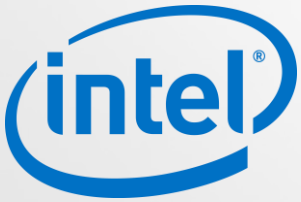
Independent market research and competitive analysis of next-generation business and technology solutions for service providers and vendors

HEAVY READING WHITE PAPER

Virtual Probes for NFVI Monitoring

A Heavy Reading white paper produced for Intel and Qosmos, a division of ENEA



www.intel.com www.qosmos.com

AUTHOR: GABRIEL BROWN, PRINCIPAL ANALYST, HEAVY READING

SOFTWARE PROBES IN NFV ENVIRONMENTS

Operators worldwide are seeking to "software-ize" their networks to capture cost savings from commercial off-the-shelf (COTS) hardware and gain operational agility from the move to automated cloud infrastructure. The intent is to make networks better able to adapt to changes in the online services market and to equip operators to more quickly seize commercial opportunities as they arise.

The network functions virtualization (NFV) initiative led by the European Telecommunications Standards Institute (ETSI) has served to catalyze this transition to software-based networks and provides a globally recognized architectural template. The NFV model comprises three main elements: NFV infrastructure (NFVI), virtual network functions (VNFs) and a management and orchestration (MANO) suite. These elements work together to provide network services.

This white paper argues that in next-generation, NFV-based networks, the ability to monitor traffic and service performance is critical and should be available across both physical interfaces and virtual interfaces. The paper discusses how monitoring should be integrated with the NFVI and makes the case for a virtual probe (vProbe) function to be deployed between virtual network elements. This monitoring capability should offer standard application programming interfaces (APIs), supporting real-time online and offline views to higher-level analytics and management functions.

Operator Need for NFVI Monitoring

As networks transition to software and NFV, it is clear that monitoring solutions must adapt in tandem. The NFVI is the platform used to run the VNFs that combine to create network services. It comprises the compute, storage and networking components with a virtualization layer that abstracts the application (VNF) from the underlying hardware. This pool of resources is managed by a Virtualized Infrastructure Manager (VIM), such as OpenStack, which itself is a critical part of the NFVI. The NFVI should run VNFs from multiple vendors without adaptation.

The NFVI should also meet stringent key performance indicators (KPIs) for performance. Service provider networks are critical infrastructure, with strict reliability requirements that do not permit for downtime due to failed hardware or applications. This places great importance on the performance of the NFVI, since it must run mission-critical VNFs and be able to adapt to component failure without impacting services. For example, if a server or switch fails, an OS crashes or an application freezes, the NFVI, in association with the MANO and VNF managers, should support a seamless failover to new hardware, or a new software instance, to maintain the network service.

In many senses, NFV moves responsibility for service continuity away from the network function itself into the MANO and NFVI – that is to say, into the network cloud platform. Classical physical appliances are designed with built-in redundancy at the line card and chassis level, and are then deployed in redundant configuration (1+1 or N+1) in the network. This helps to ensure failover and continuous operation of the network service. This works well, but is expensive and hard to adapt to changing circumstances, limiting the operator's commercial opportunities.

The NFV model moves some (but not all) of the responsibility for failover to the cloud platform. The intent is that the redundancy and resiliency are inherent to the platform so that operators no longer need to spend as much on redundant equipment that may only be used rarely. This model allows for simpler VNF design (so-called "lean VNFs") and will result in the emergence of cloud native applications.

Meanwhile, the NFVI (a.k.a. the NFV cloud platform) must monitor the performance of the infrastructure to enable scaling out of VNFs, and associated virtual machines (VMs) or containers, according to demand, or to recover in the event of failure. Essentially, NFV apes the model used by hyper-scale cloud providers, which has been shown to offer high-reliability services using low-reliability components.

Challenges With the Hardware Approach

Operators typically use a range of hardware elements to monitor their networks, depending on the layer and service in question. Ethernet NIDs, for example, can be used to monitor Layer 2 services, probes can be used to gain visibility into various traffic types, and packet brokers can be used to correlate data crossing multiple interfaces. This tried and tested approach works well, albeit at relatively high cost.

