Flexible MAC Architecture in the Cloud: Architectures for a virtual world

Douglas Johnson
Principal Software Architect
Vecima Networks, Inc.
Saskatoon, SK
douglas.johnson@vecima.com

Jeremy Thompson
Sr. Software Architect
Vecima Networks, Inc.
Saskatoon, SK
jeremy.thompson@vecima.com

Table of Contents

List of Figures

List of Tables

1. Introduction

Over the last few years, and accelerated by COVID-19, the approach to corporate IT has fundamentally changed and companies are undergoing significant shifts in IT strategy and culture. The illusion of a "private, secure" network run by undersized IT teams has been shattered and companies are left grappling with the complexity and security of large, remote, organically grown networks.

Cloud as-a-service operators offer a reprieve: augment your team with our offerings. Their teams manage the day-to-day complexity and security of the services in a cost-effective way while your IT teams focus more on value-added services specific to the business. In some domains, that value-add is as simple as on-prem tech support. In other domains, the value-add can be a significant network unto itself.

The latter domain is our focus: MSOs operate significantly complex networks and domain-specific applications. MSO teams have specialized technical skills and experiences which allow operators to provide scalable and robust Internet connectivity to their end-users. Cloud offerings can augment existing investments by reducing time to deploy new services and enabling existing teams to focus on domain specific problems and solutions.

In this paper we look at a Flexible MAC Architecture (FMA) deployment following these principles. Some of the network is domain specific and managed by specialized in-house teams which are then augmented by resources and teams provided by 3rd party Cloud offerings. We examine the viability of a hybrid approach to FMA deployment through design, constraints, security, and costs using a prototype deployment.

2. FMA Overview

CableLabs®'s Flexible MAC Architecture defines an architecture for deploying a Remote MACPHY architecture, where the DOCSIS processing is done remotely in specialized hardware and the management is disaggregated into software components. The architecture is comprised of 3 primary components:

- MAC Manager (MM). A Management plane component that aggregates many RMDs into a single, unified controller. It provides a backwards compatible OSSI interface to legacy cable backoffice technologies.
- Remote MACPHY Device (RMD aka MAC-NE). A physical device containing a DOCSIS MAC and DOCSIS PHY expected, but not required, to be housed within an outside plant Node enclosure.
- PacketCable Aggregator (PAG). An aggregation component which bridges between existing PacketCable infrastructure and a population of deployed RMDs.

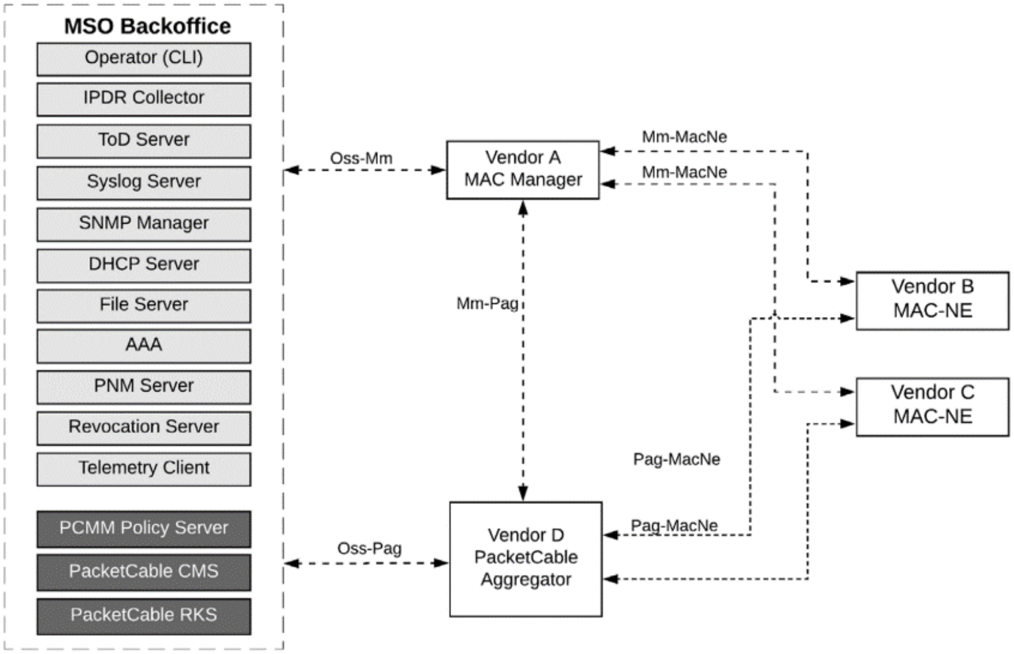

Figure 1 - CableLabs FMA

A key differentiation between FMA and Modular Headend Architecture v2 (MHAv2) Remote PHY Devices (RPDs) paired with (virtual) Cores is that the DOCSIS portion of the access network is terminated at the remote RMD and customer bearer traffic is readily available at the first-hop aggregation switches within an operators' network. FMA separates the data plane packet handling from the management components, placing data plane into the RMD and management plane concerns in the MAC Manager. This separation removes the need for the MAC Manager to handle high throughput packet processing and allows the MAC Manager to be more easily virtualized.

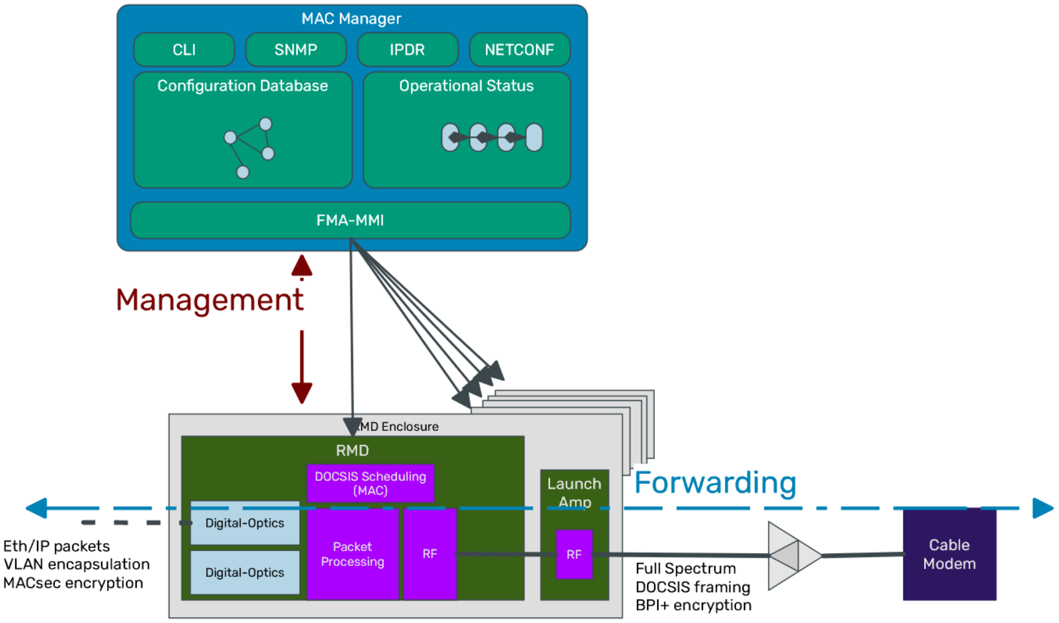

Figure 2 - FMA Management / Data Plane Separation

We take advantage of this property to build a best-of-breed hybrid network: domain specific networking concerns for bearer traffic are handled by in-house specialists while generic compute resources are augmented into the team by scaled cloud providers.

