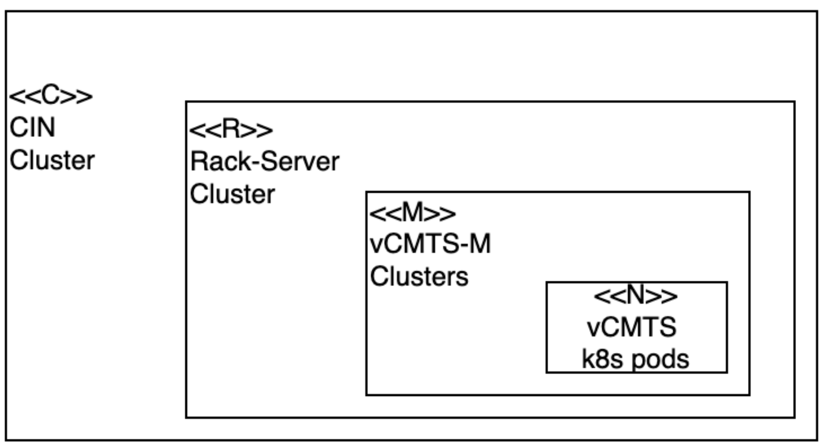

Figure 3 – vCMTS PacketCable Sizing Options

1.4. Network as a Service

At Comcast, we have an overarching goal of simplifying our end-to-end system in which reducing the operational complexity and engineering cost for our cloud native platform is significant. As we continue to expand our vCMTS deployment and learn as we progress, finding a simpler alternative to PCMM has emerged as a possibility. We aim to design and develop such an alternative solution to ensure smooth transitioning from PCMM and resolution to the pain points, and to establish a simpler foundation for onboarding new engineering talents who may not possess deep PCMM domain knowledge.

What could be an alternative? One option is to make the network functions and services available as application programming interfaces (APIs). This idea is not new, but it provides enabling technologies to help explore new monetization opportunities which ultimately fuel demand and growth – in a way intended by the PC/PCMM specifications by design.

As we envisioned a new solution, we also recognize that there is renewed interest in similar initiatives categorized as "Network as a Service (NaaS)" in the cable, telco, and mobile industries. Realizing this shared interest and understanding the modern software architecture/development patterns and the capabilities provided by the cloud motivated us to design a NaaS solution as an alternative to PCMM.

In the following sections, we introduce the "vCMTS as a service" solution as a feature-rich PCMM alternative. The vCMTS network service APIs are lightweight, stateless, secure, scalable, extensible, and are designed for eventual consistency, which significantly reduces the complexities when comparing to the traditional stateful protocols.

2. vCMTS as a Service APIs

In this section, the use cases and benefits are discussed to support the assessment of the proposed solution, and the high-level considerations of the "vCMTS as a service" APIs are discussed to provide a holistic view of the implementation references and requirements.

2.1. Use Cases

2.1.1. PCMM Alternative

A primary use case of the "vCMTS as a service" APIs is becoming a PCMM alternative. There are several improvements to be realized. First, building upon modern software techniques and patterns, the light-weight APIs provide stateless, secure, scalable, extensible, and consistent transactions compared to the traditional stateful protocols. This helps us simplify the vCMTS and the end-to-end system and improve their reliability. Furthermore, the removal of the gate controller and PCMM COPS interface support, the stateful message transactions, along with few other components reduce the overall system load and allow for elastic scalability in the new vCMTS QoS API service, which resolves the scaling challenges observed earlier. Finally, the APIs largely simplify the concept of dynamic QoS compared to PCMM and provide self-documenting abstractions for new engineers to onboard and contribute without deep PCMM domain knowledge.

These benefits alone justify the effort of developing and migrating from the PCMM to "vCMTS as a service" APIs especially when becoming a PCMM alternative is an achievable short-term incremental feature of the "vCMTS as a service" APIs.