There is an obvious problem, however: physical probes cannot access the logical interfaces between internal VM-to-VM communications to monitor functions hosted on the same server or between VMs located on different servers connected via virtual overlay networks. This will become even more of a problem in the future because new virtualization techniques, such as containers tend to rely heavily on distributing the traffic between many light-weight, even transient, virtualized resources. Legacy hardware probes cannot serve such virtual network elements.

Monitoring the NFVI

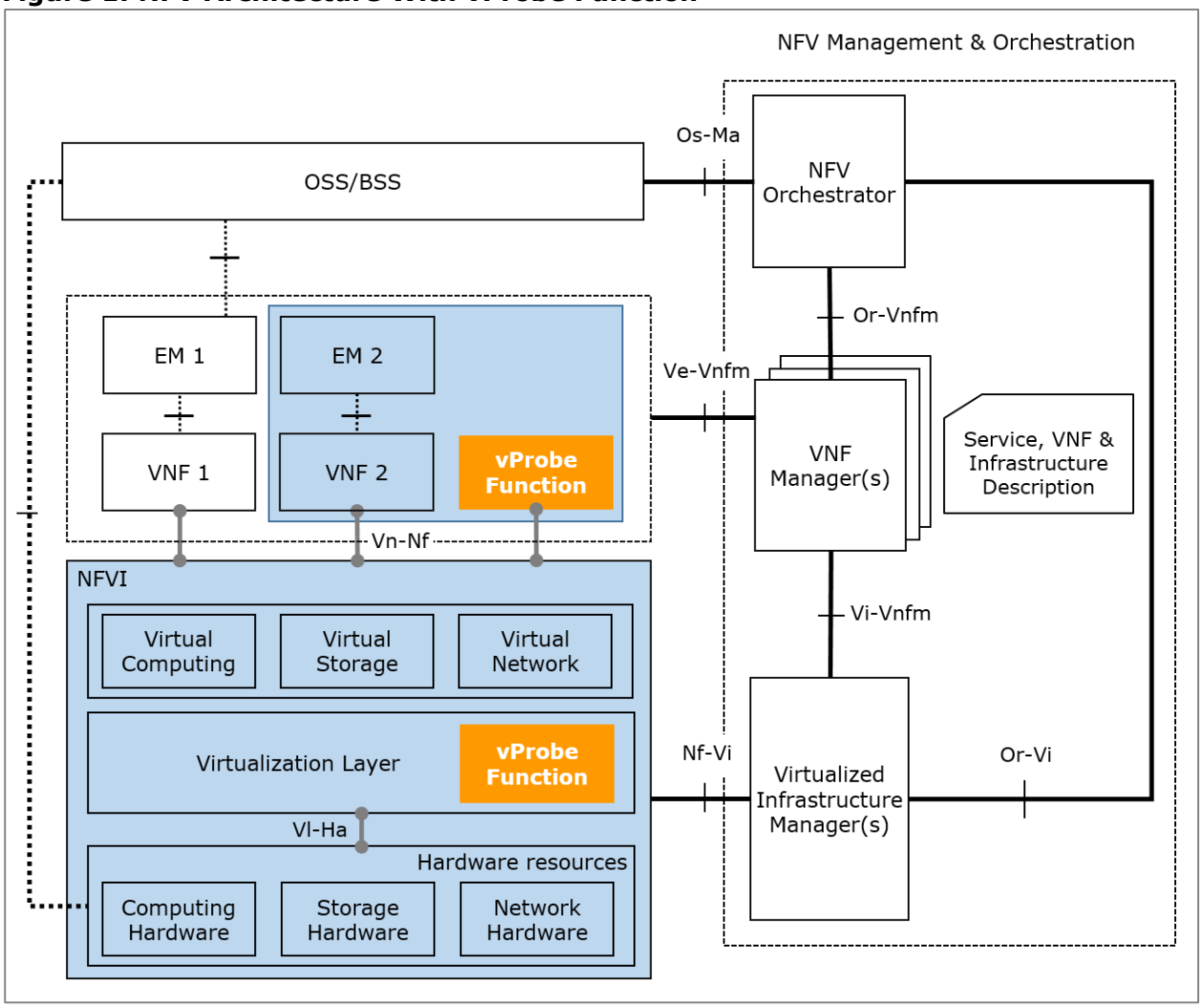
The development of monitoring tools has not kept pace with the vision that NFV will underpin critical telecom services. This poses a serious risk to the timetable of NFV itself. Without effective monitoring, the NFVI platform cannot be expected to run mission-critical networks. Virtual probes (vProbes) integrated with the NFVI are an important part of the solution to this.

