Cloud-native 5G calls for data sources with common standards for use in automated assurance

May 2020

Anil Rao

Webscale companies such as Amazon, Facebook, Google and Microsoft are leading the way in software innovation and are creating some of the major trends that are being embraced by enterprises and communications service providers (CSPs). Chief among them are the adoption of open-source technologies for software engineering and the move to an analytics-driven platform approach to infrastructure and service operations.

Open-source software is becoming mainstream in telecoms networks and operations as CSPs embark on digital transformations

CSPs have a history of deploying open-source software such as Linux within their IT functions. However, open-source software is expected to play a more-central role as CSPs embark on digital transformation initiatives around network virtualisation, containerisation and operations automation in order to increase service agility and transform opex economics.

The Linux Networking Foundation is leading a plethora of industry initiatives such as ONAP, FD.io, OPNFV, Open Daylight and Tungsten Fabric, and is also collaborating with other industry initiatives such as the Cloud-Native Computing Foundation (CNCF), ETSI, LF Edge, ONF and O-RAN Alliance to drive the adoption of open-source software natively within networks (as container network functions (CNFs)) and network automation platforms.

The telecoms industry is coalescing around the Kubernetes ecosystem

Cloud-native architecture is a requirement not only for 5G networks, but also for the applications deployed on top of those networks. CSPs need to fully understand this architecture, which is characterised by containerised applications composed of microservices. Microservices-based applications are deployed and orchestrated by Kubernetes (K8s), an abstraction layer that permits the portability of containers across different infrastructure and facilitates an unprecedented level of service scalability and automation.

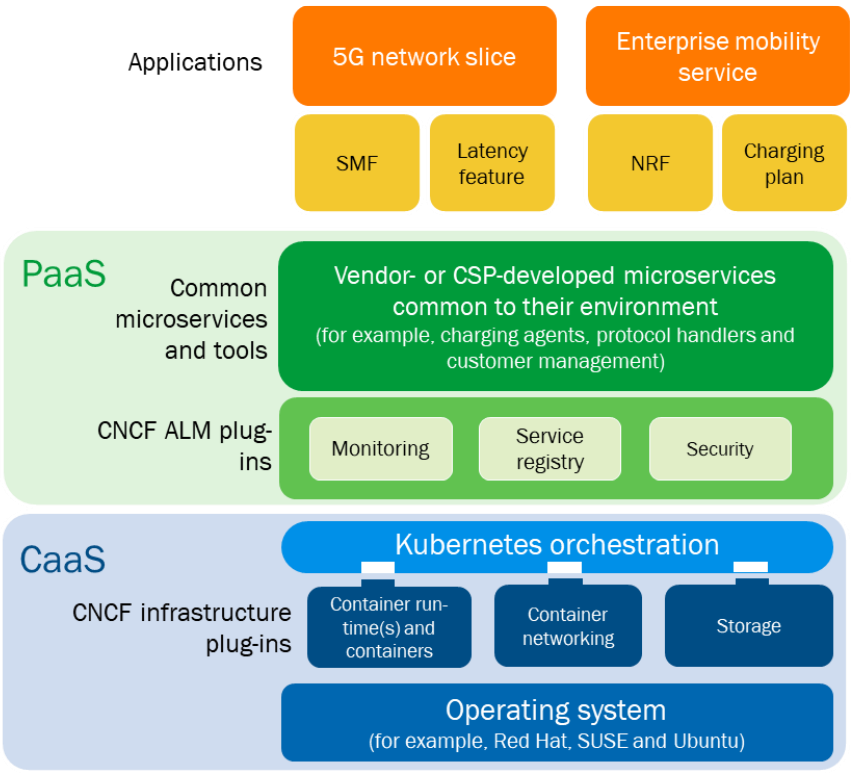
The software stack is formed from the container management layer (CaaS) and a pool of reusable, common components that developers can access when building applications (PaaS). These microservices can be classified into three following categories (Figure 1).

- **Generic or non-functional microservices such as CNCF.** Application lifecycle management plug-ins are ideally implemented using off-the-shelf components, and include the monitoring data for the underlying K8s layer, including providing telemetry and platform health metrics.

- **Common microservices and tools.** These support the higher-layer business logic but are not differentiators.

- **Application code.** This sits at the top of the software stack and, along with the business-specific microservices, is the key differentiator for the business, making it the most-valuable part of the stack.

