

## **Persistent Operation of OF@TEIN+ Playground Verified by SmartX Multi-View Leveraged Visualization**

Muhammad Ahmad Rathore<sup>1</sup>, SeungHyung Lee<sup>1</sup> and JongWon Kim<sup>1 +</sup>

<sup>1</sup> School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology,  
Gwangju 61005, Korea

**Abstract.** In OF@TEIN+: Open/Federated Playgrounds for Future Networks (SDN/NFV/Cloud-integrated) testbed, the persistence operations of infrastructure is extremely critical to ensure availability of physical/virtual resources and identify key performance bottlenecks, which can degrade testbed operations. Persistence-enabled operations are increasingly evident in testbeds, and as a result, there is a growing focus on monitoring its long-terms effectiveness and effectively utilizing the deployed services, enabling end-users to reliably utilize playground applications. Significant challenges in measuring persistence, however, contribute to both a tendency towards identifying efficient metrics with resilient functionalities and effective verification methods. In this paper, we present our idea of maintaining persistent operation of OF@TEIN+ playgrounds and verify through visualization of gathered operational data.

**Keywords:** Persistent operations, multi-layer visibility, visualization, cloud-native

### **1. Introduction**

Maintaining a steady state of monitoring has always been a key element in ensuring the sustainable performance of complex distributed systems, being the first step to control quality of service, detect failures, or make decisions about resource allocation. Furthermore, enabling end-users (developers, etc.) for effectively utilizing the available services is crucial for persistent operations. Aligned with Future Internet testbeds, we launched OF@TEIN+ in 2017, to further extend and expand OF@TEIN collaboration [1]. Fig 1 shows, OF@TEIN+ multisite cloud (denoted as OF@TEIN+ Playground) connects around 10 international sites in 10 countries (Korea, Malaysia, Thailand, Indonesia, Laos, Cambodia, Vietnam, Myanmar, Bhutan, and India) and interconnected via OF@TEIN+ network. In the OF@TEIN+ project, multi-site affordable Playground sites are established which consists of hyper-converged box-style resources named (Type O) “SmartX Micro-Box” having interfaces for management/control and data and IoT devices. SmartX Micro-Box is configured to support Cloud-native computing with containerized IoT-SDN-Cloud functionalities and capabilities of edge computing that allow computation to be performed at the network edge near data sources [2].

OF@TEIN+ Playground as shown in Fig. 1 supports multiple resource types: physical, virtual, and container types [3]. In the multisite cloud, infrastructure burdens of large volume generated by end-users or devices and transferred to a centralized cloud lead to inefficient utilization of bandwidth, storage and computing resources [4]. In order to effectively operate OF@TEIN+ Playground, it is truly requisite to recognize how resources (physical/virtual/container) are running over a time period. Furthermore, maintaining the knowledge of networking connection, through monitoring and visualization is essential to understand and troubleshoot server and network issues before they affect the end-users. Lately, we have been developing tools for monitoring and measurement of OF@TEIN+ Playground resources enabled with cloud-native edge computing that delivers a rich understanding of operations through visibility data and provides

---

<sup>+</sup> Corresponding author. Tel.: +82-62-715-2219  
E-mail address: jongwon@gist.ac.kr

deep insights into current operations, without worrying about underlying infrastructure. By applying persistent visibility to OF@TEIN+ Playground, leveraging modified ‘SmartX Multi-View Visibility Framework (MVF)’ we can monitor various dynamic visibility metrics from multiple measurement points across both physical and containerized resources and associated flows of the playground [5]. However, “persistent operations” deal effectively with managing playground resources leveraging visibility data as well as enabling end-users to utilize playground applications in a stable and reliable way. In addition, visualization of playground operations enables the operators to make informed decisions to tackle the above-mentioned challenges.

