Data Collection, Analytics and Events (DCAE)

1.1 (R1)

Dependencies

https://docs.openstack.org/designate/pike/index.html

TBD
https://onap.readthedocs.io/en/latest/submodules/dcaegen2.git/docs/sections/blueprints/DockerHost.html
https://onap.readthedocs.io/en/latest/submodules/dcaegen2.git/docs/sections/blueprints/deploymenthandler.html
https://onap.readthedocs.io/en/latest/submodules/dcaegen2.git/docs/sections/blueprints/centos_vm.html
https://onap.readthedocs.io/en/latest/submodules/dcaegen2.git/docs/sections/blueprints/ves.html

filled
https://onap.readthedocs.io/en/latest/submodules/ccsdk/platform/plugins.git/docs/dnsdesig.html?highlight=designate

http://onap.readthedocs.io/en/latest/submodules/dcaegen2.git/docs/sections/installation.html

https://gerrit.onap.org/r/gitweb?p=dcaegen2/deployments.git;a=blob;f=bootstrap/README-docker.md;h=3fa1b7d4d76b47cb1f1db1eeb2f7424877524e66;hb=refs/heads/master

1.0.0 (Feb 2017)

The Data Collection, Analytics, and Events (DCAE) subsystem, in conjunction with other ONAP components, gathers performance, usage, and configuration data from the managed environment. This data is then fed to various analytic applications, and if anomalies or significant events are detected, the results trigger appropriate actions, such as publishing to other ONAP components such as Policy, MSO, or Controllers.

The primary functions of the DCAE subsystem are to

- Collect, ingest, transform and store data as necessary for analysis
- Provide a framework for development of analytics

These functions enable closed-loop responses by various ONAP components to events or other conditions in the network.

DCAE provides the ability to detect anomalous conditions in the network. Such conditions, might be, for example, fault conditions that need healing or capacity conditions that require resource scaling. DCAE gathers performance, usage, and configuration data about the managed environment, such as about virtual network functions and their underlying infrastructure. This data is then distributed to various analytic micro-services, and if anomalies or significant events are detected, the results trigger appropriate actions. In addition, the micro-services might persist the data (or some transformations of the data) in the storage lake. In addition to supporting closed-loop control, DCAE also makes the data and events available for higher-level correlation by business and operations activities, including business support systems (BSS) and operational support systems (OSS).

Usage and other event processing applications can be created in the DCAE environment. In addition to real-time processing of events, these applications can perform mediation of the usage and other events to external BSSs or OSSs. For example, events about bill-impacting configuration changes or consumption of any new product or service can be subscribed to by external BSS applications for various purposes such as rating, balance management and charge calculations.

The following figure provides a functional view of the DCAE Platform architecture.