2.1.2. Speed Boost

Another use case can be called the "Speed Boost". The Speed Boost is an application to allow the customers to request a temporary adjustment in their provisioned speeds through dynamic provisioning and QoS changes. This can be offered as an on-demand product where the customers pay for short-term speed boosts, or even as an advertisement offering where the customers can try and experience different speed tier upgrades.

Although it is theoretically possible to support Speed Boost using the PCMM, the required consistency and reliability can be challenging for the PCMM in practice. In comparison, the "vCMTS as a service" APIs allow such on-demand changes to be made atomically and consistently to avoid race conditions in resource acquisition and stale, invalid states.

2.1.3. Service Mobility

Instead of tying a service tier for a customer to a fixed device at a static location, with the flexibility and consistency of the "vCMTS as a service" APIs, customers' broadband offerings can be provided to any device with simple automatic device identification or account verification. We call this "Service Mobility". For instance, a mobile customer can have the same service tier provided anywhere no matter what network the mobile device is on. This offers seamless user experience whether the device is on the owner's home Wi-Fi, on a friend's or a neighbor's home Wi-Fi, or on a hotspot. The same applies to the home Internet services where customers can choose to move their bundles to any location and any device with just a few clicks on their smart phones.

2.1.4. Low Latency DOCSIS

In today's network, low latency is critical for delivering smooth broadband experience to the customers. The configuration of the low latency DOCSIS (LLD) today is designed to be statically applied to all customers. However, in such case, the customers do not have control over the low latency parameters such as its enabled/disabled states, and the active queue management (AQM) algorithm parameters for both low latency queue and the classic queue. If the customers can have a pre-defined set of low latency configuration options to choose from based on their immediate usage needs, the LLD experience is further enhanced and there is a potential boost to customer satisfaction which is important for customer churn reduction. This is a use case for which the vCMTS APIs can be an enabling technology.

2.1.5. Other On Demand Services

The "vCMTS as a service" APIs also enable the offering of various other on-demand services such as:

- On-demand virtual private network (VPN)
    - Remote healthcare visits.
    - Secure business VPN for travelers.
    - Seamless integration with software-defined wide area network (SD-WAN) service and providing the service level objective (SLO) metrics for network QoS insights.
- On-demand broadband service
    - Any customer can choose to purchase temporary broadband service passes at any time for arbitrary durations as microtransactions. This includes customers that are traveling, moving, or temporarily staying.

These offerings add diversity and flexibility into our broadband products to boost customer experience based on their needs. They also provide monetization models to the operators to expand their business models and find opportunities for growth.

2.2. vCMTS APIs

In this section, we discuss the approaches to implementing the vCMTS APIs for NaaS, the architecture design, the system components, and the service APIs and object models.

2.2.1. Approaches and Considerations

2.2.1.1. The CAMARA Project

When considering the approach to implementing the vCMTS APIs, the CAMARA project [4] demonstrates shared interests in defining and developing NaaS APIs among the service providers. The CAMARA is an open-source project launched by the Linux Foundation while collaborating with the

GSMA with a mission of fostering the definition, development, and validation of user NaaS APIs. The GSMA Open Gateway initiative defined the system architecture and highlighted the CAMARA Service APIs northbound to the communication service provider (CSP) domain [5], as shown in Figure 4 – Open Gateway NaaS Architecture and Contributing Stakeholders.