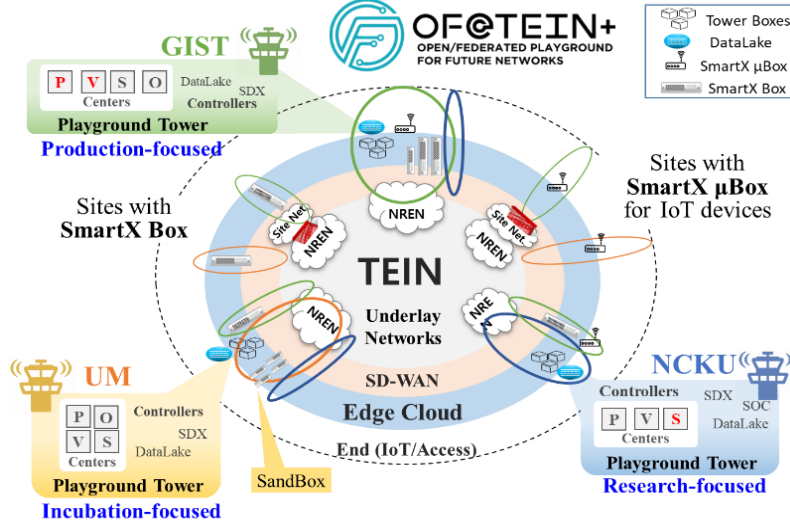


Fig. 1: OF@TEIN+ Playground as multisite clouds.

Thus in this research, leveraging multi-layer visibility data and supporting efficient usage of playground applications, we propose a solution for persistent operations of playground with visualization support for verification. In summary, the key contributions of this paper are:

- We formulate the requirements for maintaining visibility data together with utilizing the cloud-native applications.
- Present design and implementation for maintaining resilient and robust collection of visibility data, followed by integration (i.e. prepared and inferred) at a centralized place.
- Present design and implementation for employing cloud-native containerized applications, distributed across playground.
- Verify persistent operations of playground with visualization schemes such as multi-belt onion-ring visualization.

The remainder of this paper is organized as follows. In Section II, we briefly discuss requirements. In Section III, we collectively cover both design and preliminary prototype implementation of achieving persistent operations. After that, we discuss verification results in Section IV. Finally, section V concludes the paper.

## 2. Background and Related Work

### 2.1. SmartX Multi-View Visibility for Persistent Operation

In OF@TEIN+ playground persistent monitoring/ measurement leveraging SmartX MVF is proposed to deal with the multiple layers of visibility and visualization such as resource layer (underlay, physical), flow-layer, and workload-layer [6]. In resource-layer visibility, monitoring and visualization of playground physical resources and inter-connects (e.g. paths and links) are considered. Whereas flow-layer visibility monitors overlay network traffic in near real-time through packet tracing. A network flow is typically a sequence of network packets that belongs to a certain network sessions between two endpoints. Flow layer visibility deals with different levels of flow information (i.e., collected, clustered, identified and un-clustered

flow) by utilizing a balanced flow collection, clustering, and tagging. Next, workload-layer visibility is responsible for monitoring and visualization of inter-connected containerized functions (e.g. Web, App, DB).

## 2.2. Limitations of Existing Tools

A number of open-source visibility monitoring and visualizations tools already exists such as Vendetta [6] for monitoring distributed testbed, monitoring of OpenFlow-based SDN [7] and Monarch based on Software-defined infrastructure [8]. However, these tools do not directly cater to persistent operation and intuitive visualization for single-box-virtualized deployment style of OF@TEIN+ playground. In order to solve this limitation, we propose a unified persistent monitoring solution to monitor entire infrastructure including heterogeneous resources (e.g. physical, virtual, and container), inter-connects and applications (infrastructure). The initial design of SmarX MVF applies network-packet precise collection at Micro-Box, and then this raw collection is transferred at the Visibility center to perform the processing for flows creation. This process induces burden on storage capacity and network bandwidth. To handle these issues, we introduced the process of data aggregation, tagging to crate flows at the Micro-box instead of doing the same at visibility center.