A cloud-based Flexible MAC Architecture can be readily designed in many ways, some of those options are explored within this paper as a thought-exercise. To explore a cloud-based solution more concretely, we deployed our solution for testing as follows:

- The MAC Manager was deployed into Amazon AWS, although any large cloud provider could be used.
- A VPN connection was established to a private lab network
- The MSO backoffice was on the private network
- The cable modem traffic was routed on the private network

This leads to a hybrid network where AWS was used as an extension or expansion of our private network. Customer traffic was not routed to or originated from an AWS address space.

3. Clouds

There are many options for cloud providers today, both large and small. The large cloud providers, such as Amazon, Google, and Microsoft, offer similar competitive portfolios and can be compelling partners when investigating cloud augmentation.

Cloud service offerings can be approached and purchased in different ways, and we categorize the offerings into the following from higher level to lower level. The service provided by the cloud operator dictates the required software packaging, service model, and abstraction level for any application deployed into that service.

- **Software-as-a-Service (SaaS)**: Building an application on top of provider-specific applications services. SaaS applications can be augmented to other deployment technologies to address specific application requirements. SaaS offerings have the largest variance between cloud providers.
- **Containers**: Building an application as a set of containers deployed onto a cloud-managed Kubernetes or another container orchestration platform.
- **Virtual Machines**: Building an application bundled with an operating system and targeting an ideal hardware environment which would run on a hypervisor and be deployed as a unified whole.
- **Bare Metal**: Building an application bundled with an operating system targeting a specific hardware environment and running directly on the hardware resources.
- **Racking**: Renting rack-space, lab resources, and connectivity while purchasing and managing hardware life cycles and depreciation yourself.

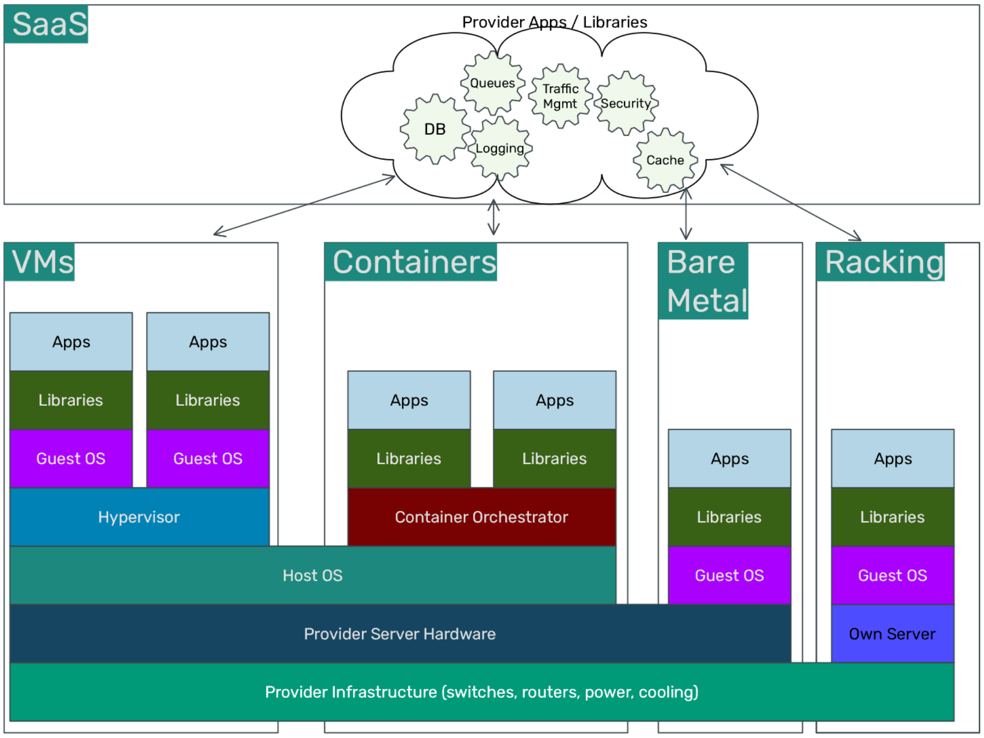

Figure 3 - Cloud offering types

The lowest level options, Bare Metal and Racking, are not as attractive to most operators because there is little value-added services added to an operator team; as such, they are not discussed in this paper.

The final 3 options of Virtual Machines, Containers, and SaaS each offer different advantages and disadvantages that need to be considered when investing in Cloud solution architecture.

4. Deployment Models

Virtual Machines, Containers, and SaaS augmentation are three ways to engage with cloud providers and each type of engagement has different strengths, weaknesses, and costs. In this section we provide an overview of these engagement models.

4.1. Virtual Machines

The most straightforward cloud-based deployment model is the placement of virtual machines (VMs) into a cloud provider network. A VM combines the application software with a bundled, often customized, operating system into a portable VM image which can be launched on a hypervisor.

Virtual machines are attractive, in part, due to their low coupling between the software application and the virtual infrastructure. This makes the VM easy to target as an application developer and easy to deploy into any cloud offering, reducing cloud vendor lock-in. Virtual machines can bridge gaps between defined hardware appliances and a fully virtualized world making it easy to work with internal teams and external vendors.

There are some inherent disadvantages to bundled virtual machine deployments. Given their agnostic attitude to their infrastructure and that they include their entire operating system, they can sometimes cost more than other options. In addition, all application dependencies are usually included directly in the virtual machine image, making it difficult to offload application features, such as database redundancy and resiliency, to the cloud operator.