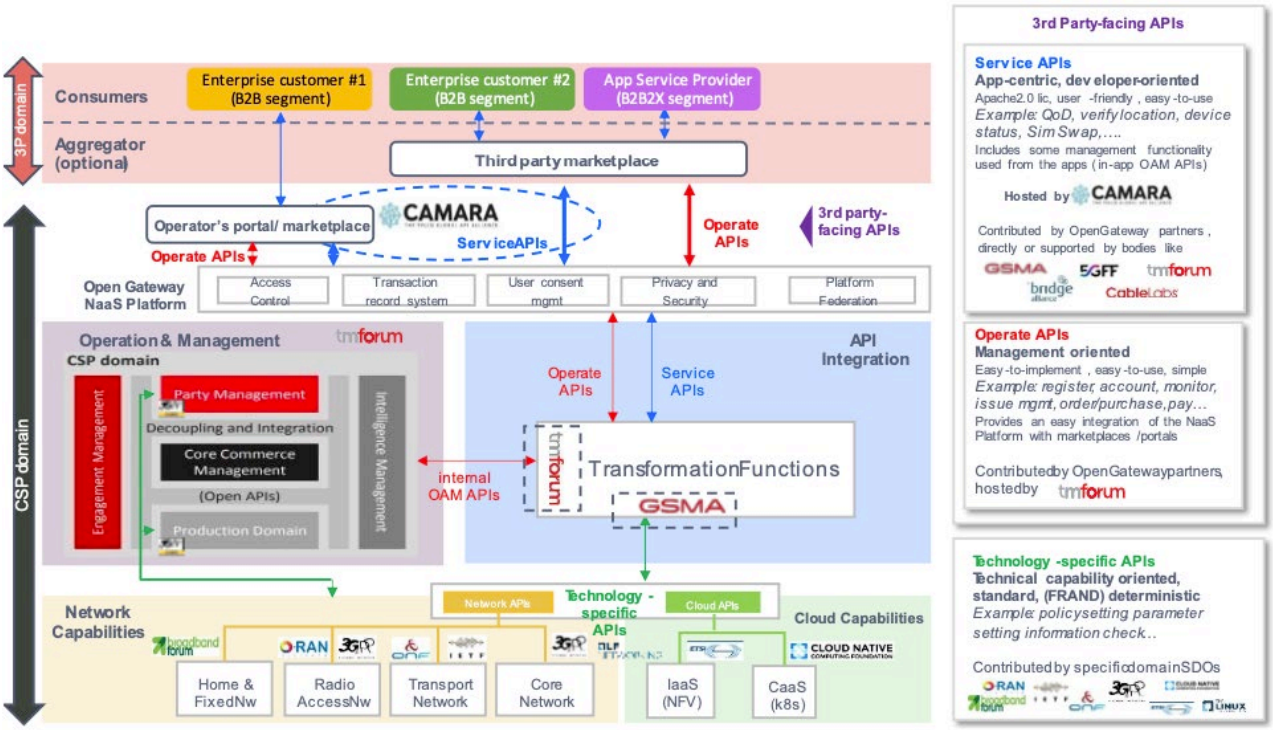

Figure 4 – Open Gateway NaaS Architecture and Contributing Stakeholders

The CAMARA APIs are exposed to the customers directly or indirectly through aggregators. The APIs are categorized into two groups: Service APIs, and Service Management APIs. The Service APIs provide purpose-specific capabilities, such as quality on demand (QoD), device location, edge discovery and selection, etc. The Service Management APIs are service request APIs enabling applications to order the enablement of a certain functionality.

2.2.1.2. Phases

The GSMA Open Gateway architecture framework provides a comprehensive view of the application layers and domains. As a CSP, we envision a bottom-up approach which starts from the CSP's network capabilities layer to prioritize technology-specific APIs. This initial phase is designed for building a PCMM alternative as discussed in the previous sections. The prioritization of the first phase is to simplify and reduce the total cost of ownership (TCO). And its subsequent phase will focus on NaaS use cases where microtransaction based, pay-as-use services, and on-demand service enablement are directly provided to the customers through APIs used by software applications.

2.2.2. Architecture

When considering the architecture design and migration strategy, backward compatibility is a high-level requirement from the operational point of view to support rolling upgrades or downgrades within the compatible versions. Similarly, the system must support both operation paths during its transition period to maintain reliability and consistency.

