Flavors of Closed-loop Implementation in Today's Hybrid Networks:

One Size Does Not Fit All

By Yuval Stein, Associate Vice President of Product Management, TEOCO

As Network Function Virtualization (NFV) continues to rapidly evolve, the understanding of closed-loop use cases is improving. What we are finding, however, is that there is no single 'picture' of a closed loop system, but in effect, a whole gallery of images with different themes and colors. Each use case is very different in its scale, in the complexity of the decision making, and in the required time constraints for execution. Despite the diversity, there does remain a common set of shared steps, as shown in Figure 1 below:

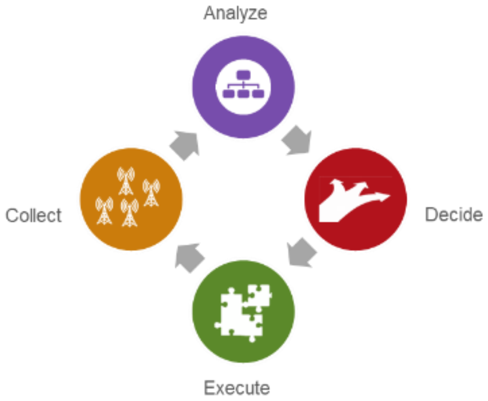

Figure 1 – The Closed-Loop Process

Work done by standards organizations (ETSI, NGMN, TM Forum), Open Source organizations (e.g. ONAP) and by communication service providers (CSPs), has resulted in several network architecture options, indicating where each one of these steps should take place. These options vary, depending on the following aspects:

- Centralized vs. Distributed Analysis for problem Identification
- Centralized vs. Distributed decision making and Policy management

Centralized vs. Distributed Analysis for Problem Identification

There are two main options for where Analysis should be done:

- Distributed (intra-layer) Analysis – good network architecture is typically layered, and component based, creating good reasons to try and embed functions in a layer or in a platform. With this approach, assuming the CSP is using efficient architecture concepts, the Assurance and Analysis function is embedded in each layer of the OSS solution (Customer, Service, Network Function, Infrastructures). Additionally, in cases where low latency is required, it is better to localize any possible function.

- Centralized (Global) Analytics – in this option (Figure 2, below), the OSS system includes a single Analytics module that can analyze data across the various layers. The main advantage of this option is the benefit of cross-layer data sources with analytics based on advanced algorithms & artificial intelligence.

The way to decide which option is best for you is to consider the various use cases. Possibly, the two options can co-exist in creating a hybrid Analysis architecture, but I would argue that in many cases there is a need for data analysis to be more holistic than expected.

Let us examine a common use case, one that might appear simple at first glance. In this scenario, let us assume that insufficient processing power has been identified for a specific VM or a VNF. The simplistic approach would be to allocate more resources to that VM/VNF if possible. Yet, a more holistic approach should consider the root cause of the problem, and apply logic like the following:

- If the need for more processing power is "natural" (and that is to be defined, or self-learned), additional processing power should be allocated.
- If the need is the result of a sudden exceptional behavioral of a specific VM/VNF, the resolution may be to reset/restart/reconfigure that VM/VNF.
- If the need is coming from a global phenomenon (such as bad weather or even a large-scale sporting event), the course of action may be totally different. As the whole system may be very busy, a decision can be made to preserve only vital VMs/VNFs to keep the network running in tough conditions.

As with many areas of modern life, problems can quickly escalate and cause other problems. In the case of a global phenomenon, handling each VM or VNF separately may be the wrong approach, and cause severe system problems somewhere else in the network.

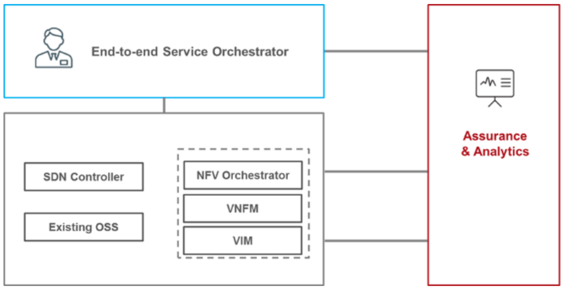

Figure 2 - Centralized Assurance & Analytics

Centralized vs. Distributed Decision Making and Policy Management

A lot of work has been done by several organizations in recent years in setting policy management standards, such as developing a conceptual framework using the ETSI definitions of PDPs (Policy Decision Points) and PEPs (Policy Enforcement Points).