A single MAC Manager is expected to manage many RMDs, so consideration must be given to redundancy, resiliency, and the impact of an outage (planned or unplanned) on customer services. In a similar way to the physical deployments that VMs emulate, high availability strategies come from the VM vendor implementation and are difficult to transparently offload to a cloud provider. When deploying redundant VMs, it’s important to ensure cloud availability zones (AZs) are a part of the strategy which limits or restricts the use of layer 2 based high availability techniques. Most cloud providers offer some insight and monitoring into the health of VMs but do not have visibility into the health of the application(s) running in the VM.

The VM resources are analyzed in a similar fashion to physical deployments - in increments of CPU, memory, and storage. Evaluation and costing of compute resources in a VM model is straight-forward as the costs of a virtual machine are obvious from the cloud provider. Network traffic needs are more sensitive to the running application design and configuration, as certain parameters such as telemetry, logging frequency, and communication density may be adjustable in the application. However, between the two cost considerations of compute and traffic, network traffic costs will outpace those of the compute resources except in certain edge-cases, such as GPU compute.

There is significant parity among all the major cloud provider offerings regarding virtual machine deployments and compute resource offerings.

4.2. Containers

Containers offer a deployment option in which individual microservices can be launched and managed as discrete entities running on a virtual environment. Containers can be deployed manually but are more often paired with an orchestration service, such as Kubernetes, which performs the role of container lifecycle management, redundancy, storage virtualization, and load balancing functionality.

Containers decouple the applications into individual services, allowing each to be deployed independently. Where a VM commonly bundles "everything" into a single integrated deliverable, a container deployment provides a model to offload key functions to an operator or third parties. By discarding the Guest OS, containers reduce the virtualization overhead required to deploy applications and may reduce overall costs of a solution.

In a VM model, a hypervisor limits each VM to a specific set of resources and ensures two or more VMs operating on the same physical hardware do not interfere with each other. Containers run without a hypervisor and without a Guest OS, so resource constraints need to be considered on a per-container basis. Most container models allow for fine-grained tuning of resources and resource limits to give priority resources to the applications that need them most while ensuring that none overstep defined limits causing negative side effects to the broader system. Shared resources such as disk, memory, and network access can be defined on an individual basis that are best suited to the requirements of the application.

Orchestration systems, like Kubernetes, enable larger shared resource pools across many physical devices to be managed and container instances to be deployed automatically within the pools. Using load balancing or distribution strategies, individual containers managed by the orchestrator can have workloads distributed evenly (in, for example, a round-robin fashion) or as a redundancy strategy. Containers can exist as long-term entities for persistent, permanent operation or short-term entities for distributed workloads such as metrics processing or batch operations.

Container orchestration systems can be deployed in VMs or onto bare metal by an operator manually, however, most major cloud providers offer a Kubernetes container deployment target as a service. In these models, the operator doesn't think about or provision any virtual machines and can deploy containers directly into the cloud operator's container network. The underlying virtualization/HW is left to the cloud operator.

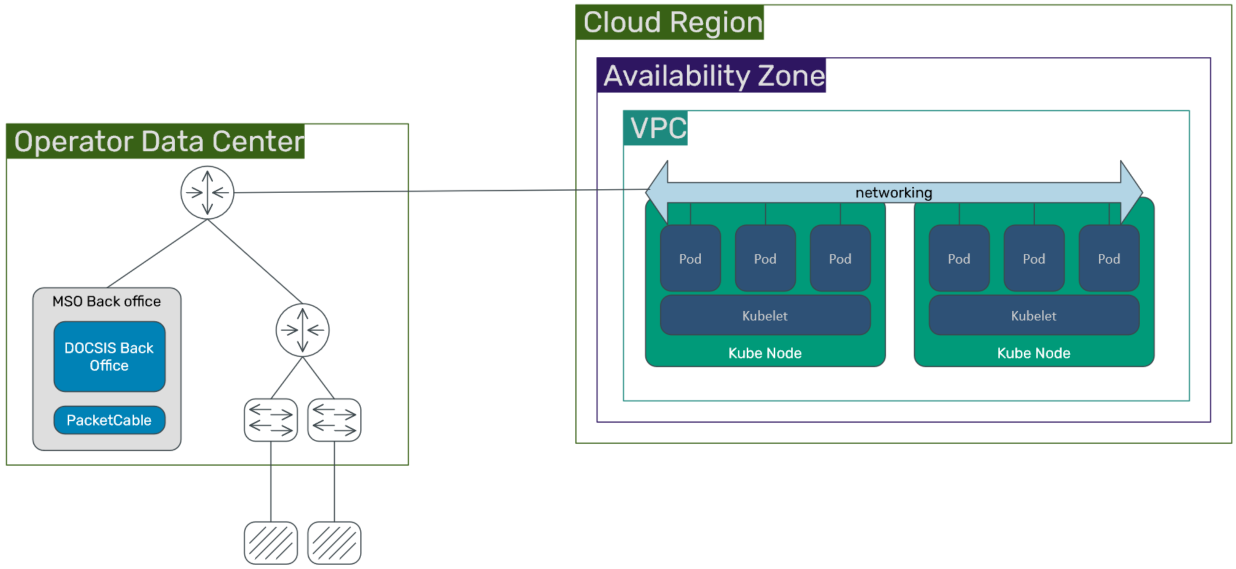

Figure 4 - Orchestrated Containers

A key principle of microservices and containers is the reduction of application-scope: A container will do one thing well and rely on other containers to provide any other services. For example, a container will often be stateless and rely on other containers or SaaS offerings to provide stateful storage, such as a database service. In some cases, the smaller container scope allows for best-of-breed application container choices and increases release velocity of individual container applications. The downside of this approach is the increase in the number of integration points an operator needs to manage. When a container needs to communicate with other containers to fulfill its responsibility, it does so with a protocol, protocol version, and specific API. These communication points need to be integration tested by the operator and vendor before confidence in the whole system can be established. These new integration tests can be empowering for an operator but also need to be understood and managed through life cycle and resource allocation.

4.3. Software-as-a-Service

Software as-a-Service (SaaS) can augment any other type of deployment, outsourcing critical generalized functionality to the cloud provider. SaaS functionality between cloud providers is the most specialized with different providers offering different SaaS products. In many cases it is also the most expensive but also the highest value functionality a cloud provider can offer. Databases, queues, traffic routers, and caches are all generic architectural components demanding high availability and reliability while also being complex to deploy and manage. SaaS offerings from cloud providers offload that complexity to their specialized teams to handle infrastructure, monitoring, security, and support infrastructure. This lowers the operator's burden and allows operator teams to focus on connectivity domain specific concerns.

Some SaaS offerings require little custom integration to gain the benefits. For example, most SaaS SQL databases are compatible with SQL client applications without any additional effort. However, in some cases, specialized high-value offerings, such as AWS Lambda, need specialized application logic and delivery mechanisms to integrate correctly.