With the edge compute platform that we are building as a converged, virtualized broadband platform, in addition to the vCMTS, the virtualized broadband network gateway (vBNG) for the ethernet passive optical network (EPON) is supported. And because all previously discussed use cases apply to the EPON, the system architecture must be extensible to vBNG or any access technology.

The system components and their high-level relationships are depicted in Figure 5 – IP Telephony Service Architecture Utilizing the QoS APIs.

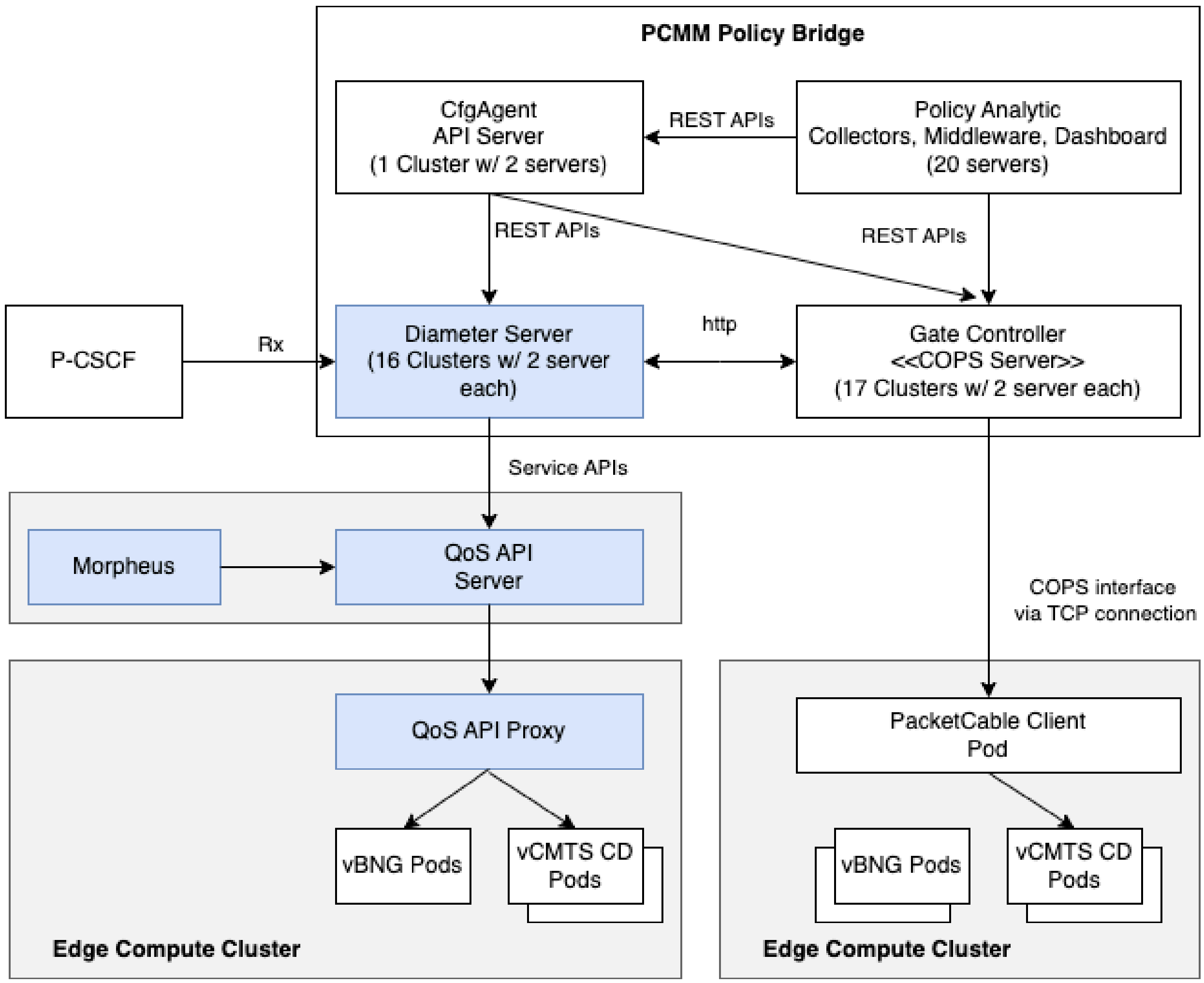

Figure 5 – IP Telephony Service Architecture Utilizing the QoS APIs

2.2.3. System Components

The modified components and the newly introduced components can be summarized as follows:

- Diameter Server

  The Diameter server component handles the AAR or the Re-Auth-Request (RAR) for creating and updating the dynamic QoS service flows. The Diameter server provides an adaptation layer to translate and route the request to an existing gate controller component or the QoS API server.

- Morpheus

  The Morpheus component is our existing deployment and change management pipeline which includes functions such as site standup, release upgrades and downgrades, etc.

- QoS API Server

  The QoS API server component provides an interface for Morpheus to update IP scope associated with an edge compute cluster FDQN instance. Morpheus also updates any IP scope changes to the QoS API server. The IP scope and the fully qualified domain name (FQDN) associations are persistently stored in the QoS API server's database which allows the QoS server to route the API requests via endpoint IP to the FQDN lookups to precisely target the correct edge compute cluster.

- QoS API Proxy

  The K8s pod's life cycle can be ephemeral when the pod is recreated or destroyed due to a failure, or its resource requirement changed. Similarly, allocating a workload servicing an RPD with a K8s pod is ephemeral. Hence, the endpoint IP of a k8s pod can be dynamic. As a solution, the QoS API proxy maintains real-time mappings and handles the final stage of the API routing within the compute cluster.

2.2.4. Service APIs and Object Models

An edge compute cluster is identified by its FQDN where an IP scope is allocated to. This is published by the Morpheus when a compute cluster is instantiated or when the assigned IP scopes changed. With that, the QoS API server maintains its own database of the edge compute cluster inventory and the associated IP scopes for the entire production network.

The service information and data model are primarily based on the PCMM specification, namely the GateSpec, FlowSpecs, and DOCSIS QoS specific parameters. We envisioned that the QoS API server will potentially be deployed in the public cloud to establish a global presence. At the QoS API server layer, the service enablement object is access technology agnostic and should be modeled after the resource reservation protocol (RSVP) TSpec and Rspec. The access technology specific mappings and QoS parameter translations are handled by the QoS Proxy.

Table 1 – QoS Attributes Map

RSVP TSpec	RSVP RSpec	DOCSIS DS QoS	DOCSIS US QoS
Bucket depth (bytes)		DOCSIS maximum traffic burst	TSpec bucket depth, TSpec maximum datagram size, TSpec minimum policed unit, DOCSIS unsolicited grant size
Maximum datagram size (bytes)		N/A	
Minimum policed unit (bytes)		DOCSIS assumed minimum reserved rate packet size	
Bucket rate (bytes/second)	Reserved rate (bytes/second)	DOCSIS minimum reserved rate	TSpec bucket rate, TSpec peak rate,

Peak rate (bytes/second)		DOCSIS maximum sustained rate and downstream peak traffic rate (DOCSIS 3.0)	RSpec reserved rate (used to calculate nominal grant interval)
	Slack term (microseconds)	DOCSIS downstream latency	DOCSIS tolerated grant jitter

The QoS API server provides a representational state transfer (REST) API interface. For the interfaces between components or k8s pods within the edge compute cluster, Google remote procedure call (gRPC) is used, and the RPC methods consist of create, read, update, and delete (CRUD) operations.

The session object is the container of the network service objects. The QoS API server and QoS Proxy manages the session objects and their lifecycle states. Their interaction and management include aging timer and reconciliation and are based on the eventual consistency model. The session object has attributes shown listed in Table 2 – Session Object Attributes.