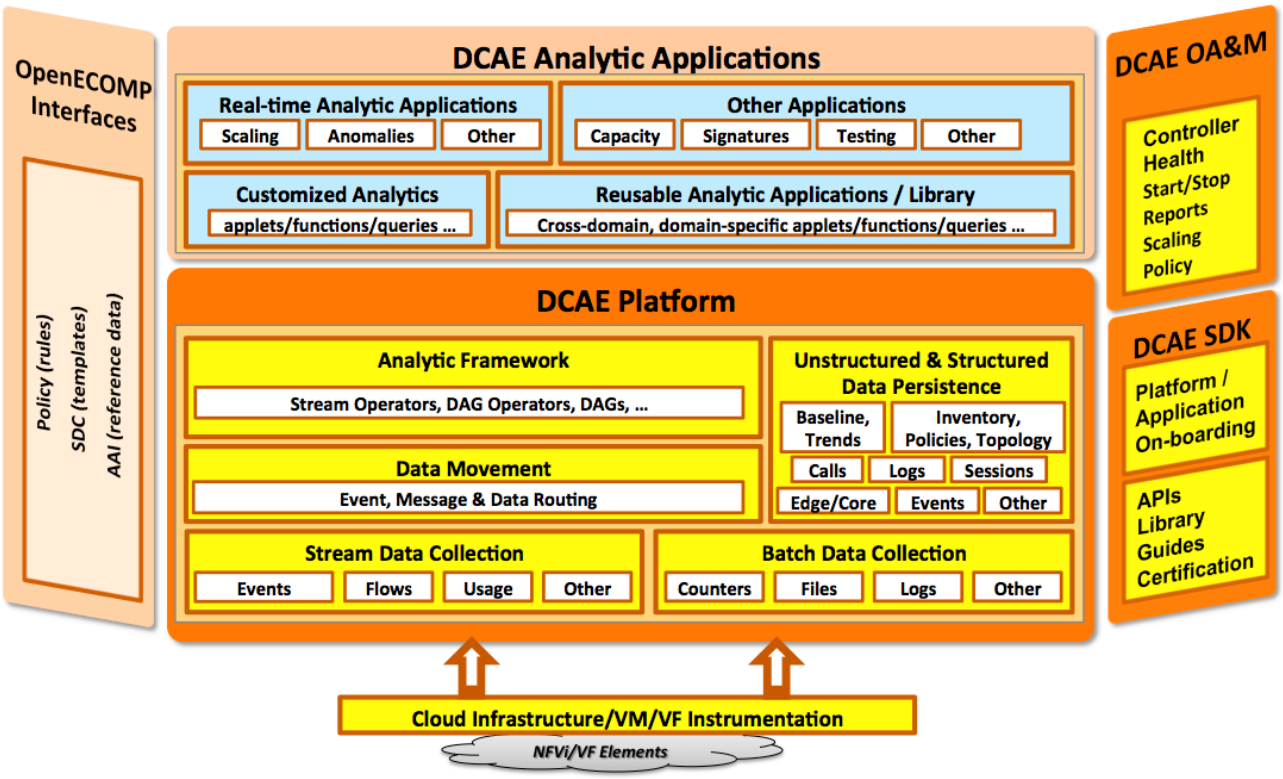

Figure 1. DCAE Platform high-level architecture

DCAE Platform Components

The DCAE Platform consists of several functional components: Collection Framework, Data Movement, Storage Lakes, Analytic Framework, and Analytic Applications.

In large scale deployments, DCAE components are generally distributed in multiple sites that are organized hierarchically. For example, to provide DCAE function for a large scale ONAP system that covers multiple sites spanning across a large geographical area, there will be edge DCAE sites, central DCAE sites, and so on. Edge sites are physically close to the network functions under collection, for reasons such as processing latency, data transport, and security, but often have limited computing and communications resources. On the other hand, central sites generally have more processing capacity and better connectivity to the rest of the ONAP system. This hierarchical organization offers better flexibility, performance, resilience, and security.

Collection Framework

The collection layer provides the various data collectors that are needed to collect the instrumentation that is available from the cloud infrastructure. Included are both physical and virtual elements. For example, collection of the following types of data is supported:

- events data for monitoring the health of the managed environment
- data to compute the key performance and capacity indicators necessary for elastic management of the resources
- granular data needed for detecting network and service conditions (such as flow, session and call records)

The collection layer supports both real-time streaming and batch collection.

Data Movement

This component (known as DMaaP) facilitates the movement of messages and data between various publishers and interested subscribers that may reside at different sites. While a key component within DCAE, this is also the component that enables data movement between various ONAP components.

Edge and Central Lake

DCAE supports a variety of applications and use cases. These range from real-time applications that have stringent latency requirements to other analytic applications that have a need to process a range of unstructured and structured data. The DCAE storage lake supports these needs and is scalable so that new storage technologies can be incorporated as they become available. The storage lake uses big-data storage technologies such as in-memory repositories and support for raw, structured, unstructured and semi-structured data to accommodate a broad scope of requirements such as large volume, velocity, and variety.

While there may be detailed data retained at the DCAE edge layer for analysis and trouble-shooting, applications should optimize the use of bandwidth and storage resources by propagating only the required data (for example, reduced, transformed, or aggregated) to the core data lake for other analyses.

Analytic Framework

Analytics and related applications run in the Analytic Framework of DCAE. The Analytic Framework enables agile development of analytic applications. This framework supports creation of applications that process data from multiple streams and sources. Applications can be real-time – for example, analytics, anomaly detection, capacity monitoring, congestion monitoring, or alarm correlation – or non-real time, such as applications that perform analytics on previously collected data or forward synthesized, aggregated or transformed data to big data stores and other applications. The framework can process both real-time streams of data and data collected through traditional batch methods. Analytic applications are managed by the DCAE controller.

Analytic Applications

The following list provides examples of the types of applications that can be built on top of DCAE:

Analytics These will be the most common applications that are processing collected data and deriving interesting metrics or analytics for use by other applications. These analytics applications range from very simple ones (from a single source of data) that compute usage, utilization, latency, and similar metrics to very complex ones that detect specific conditions based on data collected from various sources. The analytics could be capacity indicators used to adjust resources or could be performance indicators pointing to anomalous conditions requiring response.

Fault / event correlation: This is a key application type that processes events and thresholds published by managed resources or other applications that detect specific conditions. Based on defined rules, policies, known signatures and other knowledge about the network or service behavior, an application of this kind would determine root cause for various conditions and notify other interested applications.

Performance surveillance and visualization: This class of application provides a window to an operations organization, notifying it of network and service conditions. The notifications could include outages and impacted services or customers based on various dimensions of interest. They provide visual aids ranging from geographic dashboards to virtual information model browsers to detailed drilldown to specific service or customer impacts.