Monitoring the performance of the NFVI is the job of the VIM (e.g., OpenStack) working in combination with a "resource orchestrator." These functions work in conjunction with a VNF manager to ensure VNFs have the compute and networking resources they need to meet the performance targets required by the service.

There are two different, but complementary, roles for the vProbe function in the NFV architecture, as shown in **Figure 1**. These are:

- A probing function deployed as a VNF to allow per tenant views (i.e., a tenant can only see traffic for its VNFs). This is useful in the context of an operator hosting services for multiple tenants or, for example, multiple user groups in the case of internal operator services or for large enterprise customers.
- A probing function deployed in the NFVI to monitor inter-NFVI traffic. Monitoring packet flows in this way contributes to the NFVI's stability and performance and can also provide a view of how the operator's network services are performing globally.

Figure 1: NFV Architecture With vProbe Function



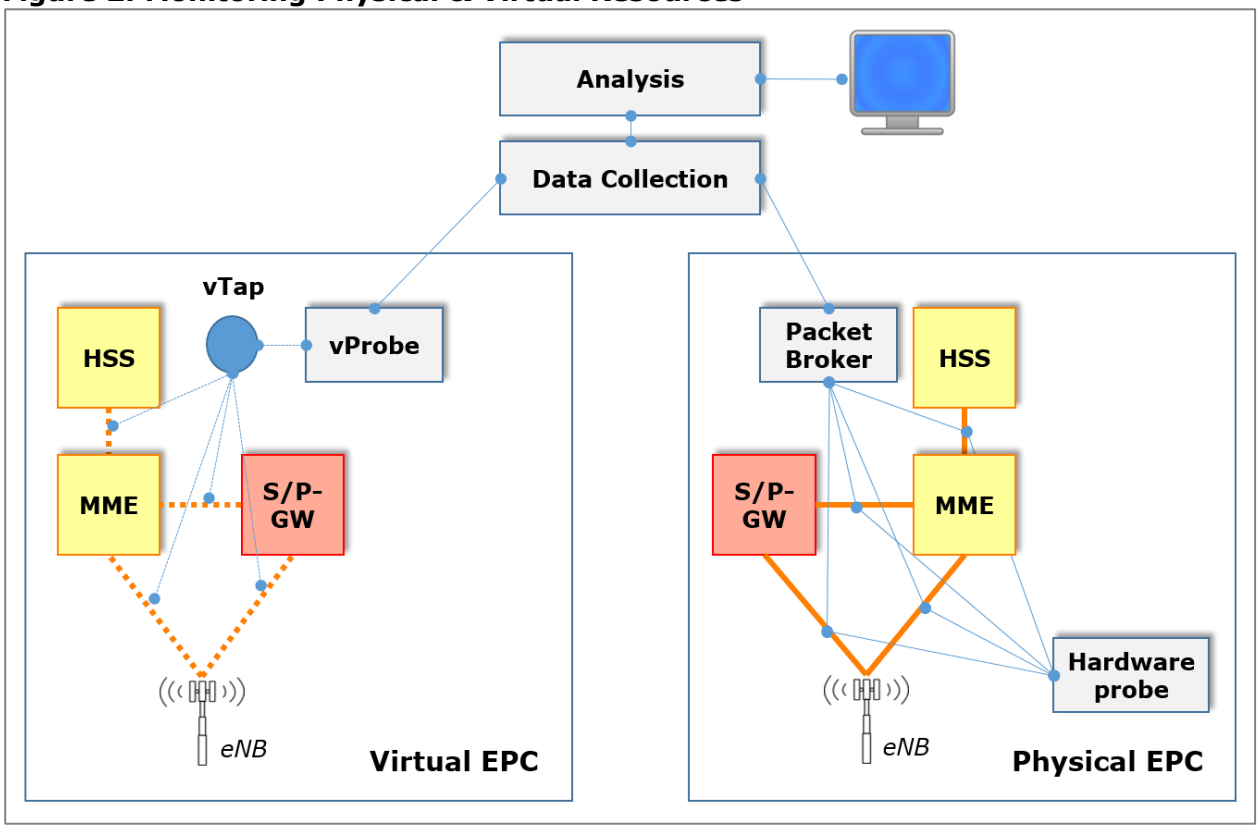
Source: ETSI, Qosmos

Hybrid Operations

The picture is also complicated by the fact that many operator networks will transition to NFV in a piecemeal fashion as parts of the network need to be refreshed or as the operator seeks to introduce new service capabilities. This means that network services will often run across both physical and virtual functions. Therefore, there is a need for monitoring solutions to operate in hybrid environments. This is illustrated in the mobile packet core example in **Figure 2**.

The implication is that the analytics tools that are used to interpret data should be capable of processing data from both types of physical and virtual probes. This should not be overly challenging, since analytics engines generally make use of multiple data sources. One interesting idea for hybrid environments is to use vProbes – i.e., running as a VNF in a VM – to also monitor the physical network interfaces in place of the hardware probe shown to the right in this figure.

Figure 2: Monitoring Physical & Virtual Resources



Source: Heavy Reading

A-CORD Monitoring as a Service

A good reference model for NFVI monitoring and analytics comes in the form of the monitoring-as-a-service model being developed by the Central Office Re-architected as a Datacenter (CORD) initiative. CORD is outside the ETSI NFV process, but uses many of the same concepts and technology components – for example, the CORD monitoring service leverages the open source OpenStack Ceilometer framework.

The CORD platform incorporates a generic "analytics" capability inside the XOS infrastructure. Some of the main requirements are:

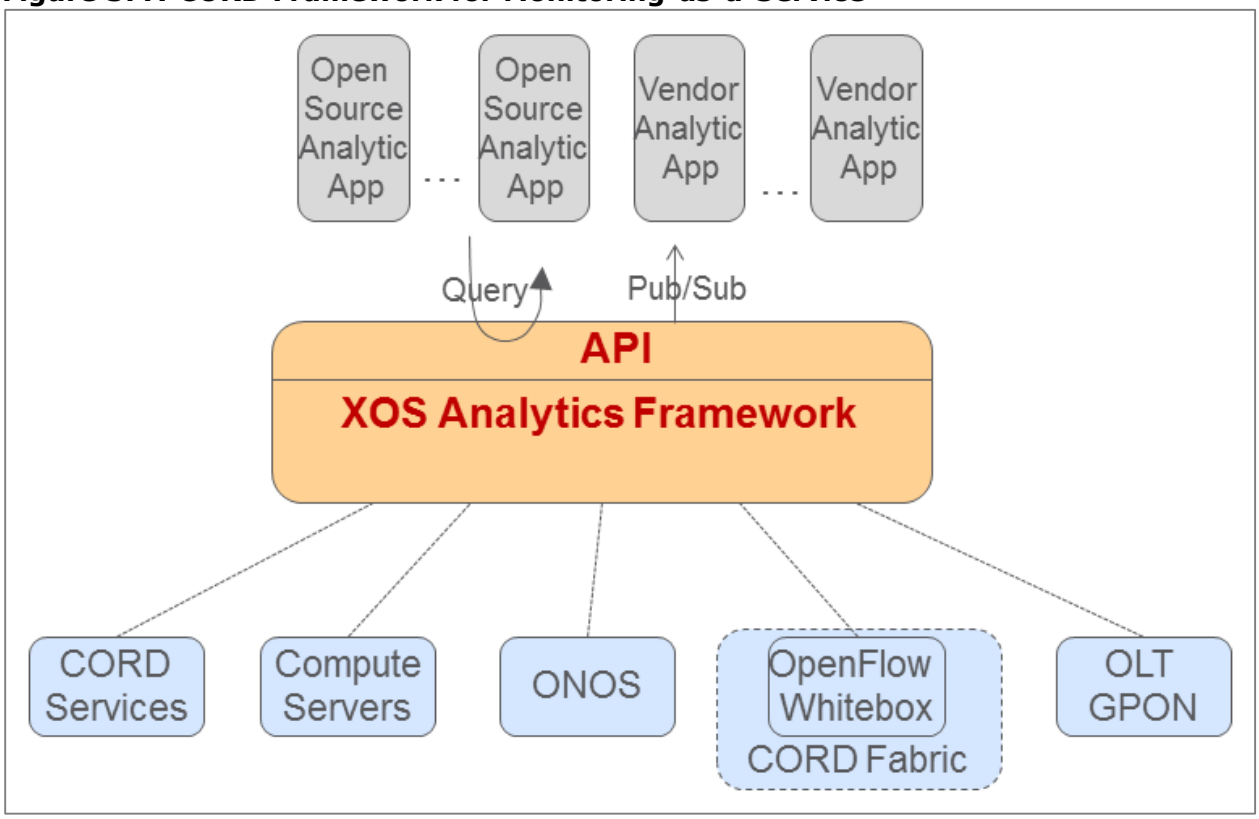
- That it is scalable and can support multitenancy
- Possible to instrument (control probing level on) services in addition to compute and network devices
- Can adjust the level of probing in the underlying devices
- To aggregate probing information where multiple data sources are present
- Is able to redirect data streams through a "probe VM" for deeper level of instrumentation that is not otherwise available from underlying devices

Figure 3 shows the framework for the A-CORD Monitoring Service (XOS is equivalent to parts of the NFVI). On its southbound side, the analytics framework interfaces with underlying

devices and services (servers, controllers, etc.) to accumulate information. This is done via both notification-based and polling-based measurement collection mechanisms.

Northbound, the framework interfaces to the analytics applications via reusable and open APIs. The analytic application correlates events from different network sources – including probes – into more meaningful events for performing real-time, closed-loop operations. Data models (in this case Django) are used to define the authoritative state for the network service.

Figure 3: A-CORD Framework for Monitoring-as-a-Service



Source: CORD Monitoring Service Paper

IMPLEMENTING SOFTWARE PROBES

The concept of a vProbe is now reasonably well established as a way to monitor VNFs and associated network services. Several vendors offer vProbe products that appear to work well and are low cost relative to hardware alternatives. In general, however, vProbes are designed to monitor VNFs and do not monitor traffic within the NFVI. As discussed above, this is problematic for high-reliability service provider networks.

Operator Requirements From vProbes

Operator requirements for next-generation monitoring solution across virtual and physical network interfaces can be summarized as follows:

- The ability to monitor dynamic VNFs. This is challenging since VMs are dynamic within the NFVI when they scale out/in. This underlines the need to incorporate NFVI monitoring in network service analytics.
- A set of reports and KPIs for network analysis based on call-tracing capabilities and trending reports.
- Ensure network and service monitoring software (based on vProbes) works in unison with a business intelligence suite.
- Be able to quickly provision monitor, troubleshoot, optimize and report all aspects of subscriber and network services.
- Monitor services across the range of access technologies the operator supports. This is particularly important for converged fixed/mobile operators.

Deployment of Software Probes in NFVI Environment

A vProbe is a software entity that can be attached to logical or physical interfaces. A vProbe can be instantiated as a VM, a container or a process belonging to the hypervisor hosting the VMs. In practice, we expect vProbes to be deployed in multiple types of hosts in the same network, with each location being able to provide particular information to the analytics system.