Table 2 – Session Object Attributes

Attribute	Description
Session UUID	Session UUID, equivalent to the PCMM gateId
subscriberUuid	Paid Subscriber UUID
appUuid	Application (Client) UUID, i.e. P-CSCF, Diameter Server
sessionState	State = {ACTIVE | INACTIVE}
sessionSpec	Service enablement specification, equivalent to the PCMM gateSpec
sessionSpec.endpointIp	Endpoint IP address, i.e. eMTA, equivalent to the PCMM gateSpec.subsId
sessionSpec.creationTs	Creation timestamp
sessionSpec.lastUpdateTs	Last update timestamp
sessionSpec.lifetimeDuration	Default = 1 day, -1 = infinite, 1 minute granularity
sessionSpec.dsFlowSpec	Downstream traffic classifiers and its QoS Profile
sessionSpec.usFlowSpec	Upstream traffic classifiers and its QoS Profile

At the high level, the API routing and calling flow are described in the sequence below:

- **Morpheus** publishes the edge compute FQDN IP scope via the endpoint api/fqdnIpScope.
- **Diameter Server** requests for service flow setup for the VoIP call via the endpoint api/Session.
- **QoS API Server** handles the request via lookup for the endpoint IP to target the edge compute FQDN and routes the request to the destination **QoS API Proxy**.
- **QoS API Proxy** translates the QoS parameters to access technology specific parameters and invokes the gRPC call to the targeted **K8s pod.**