## 2.3. Requirements of Persistence Operations for OF@TEIN+ Playground

Maintaining persistent visibility in cloud-native infrastructure poses several challenges such as Micro-Boxes may join and leave the network (e.g. connection failure). Multi-view visibility is required to understand downtime, and manage physical assets. Secondly, job scheduling, resource provisioning, and allocation mechanism require monitoring metrics with near real-time collection and low latency. However, in a graphically distributed infrastructure edge Boxes could induce delay. This creates a need for keeping the monitoring data at the edge and keeping the resource collection at a reasonable volume. Thirdly, in edge devices, for reliable transfer of visibility data at the Visibility Center, a local/temporary storage is required during disconnections, unexpected applications closure. Lastly, for faster troubleshooting, visualization must flexibly incorporate multi-layer visibility data, collected from multiple sites with multi-tenancy support.

## 3. Persistence Operations of OF@TEIN+ Playground: Design and Porotype Implementation

In this section, we present design and preliminary prototype implementation of proposed persistent operation.

### 3.1. Selected Tools for Persistent Operation

The persistent operations of visibility collection require the identification of key monitoring tools that sustains major impact on operational states of the playground . For resource layer visibility collection we utilize *Collectd*<sup>1</sup> to collect performance statistics of Micro-Box. For flow-layer visibility collection, we use *eBPF*-based packet tracing tools. While *Apache-Spark*<sup>2</sup> with Scala is utilized to generate flows from packet tracing. Besides, for resource-layer visibility we employed *PerfSONAR*<sup>3</sup> command-line tools for monitoring playground resources, interconnects and end-to-end network performance metrics. To ensure consistent running of monitoring applications we placed Box-agents at each resource that periodically check the running status and actively start the application within a short interval. Besides, Box-Agents communicate with Centre-Agent through *zeroMQ*-based communication to handle asynchronous communication. At the visibility Center together with *Apache Kafka*<sup>4</sup>, to manage reliable and persistent transport of visibility data, *Apache ZooKeeper* provides automatic management of metadata and synchronization issues. A customized java-based tool parse and validate the visibility data before storing it in the appropriate databases (i.e., *MongoDB*<sup>5</sup>, *ElasticSearch*<sup>6</sup> and *InfluxDB*<sup>7</sup>). A scheduler aggregates values of measurements at end of the

<sup>1</sup> collectd collects metrics from a number of sources, e.g. the operating system, applications and stores this information over the network

<sup>2</sup> Apache Spark is an open-source fast and general engine for large-scale data processing.

<sup>3</sup> perfSONAR is a test and measurement infrastructure that is used by science networks around the world to monitor and ensure network performance.

<sup>4</sup> Apache Kafka is an open-source distributed stream processing platform for building real-time data pipelines and streaming apps written in Java etc.

<sup>5</sup> MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling.

<sup>6</sup> Elasticsearch is an open-source, multi-tenant, distributed, search engine for cloud environment with various set of APIs.

<sup>7</sup> InfluxDB is an open-source time series database useful for recording metrics, and performing analytics with built-in HTTP API support.

day to be useful for analysis purposes. For single view unified visualization, we apply multi-belt onion-ring visualization together with Kibana and Grafana. While, for orchestration of containerized application, we employ Kubernetes.

### 3.2. Design and Porotype Implementation

We leverage distributed multi-site SmartX Micro-Box supporting cloud-native (containerized) IoT-Gateway and prepared as Kubernetes-orchestrated workers with SDN-coordinated special connectivity. At the Micro-Box the *persistent visibility collection and validation* stage collects and validates monitoring and measurement data based on selected visibility metrics. Formatted visibility data is sent to *Persistent Visibility Storage* and *Staging* stage where data is stored. Next *Persistent Visibility Integration* stage integrates collected data for generating consolidated reports over a period and identifies any anomalies. Finally, *persistent Visualization* stage accesses processed data and auto-generates graphical views. As shown in Fig 2, we deploy SmartX-Micro-Box at multiple Sites in OF@TEIN+ Playground with capabilities of sending control/data messages at the Visibility Center leveraging multiple tools. For utilization of playground operations by end-user, we developed a cloud-native service through Kubernetes and distributed it through OF@TEIN+ playground. One Kubernetes cluster was formed with Edge IoT-gateways distributed at Multi-site of OF@TEIN+ playground.