While SaaS offerings can be cost-effective, reliable, and easy to use, they can cause an application to be locked into a specific cloud provider due to proprietary APIs or service offerings from a specific cloud provider.

Operators can compare SaaS offerings versus their own resources and areas of expertise and make individual evaluations for each of these categories of need. Ultimately, decisions will come down to costs and benefits. While maintenance of logging stacks, metrics stacks, or databases is possible independent of cloud providers, SaaS choices remain available, and these options are where distinctions between cloud providers are more evident. Furthermore, it becomes a matter of comfort, preference, or experience that may define more attractive choices to operators.

4.4. Cloud location & Connectivity

Operators will most often have, and want to take advantage of, the opportunity offered by major cloud service providers to geographically locate cloud-based deployments into areas that pair best with MSO datacenters. Latency tests indicate that even cross-continent, round trip times are low enough that RMD to MAC Manager communications are feasible without failures but reducing RMD to MAC Manager latency could improve the overall performance of interactive tasks.

A standard connection to a cloud provider would be over the public Internet with best-effort delivery. At a minimum, a VPN connection established between the cloud provider network and the operator network would be expected, but this is still delivered as best-effort over the Internet. The largest cloud operators offer additional services to reduce the latency of best-effort connectivity between an Operator core network and the cloud services. These additional services vary between the cloud operators, but we attempt to unify the concepts here.

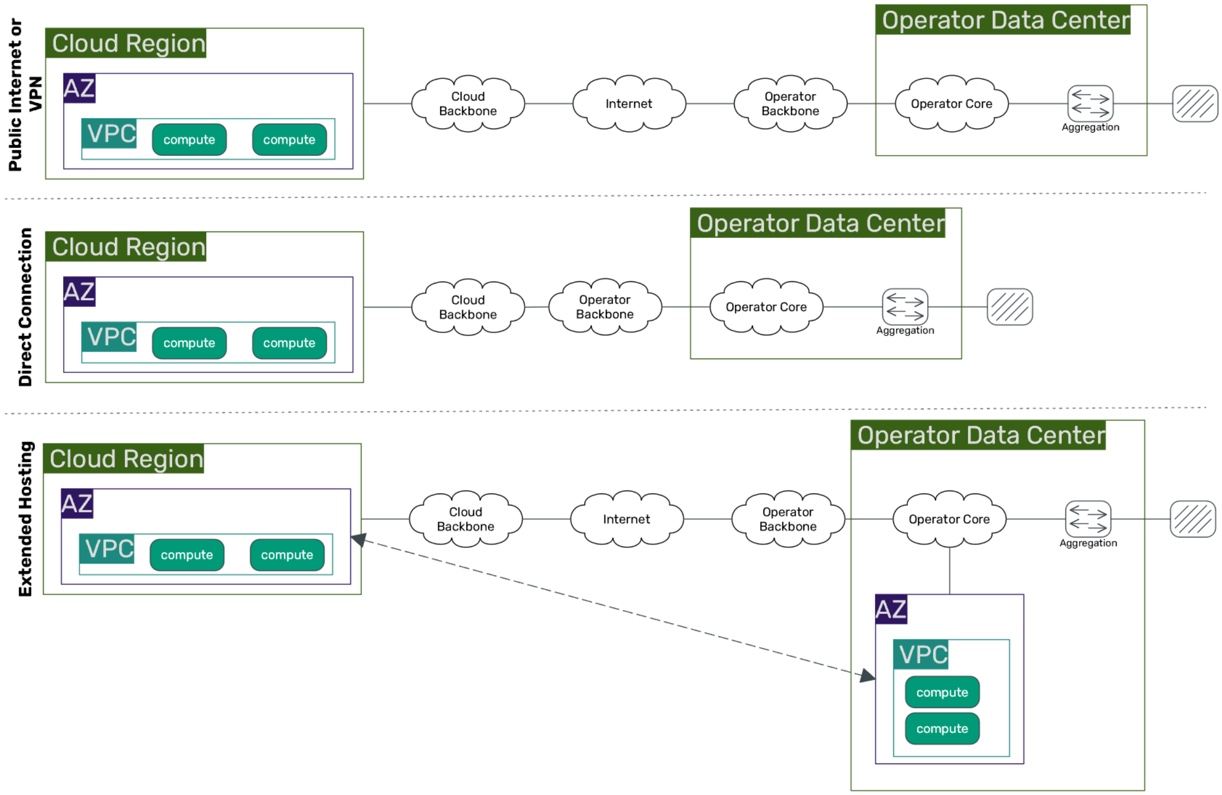

Figure 5 - Cloud Connection Types

One option is to use "accelerator" functions to attempt to route primarily on the cloud operator backbone and stay off the general Internet. This may be particularly attractive if the MSO already has peering established with the cloud operator. Another option is establishing a direct connection physically between two data centers; however, this requires presence in a common location and provisioning the connection. Another option for some cloud operators is to install their hardware directly into an MSO data center, which can then be provisioned through the cloud user interfaces.

With a diverse set of geographically located datacenters from cloud providers across North America, Europe, or Asia, operators should consider leveraging these locations for hosting their applications. It may also be required that cloud hosting be located in a specific region for legal, taxation, privacy, security, or other purposes. It's important to highlight that the MAC Manager will likely store and manipulate certain fields considered private identifying information in some jurisdictions and cloud data center geographic location will play a key role in complying with those regulations.

4.5. Backup and Retention

Depending on the nature of the cloud deployment, there are extensive options available for backup and retention of data. Virtual machines are often backed by block storage devices that can be snapshotted on a manual or scheduled interval. The same options are available for block storage devices attached to container images in an orchestrated containerized deployment. These block backup techniques can be achieved transparently to the VM or containers, reducing integration cycles associated with backup and retention mechanisms.

The various SaaS options, such as SaaS databases, often manage and monetize their own retention models, where retention rules can be set by volume, time, or other configurable parameters.

Some cloud providers offer additional choices for backup operations, such as large-volume cold storage, where data can be retained at very low cost, but retrieval or restoration of the data often comes at a higher cost. In some offerings, physical export of the data is possible as well.

Beyond the options cloud providers offer, direct connection data links into the cloud would allow for more conventional, self-maintained backup and retention policies. In the case of certain SaaS offerings, manual export and retention of bulk data may not be fully compatible with the software offerings, or across cloud providers.