Capacity planning: This class of application provides planners and engineers the ability to adjust forecasts based on observed demands as well as plan specific capacity augments at various levels, e.g., NFVI level (technical plant, racks, clusters, etc.), Network level (bandwidth, circuits, etc.), Service or Customer levels.

Testing and troubleshooting: This class of application provides operations the tools to test & trouble-shoot specific conditions. They could range from simple health checks for testing purposes, to complex service emulations orchestrated for troubleshooting purposes. In both cases, DCAE provides the ability to collect the results of health checks and tests that are conducted. These checks and tests could be done on an ongoing basis, scheduled or conducted on demand.

Security: Some components of the infrastructure may expose new targets for security threats. Orchestration and control, decoupled hardware and software, and commodity hardware may be more susceptible to attack than proprietary hardware. However, SDN and virtual networks also offer an opportunity for collecting a rich set of data for security analytics applications to detect anomalies that signal a security threat, such as DDoS attack, and automatically trigger mitigating action.

Other: The applications listed here are by no means exhaustive and the open architecture of DCAE lends itself to integration of additional application capabilities over time.

DCAE System Flows

The following figures show the implemented system architecture and flows for the first release of ONAP. DCAE for this release is "minimalistic" in the sense that it is a single DCAE site with all DCAE functions.

Figure 2 shows the DCAE configuration flow. The flow proceeds as follows:

1. The DCAE Controller is instantiated from an ONAP Heat template.
2. The DCAE Controller instantiates the rest of the DCAE components, including both infrastructure and service/application components.
3. The DCAE Controller configures service/application components with static configurations, configuration policies fetched at run-time (for example data processing configurations or alert configurations), and any DMaap topics required for communication.

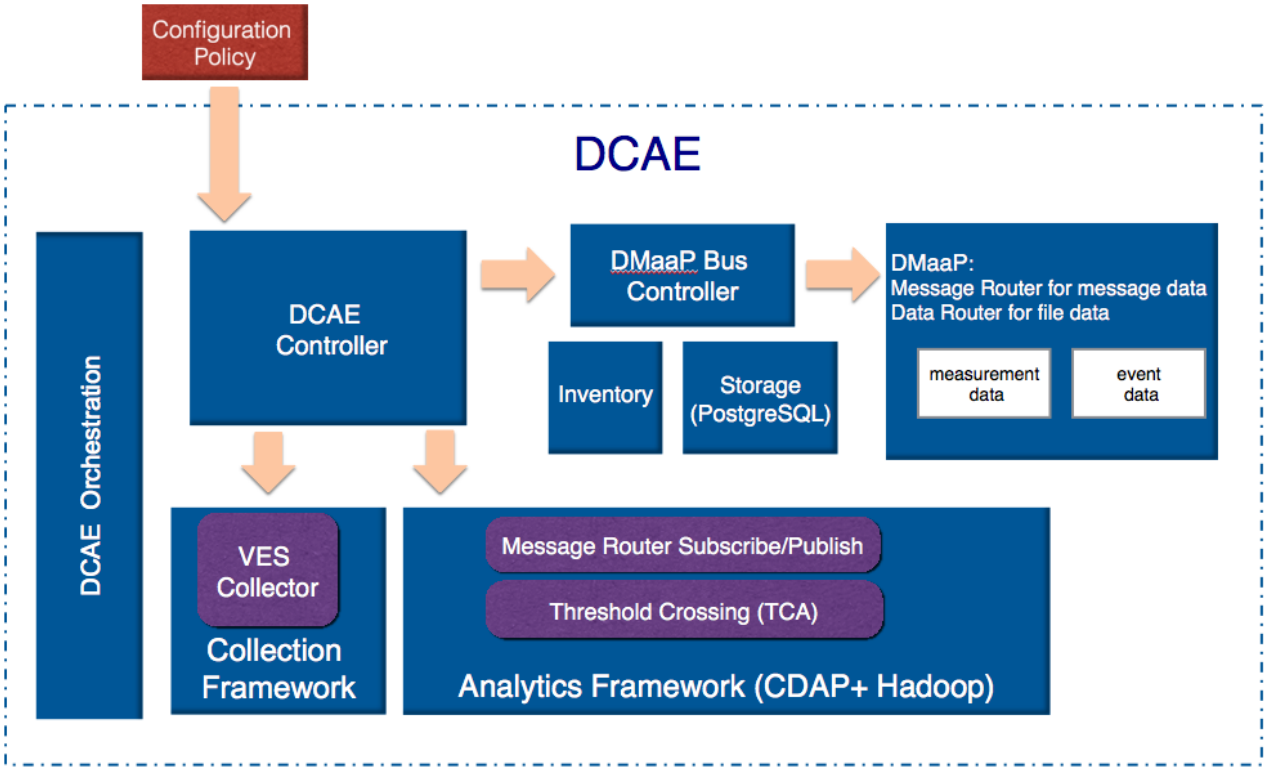

Figure 2. DCAE configuration flow (Control plane)