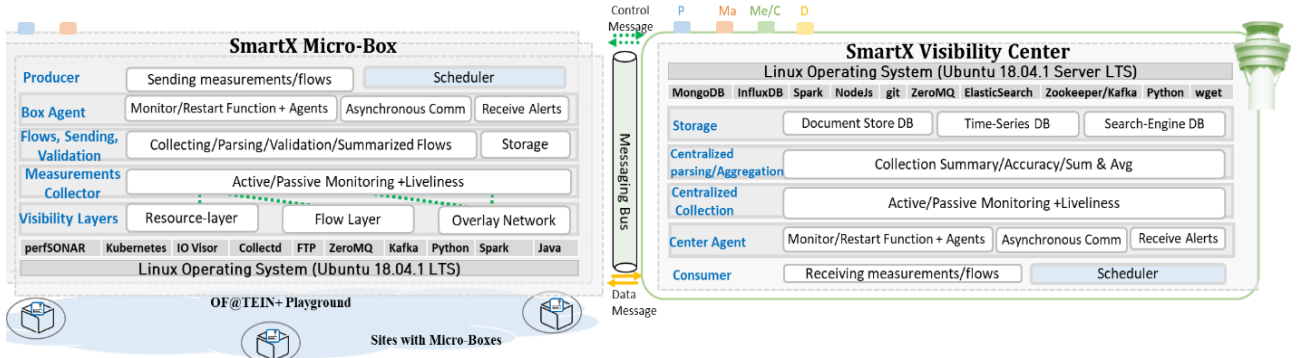


Fig. 2: Design of persistent visualization for SmartX Multi-View Visibility.

## 4. Persistent Operation of OF@TEIN+ Playground: Verification

### 4.1. Persistent Visibility Operation

For verifying the prototype implementation, we utilize the OF@TEIN+ playground. We consider metrics from visibility tools and visualized visibility data at each step of visibility workflow.