5. Cost Centers

Cost centers associated with cloud-based deployments of FMA architectures can vary greatly depending on methodology chosen and services deployed. There are some consistent elements across cloud providers that will affect cost independent of the chosen architecture:

- MAC Manager compute resource hosting/consumption (VM or container)
- MAC Manager runtime storage, with a focus on IOPS
- Bandwidth usage of FMA-MMI and FMA-OSSI traffic
- Fixed data links between MSO datacenter and cloud datacenter
- Backup and retention costs

In a resilient architecture, regardless of whether the MAC manager is situated as a monolithic software package or a microservices-based model, some measure of cost will exist. Virtual machines operated by cloud providers typically present two costing options: hourly or reserved (fixed cost by term). Hourly-costed instances offer more flexibility in terms of the actual runtime of the virtual machine. For example, operators could choose to have cold standby backup instances which may offer a cost savings approach. Similarly, in microservices models, instances could be launched or deprovisioned to support batching operations for processing data in bursts instead of instances running continuously.

Reserved instances are more cost effective outside of these types of operations. Reserved instances cost less than the equivalent hourly instance when run for the same amount of time but require a contractual lock-in over a monthly or yearly term. As a result, reserved instances require greater understanding of operational needs in advance of the actual deployment.

Storage access, particularly in IOPS, can represent a significant area of cost. Different implementations of MAC Managers may have very different IO profiles, Resilient data storage will play a key role in any MAC Manager implementation.

Bandwidth costs will vary based on several factors. Principal among them is the number of deployed RMD devices connected to MAC managers, as well as configuration on reporting thresholds and intervals of telemetry, IPDR, and logging data. As a general rule, when surveying major cloud providers, we found an asymmetric cost associated with data transfer: ingress data is cost-free and egress data is costed at total data transfer across tiered pricing intervals.

In FMA, outside of actual bearer traffic, there are two inherent modes of traffic, each bidirectional in nature but also asymmetric in the volume of data transfer. In a robust cloud-based deployment, recommended setup would have redundant or backup MAC managers distributed across availability zones to ensure impact of outage has a small footprint in terms of service impact, scope, and time of impact.

In general, bandwidth usage and anticipated cost is summarized by the following:

Table 1 - Bandwidth Usage Summary

	RMD to MAC Manager		**MAC Manager to RMD**	
FMA-MMI	High throughput	No cost	Low throughput	Costed
	MSO DC to MAC Manager		**MAC Manager to MSO DC**	
FMA-OSSI	Low throughput	No cost	Medium throughput	Costed
	MAC Manager inter-AZ communication			
MAC Manager HA	Medium throughput	Costed		

The final cost center is in bridging the MSO datacenter into the same network space as the cloud provider. The major cloud providers each offer their own variant on this high throughput, dedicated secure network link. Datacenters and links are regionally distributed, meaning that an MSO datacenter in the eastern region of the US could have a direct, non-public link into geographically matched cloud provider networks.

Similar to virtual machine costing options, these fixed and dedicated links can be found in hourly or fixed-term options, with the corresponding flexibility versus cost optimization consideration.

6. Experimental Cloud Architecture

By disaggregating the DOCSIS processing from management concerns, FMA has introduced flexibility and design choices in deployment strategies. Data plane concerns of throughput, latency, and cost-per-bit are left to be optimized by packet-processing silicon, while management plane concerns of flexibility, agility, and evolution are left to be optimized in the virtual, software-defined design space.

Shown below, we built a specific architecture to explore a Cloud FMA deployment, but other architectural choices are possible as well.

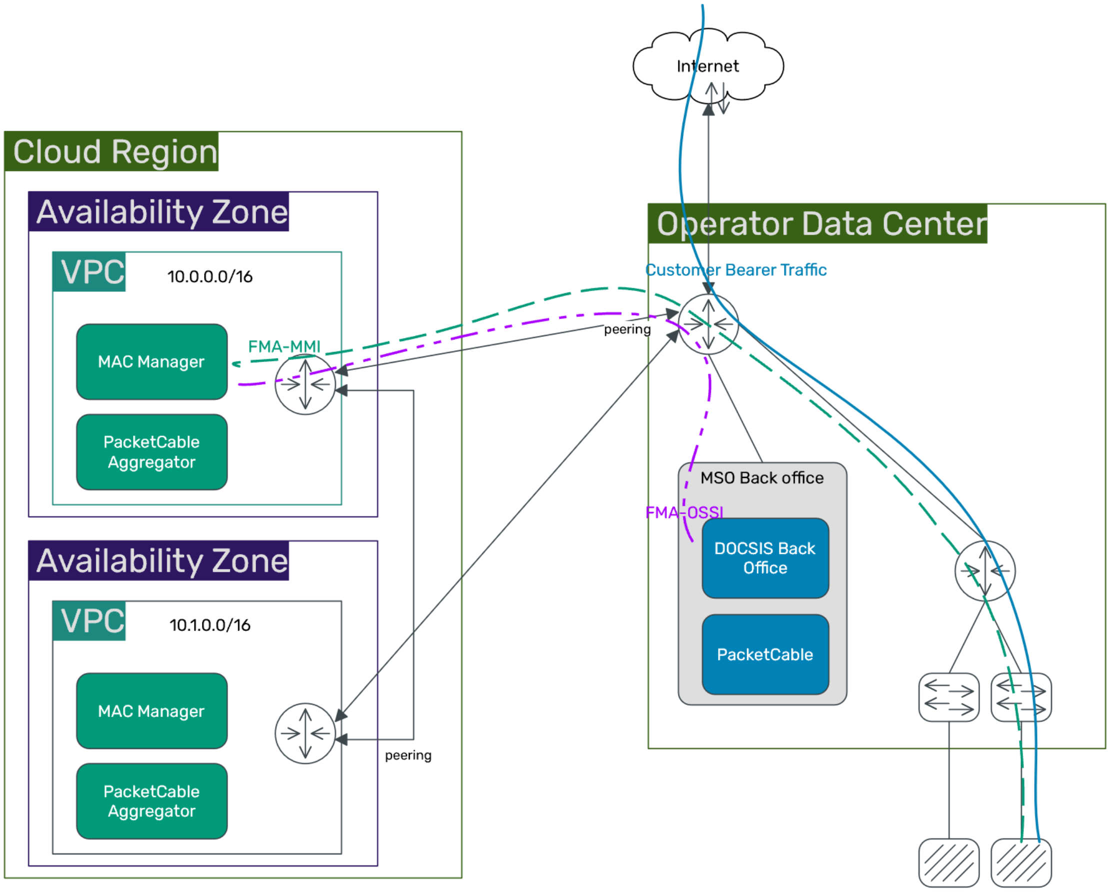

Figure 6 – Cloud-based FMA

A virtual private cloud (VPC) is an isolated network and set of cloud resources, such as compute servers, launched and configured in a cloud providers infrastructure. All resources within the VPC are private from other users, isolated from other networks including the Internet, and are assigned IP addresses from a private CIDR pool.