To begin with, it needs to be pointed out that policy management and decision making are not the same thing. Policy management is the mechanism to manage decisions, while the decision making itself can sometimes be done by software modules that specialize in specific decision-making functions, potentially leveraging artificial intelligence to the level of self-learning. In these cases, the policy manager is just delegating the decision to the specialized module.

Decision making itself can vary in its complexity, starting from simple automatic static responses that can be pre-determined (e.g. when the temperature gets too high air-conditioning should be restarted), to more complex, ad-hoc responses that rely on AI (e.g. plan network changes that will provide optimal coverage to some geographical area). CSPs need to keep in mind that the complexity of the decision making may affect the architectural location of the PDP.

There are two main alternatives for a policy-based architecture:

- **A Centralized Policy Manager** for making OSS decisions at all layers (e.g. the ONAP current approach). The technical advantages of such a policy management mechanism are clear, the same mechanism can be used by all layers and the administration is much easier.
- **A Distributed Policy management system**– Each Orchestrator/Manager has a Policy component. We need to keep in mind that in this scenario, each kind of a management system oversees a different layer and possibly of different aspects of the network, so even if the policies are defined using the same tools, they are very different in their content. Additionally, when the decision making is simple but low latency is required, it makes sense to have the PDP as close as possible to the PEP. That said, distributed policy management can still rely upon a single policy management solution.

In both cases, PEPs will be located as close as possible to their related functions.

It seems that in the coming years, we are going to see networks that rely on both options, and even hybrid approaches, where a centralized policy manager uses specific “islands” of local policy management, possibly using a hierarchical approach.

Cascading Policies

Next, let’s have a look at managing *cascading policies (Figure 3)*. This is a common use case that is handled manually today but should be automated in a virtualized system.

The need for zero-touch deployments and zero-touch operations raises the requirements for Cascading Policies. Some automation use-cases dictate ‘flows,’ where a policy at a higher orchestration and management level is automatically translated to lower level policies. This is applicable to both Assurance and Fulfillment functions. The concept of cascading policies can naturally be implemented in both centralized policy management and distributed policy management.

Each ‘service definition’ should ideally specify its service level agreement (SLA). This may include KPIs such as guaranteed bandwidth, accessibility rates, Recovery time, etc. The service definition should then be translated by the Assurance and Analytics system to a monitoring scheme that collects measurements from the network and executes a ‘threshold crossing logic’, which is a type of Assurance policy. For this process, an automated flow, or cascading set of policies, should exist that can define a required assurance policy resulting from an orchestration policy.

When cascading policies are implemented in a distributed policy management system, it is also important for the distributed policy components to support the administration of policies using APIs. The hard work of building these necessary APIs is being done today within the TM Forum Interfaces Group. For example, the newly published PM Threshold API and the forthcoming SLA management API will serve this exact purpose- supporting assurance policies in a distributed policy management architecture.

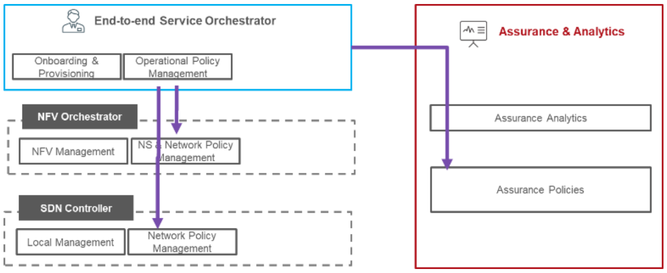
End-to-end Service Orchestrator
Onboarding & Provisioning
Operational Policy Management
NFV Orchestrator
NFV Management
NS & Network Policy Management
SDN Controller
Local Management
Network Policy Management
Assurance & Analytics
Assurance Analytics
Assurance Policies

Figure 3- Cascading Policies

We therefore see that closed-loops have different architecture implementation options, especially in the aspects of centralized/distributed detection analytics and centralized/distributed policy management and decision making. Centralized detection analytics appears to be the preferred approach, even if some analysis can be localized; but for policy management, both options may exist. This is an area that may significantly evolve during the coming years. Either way, the cascading policies concept will be required, regardless of which policy management architecture is selected.

In conclusion, closed-loops will have different architecture implementation options depending on many variables. Fortunately, the industry is working to resolve these issues and help service providers around the globe create networks that will enable tomorrow's services. For more information on how TEOCO is contributing to the development of architecture systems, standards and APIs for the communications industry, visit http://www.teoco.com/about-us/industry-associations/