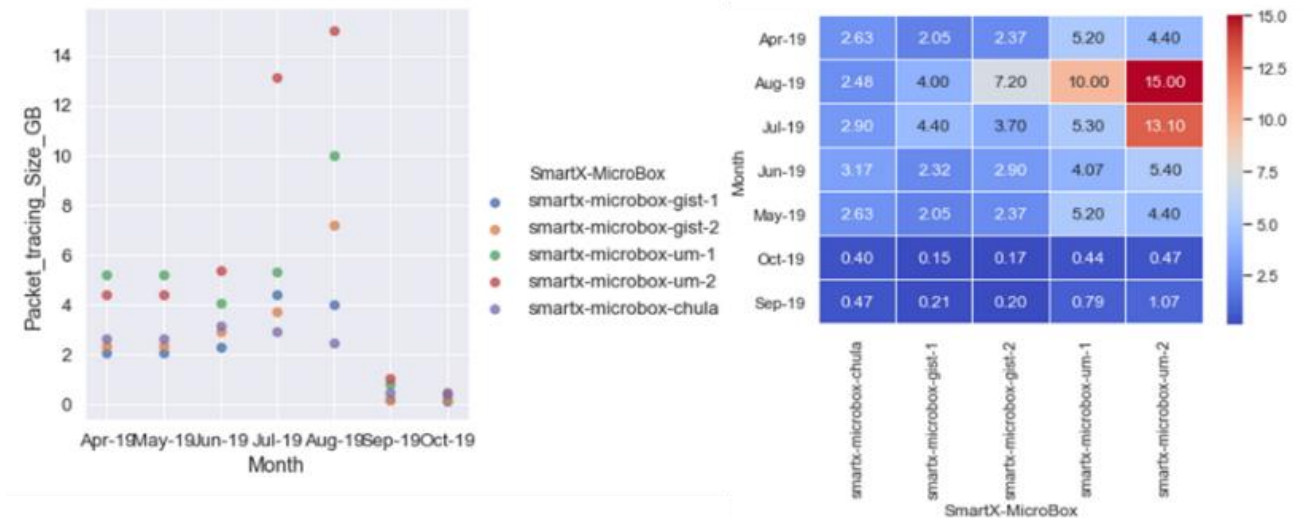


Fig. 3: Visualization for generated volume during packet tracing

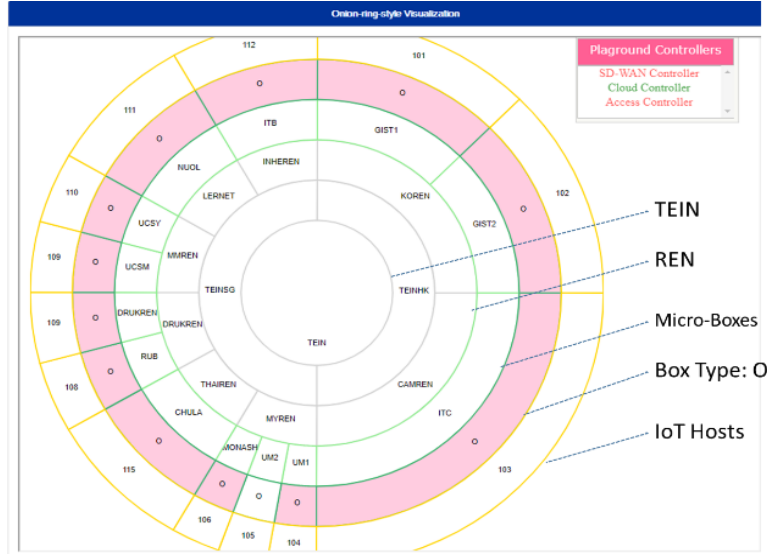


Fig. 4: Onion-ring visualization of OF@TEIN+ Playground (top), Grafana-based visualization of Resource Layer (down)

Figure 3 shows verification for reduction in collection size from previous months. In the last two months (i.e. Sept-Oct 19) where flows are generated at the Micro-box, size of visibility data was reduced by over 80% comparing with previous months (i.e. Apr-Aug 19). Fig. 4 (top) shows a multi-belt onion-ring visualization, which is an effective dashboard to present underlay resource-layer, physical resource-layer, and flow-layer visibilities together in a single unified view. Onion-ring supports interactive visualization, i.e. clicking on layer labeled “Micro-Box”, opens a sub-window that shows visualization for Physical resource layer with multiple metrics leveraging Grafana and views its relationship with the underlay networks as shown in Fig. 4 (down).

## 4.2. Cloud-native Service Operation

For verification of operations usage, we develop applications on a cloud-native platform i.e. Kubernetes to facilitate in terms of automatic deployment, scaling, and management of distributed cloud-native applications. These applications support a service that combines IoT and cloud using SmartX framework. Thus, cloud-native service operation is verified through Smart energy service, as SmartX IoT-Cloud Service. Persistent operation needs to understand the current state of the service functions and make sure they are working well in isolation without conflicting with other containers. In this paper, we check the status of service containers deployed on OF@TEIN+ Playground using Weave scope, an open source visualization tool, to verify that services are normally deployed and running. As shown in Figure 5, we verified that smart energy IoT-Cloud service is well running by using the cloud-native Kubernetes cluster deployed in multi-site OF@TEIN+ Playground by monitoring the operation status of the containers that make up the service.