Resources launched within an Availability Zone (AZ) are on the same physical network, connectivity, and power infrastructure. Resources launched in two (or more) separate AZs are on unique physical infrastructure to provide redundancy and resilience, limiting the impact of an outage in one AZ. AZs are usually inter-connected with high-speed redundant links to allow for common data-replication techniques to be useable across AZs. To ensure high availability it's important to deploy applications across AZs.

Cloud providers offer different options when connecting an Operator data center to the cloud provider. Two common methods are, generically, a VPN and a direct connection. The VPN option is quick and easy to setup and provides an encrypted tunnel between the cloud and the operator network. While encrypted and secured, the VPN tunnel is routed over the open Internet and may have a variable performance profile and specific maximum throughput limitations. Another option is to directly connect from your own data center into the cloud providers data center. This direct connection is not software controlled and involves people from both companies to install and provision physical connections between the sites.

A VPN connection can be 'upgraded' to a direct connection without impacting the logical architecture of the solution. This allows for initial deployment trials to be setup with a VPN and later, optimized into a direct connection. For our setup, we used a VPN based connection between our deployment and AWS.

Once a connection between a VPC and the MSO data center is established, routing rules need to be installed to allow communication into and out of the VPC. Routing to the cloud over either connectivity option can be done with static routes or dynamically using eBGP. Given the small network in our testing we used static routing, but larger deployments will likely want to make use of eBGP.

We placed the MAC Manager and PacketCable Aggregator into the VPC and setup a static route to our physical infrastructure. The VPC was isolated from the Internet. The RMD to MAC Manager control communication transited our aggregation network, through our traffic router, across the VPN peer connection, and to the MAC Manager in the VPC. The RMD customer bearer traffic transited our aggregation network, through our traffic router, and out to the Internet. This deployment did not make use of PacketCable. Our implementation hosted firmware files for software downloads for the RMDs in the MAC Manager component.

7. Bandwidth

Bandwidth consumption over the Cloud connection is a primary concern when moving to cloud deployments. In an FMA deployment, the management plane traffic is separated from the customer traffic and the MAC Manager is not doing per-packet data processing. This disaggregated architecture allows for hybrid network deployments by placing management components in virtualized Cloud networks and keeping customer bearer traffic within the operator core network.

Bandwidth consumption between the MAC Manager and RMD will vary between vendor implementations and services deployed. However, to attempt to understand possible real-world consumption of the cloud provider transit, we investigated bandwidth consumption of an implementation of a MAC Manager and RMD, deployed in a 2x2 configuration, with a modest number of Cable Modems.

Communication between a MAC Manager and RMD in FMA can fall into one of two traffic patterns: steady state and on-demand. Steady state traffic is a continuous exchange of data during normal operation and on-demand traffic is bursty and usually triggered by an external command, such as a software upgrade.

We then classified the different streams of communication into the following categories:

Table 2 - Communication Classifications

Category	Type	Description
Telemetry	**Steady State**	**Regular streaming of status, operational, and statistics which the MAC Manager uses to monitor RMD population.**
Configuration	**On-demand**	**MAC Manager actions to configure changes in the RMD.**
IPDR	**Steady State**	**Regular streaming of customer related statistics to fulfill IPDR interface north of the MAC Manager.**
Support Info	**On-demand**	**Extra support and trouble-shooting data gathered and stored for historical data during support cases.**
Logs	**Steady State**	**Streaming of system logs.**
Heartbeats	**Steady State**	**Regular heartbeat and RMD discovery processes.**
Firmware Upgrades	**On-Demand**	**Download of firmware to RMDs.**
SSH/CLI	**On-Demand**	**Direct SSH/CLI connections to RMDs if needed.**

We monitored the communication between the MAC Manager and the RMD over time and through regular use, classified all protocol connections into one of the above categories, and aggregated the consumed bandwidth into an average consumption rate. The values are specific to our implementation but can provide an "order of magnitude" value to allow us to understand cloud provider transit costs.

Table 3 – Bandwidth Consumption by Classification

Category	Size	Downstream	Upstream
		Bandwidth	Bandwidth
Telemetry	**Constant**		**4.3 Mb/s**
IPDR	**Constant**		**1.0 Mb/s**
Support Info	**~ 30MB**		**76 Mb/s**
Configuration	**n/a**	**Negligible**	**Negligible**
Logs	**Constant**		**Negligible**
Heartbeats	**Constant**	**Negligible**	**Negligible**
Firmware Upgrade	**~ 131MB**	**116 Mb/s**	
SSH/CLI	**n/a**	**Negligible**	**Negligible**

The constant steady state traffic between each MAC Manager and RMD pair is about 6 Mb/s when communicating with a 2x2 RMD with a modest Cable Modem count. Telemetry and IPDR making up most of this traffic means that the consumption will vary with the number of services deployed within the RMD. To understand Cloud transit consumption, we need to take the steady state values and multiply them by the RMD population size served by the MAC Manager. So, with the experimental implementation, the MAC Manager deployed with 100 RMDs might constantly consume ~ 600 Mb/s (75 MB/s) of cloud transit. Some cloud providers also charge asymmetric rates, where "download" out of the Cloud is charged at a different rate than "upload" into the Cloud. Our investigation found most of the steady state traffic is "upload" from the RMDs to the MAC Manager.

The largest on-demand "download" operation was the software upgrade functionality. This function, managed by an operator, commands one to many RMDs to download their software upgrade file from the MAC Manager. The firmware file used in the test was about 131MB and the full download for a single RMD took about 10 seconds. While short lived, this consumption could be significant if an operator commanded an entire population of RMDs to download their firmware upgrade file concurrently. The FMA architecture does not require that the MAC Manager host SSD firmware files. If the cloud-based MAC Manager is aggregating many RMDs and an operator expects a high concurrent download demand, an attractive option is to host the SSD firmware files 'on premise' on a local HTTP server and simply issue the SSD command to download the firmware files from the locally hosted file server.

8. Latency

In an FMA deployment, the latency between the MAC Manager and the RMD can affect management plane traffic for configuration and status information but does not directly add to bearer traffic latency. This is due to FMA making the bearer traffic available at the first-hop aggregation switch rather than routing the bearer traffic through a core, such as the MAC Manager.

To better quantify the impact of latency between the MAC Manager and the RMD, we injected latency into our deployment and monitored the operational status of the deployment in the presence of latency. The latency was only injected between the MAC Manager and RMD connection and not in the data plane bearer traffic, which was separated at the first-hop aggregation switch and routed normally.