Figure 3 shows the DCAE data flow. This flow proceeds as follows:

1. VNFs use REST calls to push measurement data to the DCAE VES collector.
2. The VES collector validates, filters, and packages the received measurement data, and publishes the data to the "measurement data" topic of DMaaP.
3. The analytics application receives measurement data from the DMaaP "measurement data" topic.
4. The analytics application analyzes measurement data, and if alert conditions (defined by the alert policy that was installed by the DCAE Controller) are met, publishes an alert event to the DMaaP "event data" topic.
5. Other ONAP components, for example the Policy or MSO subsystems, receive alert events from the DMaaP "event data" topic and react accordingly.

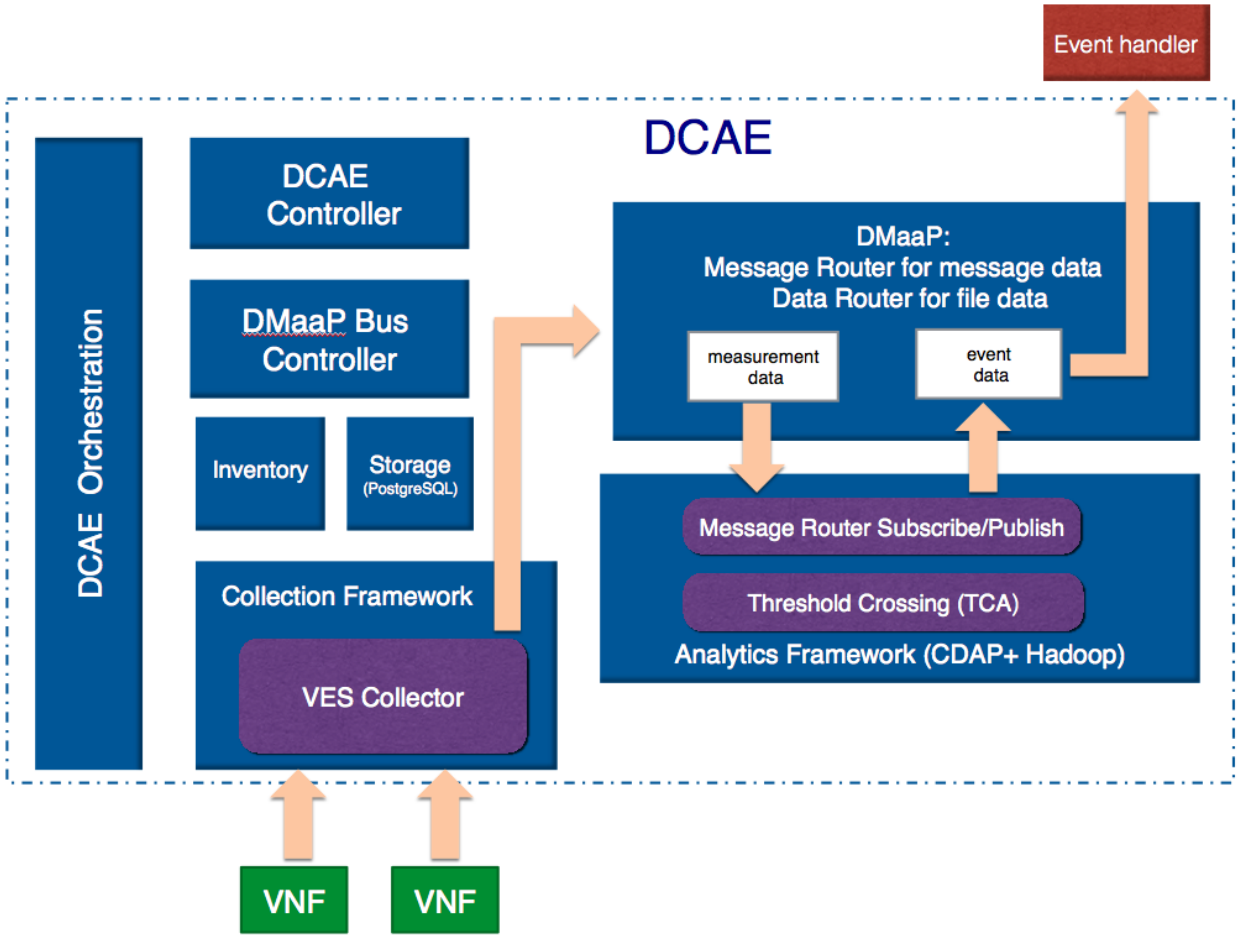

Figure 3. DCAE data flow (Data plane)

References

VES: https://wiki.opnfv.org/display/PROJ/VNF+Event+Stream