Integration in the hypervisor layer is a potentially important way to embed monitoring capability into the NFVI to help operators to monitor traffic, including VM-to-VM communications and on external physical interfaces to the platform. vProbes tightly coupled to the NFVI infrastructure add an important layer to existing service monitoring capabilities.

vProbes should support high granularity across dynamic VNFs and services. Multitenancy, for example, can be achieved by creating a lightweight proxy container for every tenant of the monitoring service, such that each tenant is able to access only to the instrumentation data of the network resources belonging to that service. Multitenant operation is fundamental to per service, per user monitoring.

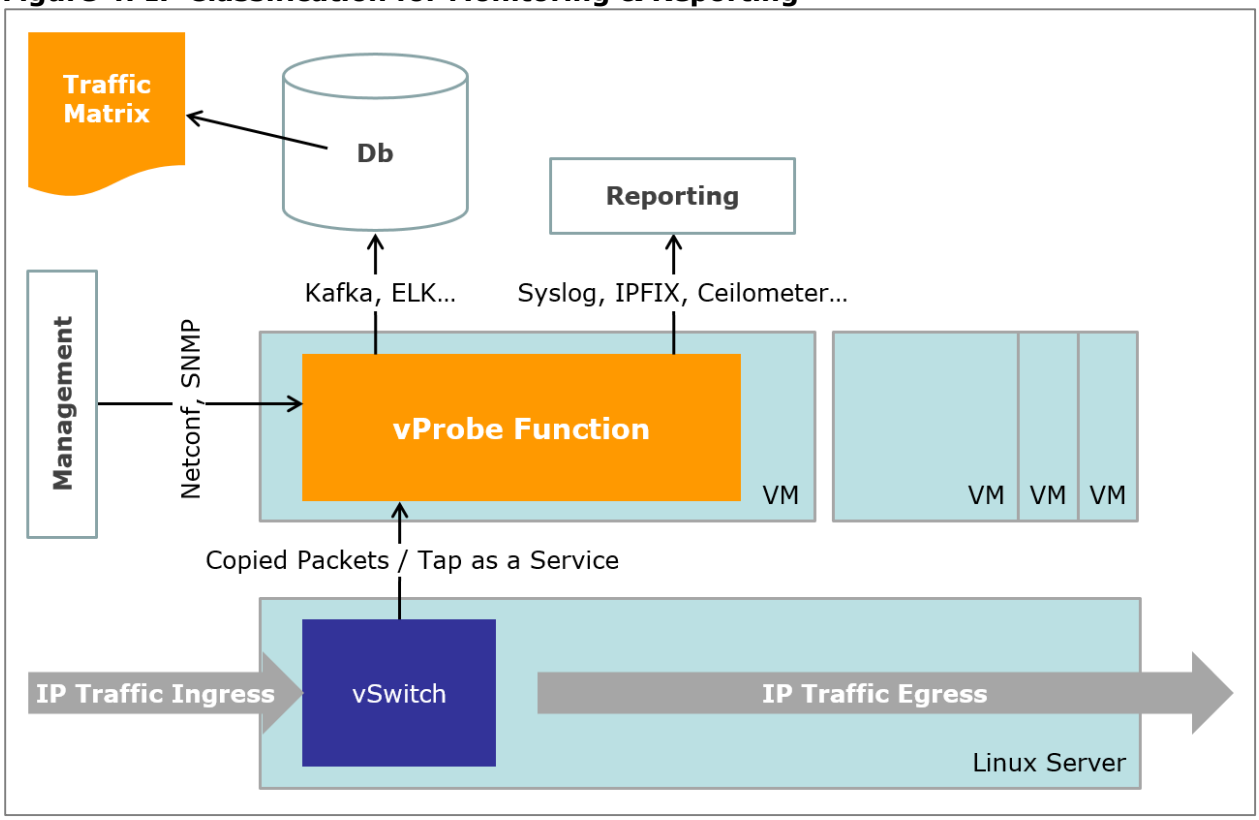
The process of embedding a vProbe in the NFVI is important. The basic method depends on the VIM and management tools that are incorporated in OpenStack, VMware, and so on. For example, the VIM can detect when a new vSwitch is deployed from OpenStack and then configure the switch to copy traffic to the vProbe using the address and port number. To work "out of the box" vendors will need to work with the NFVI suppliers to integrate their solutions.