Figure 1: Kubernetes-based cloud-native technology stack



Source: Analysys Mason, 2020

Automated assurance solutions must ingest K8s monitoring data to provide granular visibility and troubleshooting

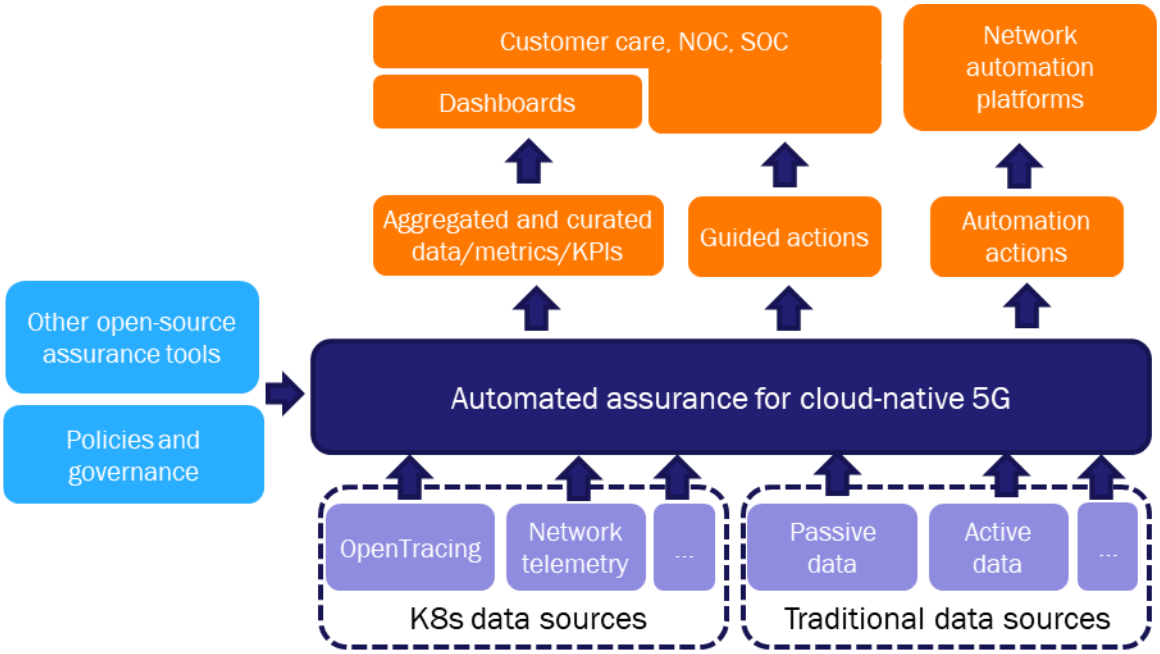
Traditional assurance systems were developed as closed, full-stack, purpose-built solutions that use predefined structured data sources such as offline SNMP data (for example, with a 15-minute delay), passive packet data or active test data, with limited or no ability to ingest third-party data sources. Operationalisation and automation of dynamic cloud-native networks calls for new assurance capabilities that can provide granular, real-time performance data with deep infrastructure and network visibility to rapidly respond to (and even pre-empt) performance degradations in the various layers of the networking stack, the service layer and the application.

To achieve this in best of breed cloud-native networks, where the infrastructure, network functions and applications can be sourced from different vendors, assurance systems must:

- provide an independent vendor-agnostic view of the end-to-end network performance
- become an open platform with open southbound and northbound APIs
- be developed and deployed using cloud-native microservices architecture (such as Kubernetes) for seamless integration with the cloud-native network stack.

An open assurance platform enables the easy ingestion of new data sources such as streaming network telemetry and OpenTracing data from the cloud-native stack. Together, these data sources can increase the accuracy of the contextual insights that can be consumed by a plethora of adjunct operational systems (such as network and service orchestration systems) in order to enable the closed-loop lifecycle automation of critical 5G services (Figure 2).

Figure 2: Example of a cloud-native and open automated assurance platform



Source: Analysys Mason, 2020

New data sources must be made fit-for-purpose for cloud-native 5G core monitoring

The OpenTracing project that is being incubated by the CNCF provides vendor-neutral APIs and instrumentation for distributed tracing in Kubernetes environments and could be used in the cloud-native 5G core. However, this technology was not originally designed for instrumentation in the cloud-native 5G core and therefore, some enhancements and adaptations will be required to make it fit-for-purpose and conducive for 5G core assurance. Some examples of these adaptations are as follows.

- **Content formats.** OpenTracing is an API, so the content/payload of the event is not standardised. This means that there is a high likelihood that 5G core vendors may implement it differently, causing non-standard data sets with missing fields to be sent to upstream assurance systems.

- **Encoding format.** OpenTracing is widely implemented using ASCII encoding techniques, while streaming network telemetry interfaces use protocol buffers or JSON. The encoding formats need to coalesce around a common approach (for example, protocol buffers) in order to provide an efficient and common data stream format for upstream assurance systems. The merging of OpenTracing and OpenCensus to create the OpenTelemetry project may pave the way for standardisation.

- **Streaming sampled versus all events data.** The OpenTracing implementation may need to offer flexibility in terms of what data is sent to upstream assurance applications (that is, whether sampled data or all events data is used). This flexibility is likely to impact the performance of the core infrastructure, so the right technology choices will need to be made in terms of encoding techniques.

These issues highlight the need for the standardisation of open-source APIs such as OpenTracing to make it easier for the vendors that are developing 5G and assurance products and solutions. Some of the standards definition bodies (for example, TMF Open APIs and Open Data Alliance) must introduce interface and event stream format definitions for the OpenTracing APIs.