Typical Headend/Hub based MAC Manager to RMD one-way latencies we see in deployments are between 0.01ms and 8ms, inclusive of standard propagation delay and the overhead of switching elements.

The introduction of a routed network over a VPN into the cloud VPC adds latency to the MAC Manager to RMD connection. The amount of latency added by the cloud connection is highly variable and based on many factors, some of which are not within the operators control. We tested a connection to explore real-world latency values to ensure our latency tests would simulate a useful range of latency targets. The latency measurements were made from a single IP location in the US Southeast to load balancers in our cloud provider network (AWS in this setup). Values are round-trip time averages.

Table 4 - Cloud Latency Measurements

Zone (US Southeast to…)	Average RTT (ms)	Std Dev (ms)
US Northeast	30.37	9.16
US Central	64.79	13.52
US Northwest	85.99	17.43
US Southwest	79.26	14.82
Europe (Frankfurt)	132.08	17.31
Asia (Tokyo)	254.31	26.26

The overseas values were included as interesting data points but are not relevant to our testing and not discussed further due to their performance and the legal and regulatory implications of hosting a MAC Manager across international borders. Focusing only on the continental US results, we see significant result differences between the AWS regions from our RMD deployment in US southeast. It’s expected that countrywide RMD deployments will likely want to be connected to MAC Managers deployed in cloud data centers with the best performance from the RMD location. In AWS, for example, we would want MAC Managers deployed in each of the 4 major regions and have RMDs connected within a single region.

For our testing, to ensure controlled injection of a range of latencies, we built a controlled network with as few switching elements as possible and used a network testing tool to inject the specific latency targets.

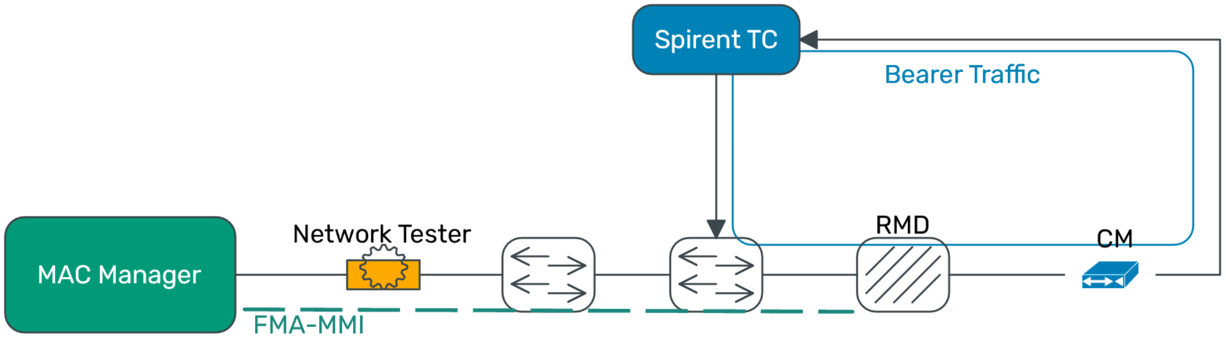

Figure 7 - Latency Test Setup

For our investigation, we choose a range of 1-way latencies between 10ms and 50ms, equivalent to 20ms - 100ms round-trip time (RTT). 10ms (20ms RTT) being worse than current on-premises, real-world deployment cases and 50ms (100ms RTT) being an upper range beyond latencies we saw with our real-world connections to our cloud provider.

When normalized to propagation distances, this results in a range as follows:

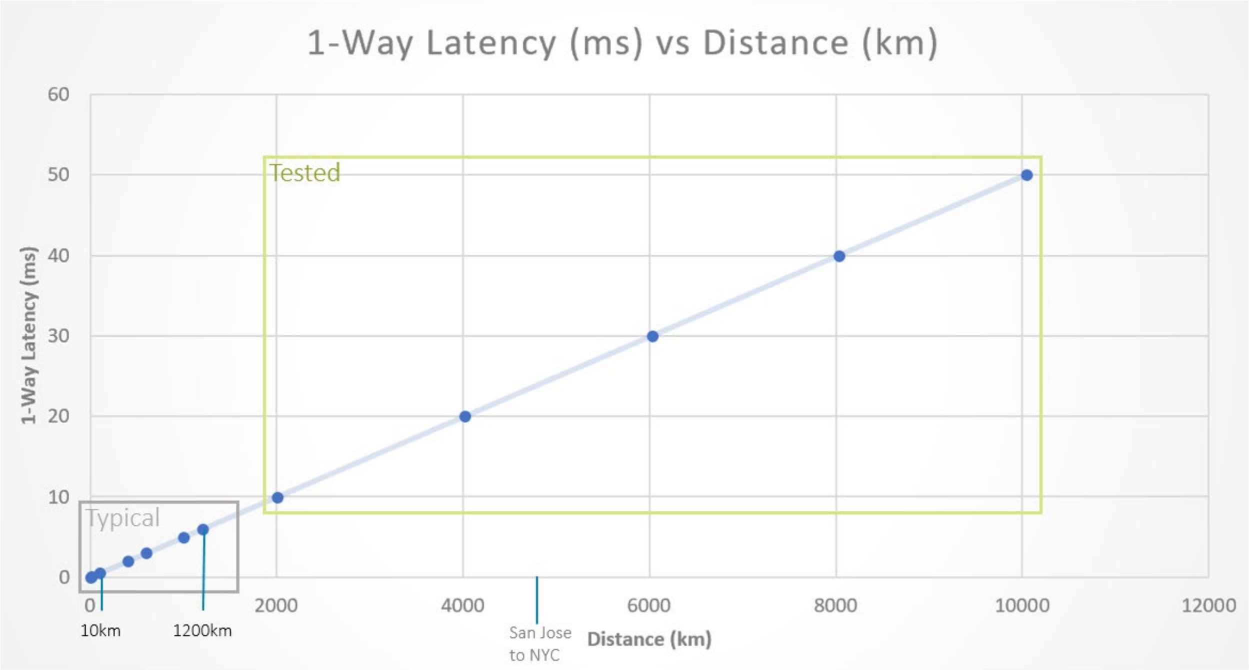

Figure 8 - Latency vs Distance (propagation)

We tested the same use cases as in the Bandwidth section, specifically:

- RMD Software Upgrade downloads
- RMD Support Info uploads
- MAC Manager to RMD Control connection
- MAC Manager IPDR (bulk data)

RMD Software Upgrade downloads represent the single largest bulk download operation from a MAC Manager to an RMD. These files are downloaded over a TCP connection to ensure reliable delivery of the upgrade file. Firmware image files sizes are highly variable between RMD implementations. In our test, the upgrade file was 131MB in size. Additionally, the specific RMD implementation we used in our test throttles firmware upgrade download speeds to 120Mb/s to ensure safe transport in the presence of other network traffic, which is visible in the baseline result:

Table 5 - Firmware Download

# of RMDs	Baseline	20ms	40ms	60ms	80ms	100ms
1	9s	10s	11s	13s	15s	19s
20	12s	32s	81s	101s	123s	144s

TCP connections used for bulk transfers are sensitive to latency due to TCP being a protocol that requires an acknowledgement from the receiver before more data is transferred. TCP utilizes a window-size scaling algorithm that accommodates a progressive increase in the amount of data transferred per acknowledgement up to a threshold. When latency is present, a throttling effect can come into play ensuring that the data has been reliably transferred.