Classification Engine

The heart of the vProbe – deployed as a VNF in a VM or container – is some form of packet classification engine. In this scenario, where the vProbe is integrated with the NFVI, packets enter the vSwitch and a copy is sent to a classification engine, as shown in **Figure 4**.

The classification engine analyzes packets and interprets them using pre-provisioned rules, before sending the information to a database and then onward to the analytics application itself. Pre-provisioned rules are expected to become less important over time, as artificial intelligence will be used to identify patterns and anomalies in traffic flows, and then interpret and act on that analysis accordingly.

Figure 4: IP Classification for Monitoring & Reporting



Source: Qosmos

Traffic Matrix Database

The vProbe should pass information over an API to a database that can store per-tenant and per-service data in a "traffic matrix" that allows for correlation of data across the service infrastructure by the analytics suite. The traffic matrix is a time-series data base that enables the aggregation and reporting of KPIs, which can be queried in real time or offline by analytics solutions to drill down into historical views of the traffic and service performance, as needed.

One candidate framework for this is the Ceilometer Project in OpenStack (also used in the A-CORD monitoring-as-a-service example discussed above). Since OpenStack provides infrastructure as a service (IaaS), it is necessary to meter its performance and utilization for billing, benchmarking and scalability over time. Ceilometer is a distributed database used to store monitor and meter OpenStack clouds and was designed to be extensible to other adjacent services. It is logical then, in these environments, to use Celiometer to collect and store data used for NFVI traffic monitoring. In other environments (e.g., VMware) an equivalent tool can be used.