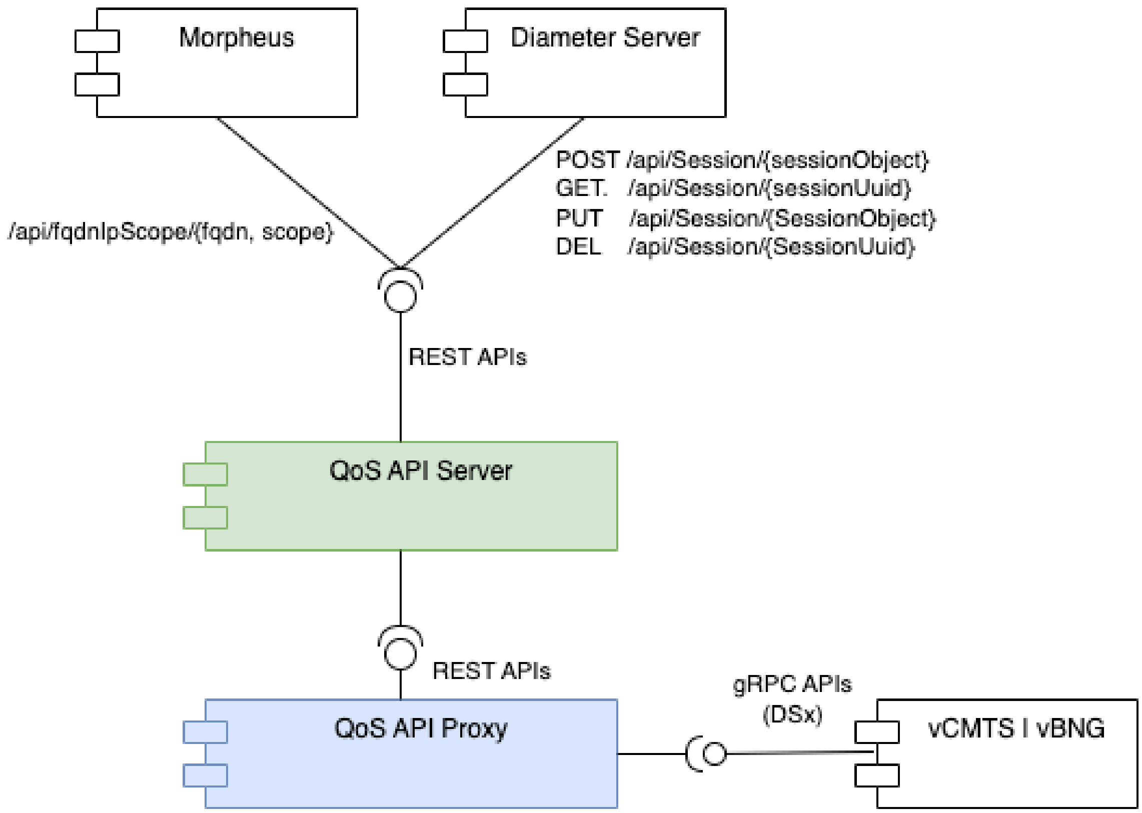

Figure 6 – Provided Service APIs

3. Conclusion

In this paper, we discussed the background, limitations and challenges of operating the PCMM at scale. We also discussed our proposed new solution at Comcast – the "vCMTS as a service" APIs which not only simplifies our end-to-end system, serves as a scalable, secure, extensible, and lightweight alternative to the PCMM, but also creates new growth opportunities for us to explore in dynamic QoS, service mobility, on-demand microtransaction use cases, and beyond.

With the CAMARA project as the North star and our edge compute platform as the foundation, the "vCMTS as a service" solution is one of our incremental products for bridging our most advanced network technologies with our customers' needs.

Abbreviations

3GPP	3rd generation partnership project
AAR	AA-Request
API	application programming interface
AVP	attribute-value pair
CD	control and data plane
CIN	converged interconnected network
CMTS	cable modem termination system
CNF	containerized network functions
COPS	common open policy service
CRUD	create, read, update, and delete
CSP	communication service provider
DAA	distributed access architecture
DOCSIS	data over cable service interface specifications
DS	downstream
DSA	dynamic service addition
DSC	dynamic service change
DSD	dynamic service deletion
eMTA	embedded media terminal adapter
EPON	ethernet passive optical network
FQDN	fully qualified domain name
GC	gate controller
gRPC	Google remote procedure call
HTTP	hypertext transfer protocol
I-CMTS	integrated cable modem termination system
IETF	Internet engineering task force
IMS	IP multimedia subsystem
IP	Internet protocol
K8s	Kubernetes
L-CMTS	legacy cable modem termination system
LF	Linux foundation
LLD	low latency DOCSIS
MAC	media access control
MMM	MAC management message
MULPI	MAC and upper layer protocols interface
NaaS	network as a service
PC	PacketCable
PCMM	PacketCable multimedia
PCMM-NG	PacketCable MultiMedia - Next Generation
P-CSCF	proxy - call session control function
PDP	policy decision point
PEP	policy enforcement point
QoD	quality of demand
QoS	quality of service

RAR	Re-Auth-Request
REST	representational state transfer
R-PHY	remote physical layer
RSVP	resource reservation protocol
SCTE	society of cable telecommunications engineers
SD-WAN	software-defined wide area network
SLO	service level objective
TCO	total cost of ownership
US	upstream
UUID	universally unique identifier
vBNG	virtualized broadband network gateway
vCMTS	virtual cable modem termination system
vCMTS-M	vCMTS management
VM	virtual machine
VNF	virtual network function
VoIP	voice over Internet protocol
VPN	virtual private network

Bibliography & References

1. *PacketCable Multimedia Specification,* PKT-SP-MM-I07-151111

2. *Multimedia Architecture Framework Technical Report,* PKT-TR-MM-ARCH-C01-191120.

3. *DOCSIS 4.0 MAC and Upper Layer Protocols Interface Specification,* CM-SP-MULPIv4.0-I08-231211.

4. *CAMARA Project: APIs enabling seamless access to Telco network capabilities,* Linux Foundation, https://camaraproject.org

5. *The Ecosystem for Open Gateway NaaS API Development*, GSMA, https://www.gsma.com/futurenetworks/gsma-open-gateway