In the FMA model, firmware is downloaded and stored on the remote devices directly and is not downloaded during each reboot by a bootloader in the RMD. This means the firmware upgrades are only issued within the FMA system when new firmware is provided by the vendor and approved for distribution by the operator. We expect firmware upgrades to be somewhat infrequent and associated with a maintenance window. The additional impacts of latency to the firmware download within this context are minor.

The RMD Support Info files are on-demand uploads from an RMD to the MAC Manager used during support activities. The size and contents of these files are vendor-specific, but the transfer mechanisms are standardized in FMA. In our RMD implementation, these files are between 2-30MB, depending on RMD history data files, and we used a 17MB file size during the testing.

Table 6 - Support Info Upload

Measure	Baseline	20ms	40ms	60ms	80ms	100ms
Time	1.8s	1.8s	2.1s	2.6s	3.4s	4.2s
Bitrate	76 Mb/s	76 Mb/s	64 Mb/s	53 Mb/s	41 Mb/s	32 Mb/s

We also tested the operational behavior of the system under latency conditions. Latency plays a complex and not directly measurable role in the operational behavior of the other connection types, so we tested the impact of latency as to a user of the function.

Table 7 – Functionality Impacts

Function	20ms	40ms	60ms	80ms	100ms
MM to RMD Control Connection	No issues	No issues	No issues	No issues	No issues
IPDR	No issues	No issues	No issues	No issues	No issues
CM Remote Query	No noticeable delay	No noticeable delay	No noticeable delay	No noticeable delay	No noticeable delay

The MM to RMD Control connection is a TCP connection transporting YANG-based object models. The transported data is much smaller and more intermittent than the previous bulk transfer leading to negligible system impact, despite the latency introduced by the TCP Ack RTT. Another mitigating factor for the Control connection is that the DOCSIS MAC is housed within the RMD itself, further reducing the systems impact of latency in the Control connection since there is no MAC signaling between the RMD and the MM.

The IPDR connection is between the IPDR collector and the MAC Manager and the IPDR protocol was not negatively affected by the latency injection. The MAC Manager has an internal cache for fulfilling IPDR data requests and a real-time TCP control connection round-trip for configuration and maintenance aspects of IPDR, which is where latency injection would impact IPDR operation. During each of the injected latency tests, our IPDR collector did not have any issues gathering required IPDR records from the MAC Manager.

The CM Remote Query function has the MAC Manager gather SNMP operational data from all subtended Cable Modems on regular intervals and cache the values within the MAC Manager. The SNMP protocol is "chatty" with many packet exchanges during SNMP operations which would be penalized by our injected latency. This test was to ensure that SNMP CM Remote Query would not have operational issues in the presence of high latency during the SNMP exchanges. We found from MAC Manager internal metrics and user tests that the additional latency did not impact the ability for the MAC Manager to provide the CM Remote Query functionality.

Our conclusion is that the FMA architecture is resilient and robust in the presence of latency between the MAC Manager and RMD components. The MAC Manager functions we found most impacted by the additional latency, bulk downloads/uploads, are still completed within acceptable ranges and, more importantly, are robust in the face of the additional latency.

9. Conclusion

The Flexible MAC Architecture decouples data plane and management plane concerns and provides a strong distributed access architecture in hybrid-network deployment models. With the MAC Manger control traffic decoupled from customer bearer traffic, the MAC Manager can be placed in a virtualized hybrid-cloud deployment without introducing latency on the bearer traffic. Furthermore, since customer bearer traffic is not routed through the MAC Manager, this traffic can be processed by high-throughput, low-latency specialized equipment designed specifically for Ethernet/IP packet processing.

We explored an implementation-specific MAC Manager to gather real-world tests to validate the FMA decoupling approach in variable-latency environments. We also gathered an implementation-specific baseline of bandwidth usage to validate relative or proportional utilization of the link bandwidth transit demand that could be seen in MAC Manager implementations.

For many operators, a hybrid-cloud approach utilizing AZs, VPCs, and compute engines to host FMA functionality may provide an agile framework to start small and pay-as-you grow into more advanced services provided by the major cloud operators. FMA is uniquely suited to agile approaches due to the separation of management and data plane traffic, allowing packet processing of data plane customer traffic to be managed separately and completely "on-premises".

Abbreviations

API	Application Programmer Interface
AWS	Amazon Web Services
AZ	Availability Zone
BGP	Border Gateway Protocol
CIDR	Classless Inter-Domain Routing
DAA	Distributed Access Architecture
DOCSIS	Data Over Coax Service Interface Specification
eBGP	External/Exterior BGP
FMA	Flexible MAC Architecture (CableLabs standard)
HA	High-Availability
HTTP	Hyper-Text Transfer Protocol
IOPS	Input/Output Operations Per Second
IPDR	Internet Protocol Detail Record
MAC	Media Access Controller
MAC-NE	MAC-layer Network Element
MACPHY	MAC layer and PHY layer, apropos networking stack
MHAv2	Modular Headend Architecture v2 (CableLabs standard)
MM	MAC Manager
MMI	MAC Manager Interface
OSSI	Operations Support System Interface
PAG	PacketCable Aggregator
PHY	Physical layer, apropos networking stack
RMD	Remote MACPHY Device
RPD	Remote PHY Device
RTT	Round-Trip Time
SaaS	Software-as-a-Service
SSD	Secure Software Download
VPC	Virtual Private Cloud
YANG	Yet Another Next Generation – Data modeling language for NETCONF

Bibliography

CableLabs. (n.d.). *Flexible MAC Architecture System Specification I02.* Retrieved from https://www.cablelabs.com/specifications/CM-SP-FMA-SYS

Spirent. (n.d.). *Attero Ethernet Network Emulator*. Retrieved from https://assets.ctfassets.net/wcxs9ap8i19s/2ikhr9AEdDBsY7xJOWdtzg/8b56d702abb60c67baad8e3df0d6e473/DS-Spirent-Attero-and-Attero-X.pdf