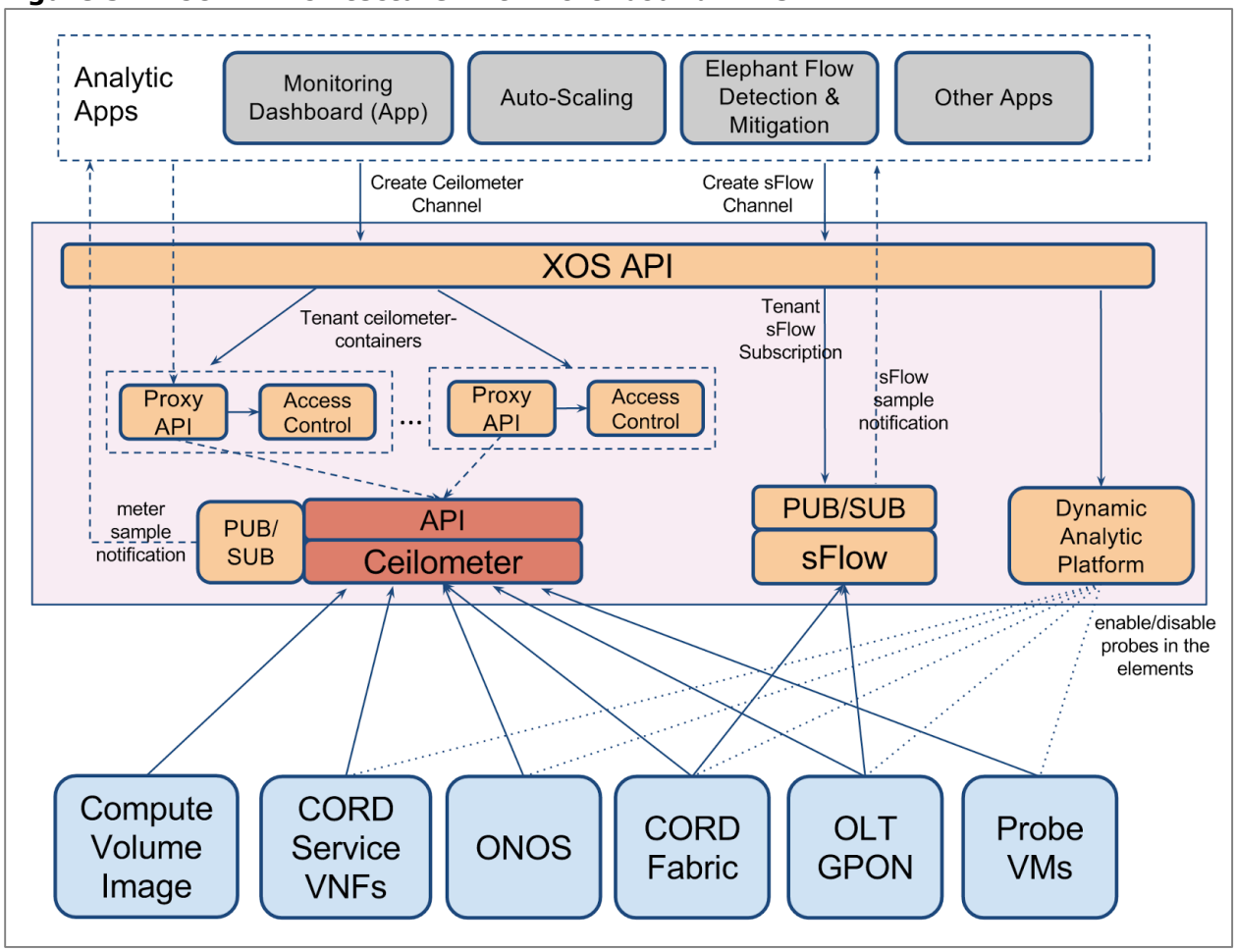
Monitoring inter-VM traffic enables operators to infer service performance using statistical analysis without monitoring the VM or VNFs themselves. However, to generate more reliable results, the analytics application can query the "traffic matrix," which includes data from multiple sources to cross-correlate data to better determine the performance of the end-to-end service.

Analytics APIs for NFVI Monitoring

Analytics and management suites need APIs to retrieve information collected from the network to perform deeper levels of instrumentation than would otherwise be available from the underlying devices. Once the analysis results in a decision, then different APIs (and corresponding data-modeling languages) are used to apply changes to the network or resources when needed. It is important that these APIs are widely supported by vendors.

Again, A-CORD provides a reference architecture, with northbound REST APIs from the ceilometer database embedded as part of XOS platform, as shown in **Figure 5**. The key question, of course, is the extent to which OpenStack and the CORD architecture will be adopted in service provider networks.

Figure 5: A-CORD Architecture With Northbound APIs



Source: CORD

The southbound APIs that enable the network orchestrator to make changes to the underlying infrastructure resources are, to some extent, out of scope of the monitoring solution. However, in practice there is widespread – and growing – support for the use of NETCONF/YANG for this purpose, especially where hybrid physical and virtual elements are used to create the service.

CONCLUSION

In next-generation, NFV-based networks, the ability to monitor traffic and services is critical. The transition to software changes the nature of the infrastructure and therefore the associated network services and performance monitoring solutions. The classic models, based on hardware probes and packet brokers, are not only challenged economically, but are also not able to monitor traffic within the virtualization platform. This is problematic for critical service provider infrastructure and means operators need new ways to instrument their networks.

This is driving the need for a vProbe function. These software probes can be embedded in the NFVI to monitor traffic between virtual network elements and pass this data to higher level analytics and management functions. This type of monitoring provides visibility into inter-VM traffic to help infer the performance of the end-to-end network service and in this way contributes to the stability and reliability of the NFV cloud platform itself. Used in combination with other network monitoring tools, it can provide a better view of performance than would otherwise be the case. vProbes themselves can also be used to replace traditional hardware probes regardless of the requirement to monitor NFVI performance.