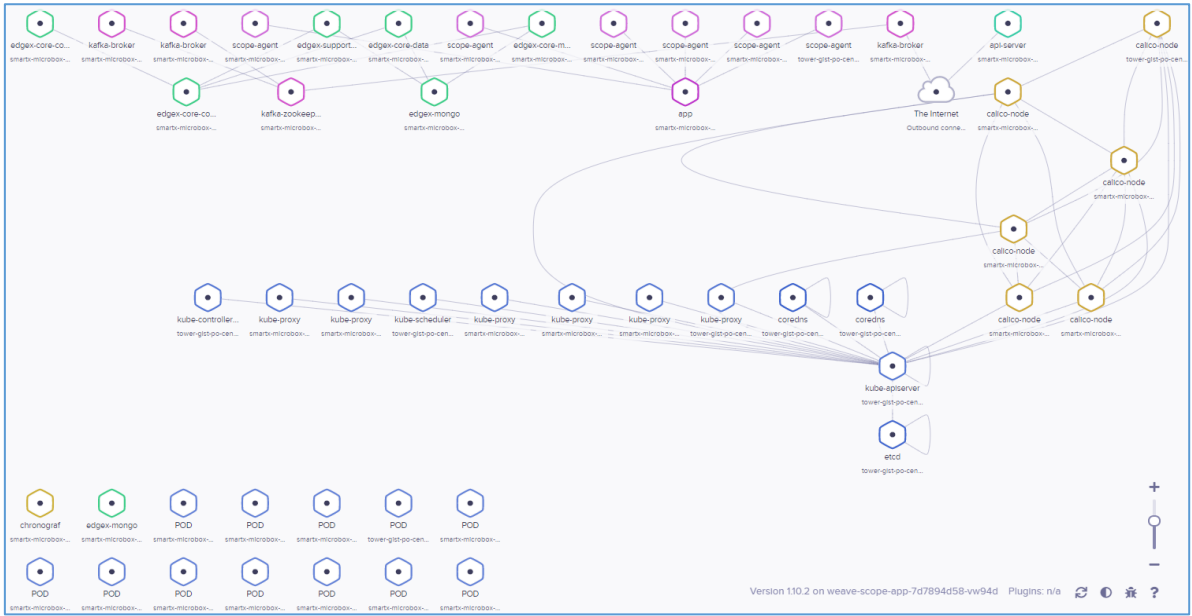


Fig. 5: Smart energy service containers deployment in OF-TEIN+ playground

## 5. Conclusion

In this paper, we presented our initial effort to provide persistent operations for OF@TEIN+ playground developers and operators to effectively operate and maintain the playground. We verified the work by persistently visualizing multiple layers of visibilities leveraging SmartX Multi-view visibility.

## 6. Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00421, Cyber Security Defense Cycle Mechanism for New Security Threats). This work is also partially supported by the Data-centric IoT-cloud service platform for smart communities (IoTcloudServe@TEIN) project under the WP4 Future Internet of Asi@Connect.

## 7. References

- [1] Kim J, Cha B, Kim J, et al. OF@ TEIN: An OpenFlow-enabled SDN testbed over international SmartX Rack sites. *Proceedings of the Asia-Pacific Advanced Network* 2013; 36: 17–22.
- [2] Shi, Weisong, and Schahram Dustdar. "The promise of edge computing." *Computer* 49.5 (2016): 78-81.
- [3] Usman, Muhammad, Nguyen Tien Manh, and JongWon Kim. "Multi-belt Onion-ring Visualization of OF@ TEIN Testbed for SmartX Multi-View Visibility."
- [4] Usman, Muhammad, et al. "SmartX multiview visibility framework leveraging open-source software for SDN-cloud playground." 2017 IEEE Conference on Network Softwarization (NetSoft). IEEE, 2017.
- [5] M Ahmad Rathore, M Usman, JW Kim. "Maintaining SmartX Multi-View Visibility for OF@TEIN+ Distributed Cloud-native Edge Boxes." *Transactions on Emerging Telecommunications Technologies* (2019): Submitted.
- [6] Rensfelt, Olof, Lars-Ake Larzon, and Sven Westergren. "Vendetta-a tool for flexible monitoring and management of distributed testbeds." 2007 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities. IEEE, 2007.
- [7] Isolani, Pedro Heleno, et al. "Interactive monitoring, visualization, and configuration of OpenFlow-based SDN." 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015.
- [8] Lin, Jieyu, et al. "Monitoring and measurement in software-defined infrastructure." 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015.