LF NETWORKING

LFN Developer & Testing Forum

LF NETWORKING

LFN Developer & Testing Forum

ONAP Architecture Update

Security, Logging and Others

November 18, 2022

Byung-Woo Jun, byung-woo.jun@est.tech, Ericsson

ARCCOM, SECCOM, OOM

Anti-Trust Policy Notice

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrustpolicy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

Security & Logging Mission Statements

Security

- Achieves simpler, uniform and open-source-based security and secure communications
- Handles security at the platform level, not at the application level
- Allows 3rd Party applications to participate by leveraging the platform-level security

Logging

- Supports open-source- and standard-based logging
- Separates log generation from log collection/aggregation/persistence/visualization
  - Handles collection/aggregation/persistence/visualization at the platform level
- Allows logging component stack to be realized by choices of vendors

Why Migrates from AAF to Service Mesh

LFN Developer & Testing Forum

AAF had served its functions with rich features, but…

AAF Shortcomings

- Is a custom solution for ONAP (or pre-ONAP) specific
  - Limited Open-Source community support
  - Complicated user store management
  - Provides language-specific plugin (CADDI)
  - Could be a showstopper for commercial use
- Handles security at the application level
  - Burdens to application development by managing certificates, AuthN/AuthZ by each application
  - No uniform-way of security handling across ONAP
- Experiences security compatibility issues with 3rd Party applications
  - Requires 3rd Party app code modifications to work with AAF
  - Difficult to integrate/federate with operators' security ecosystems
  - Not integrated with Kubernetes
- Is an unmaintained project in ONAP; does not keep up with the latest security evolution
  - Questionable SSO and Multi-factor authentication support
- More details, see AAF and Service Mesh Risk analysis

We want to achieve simpler, uniform, open-source and standard-based Security

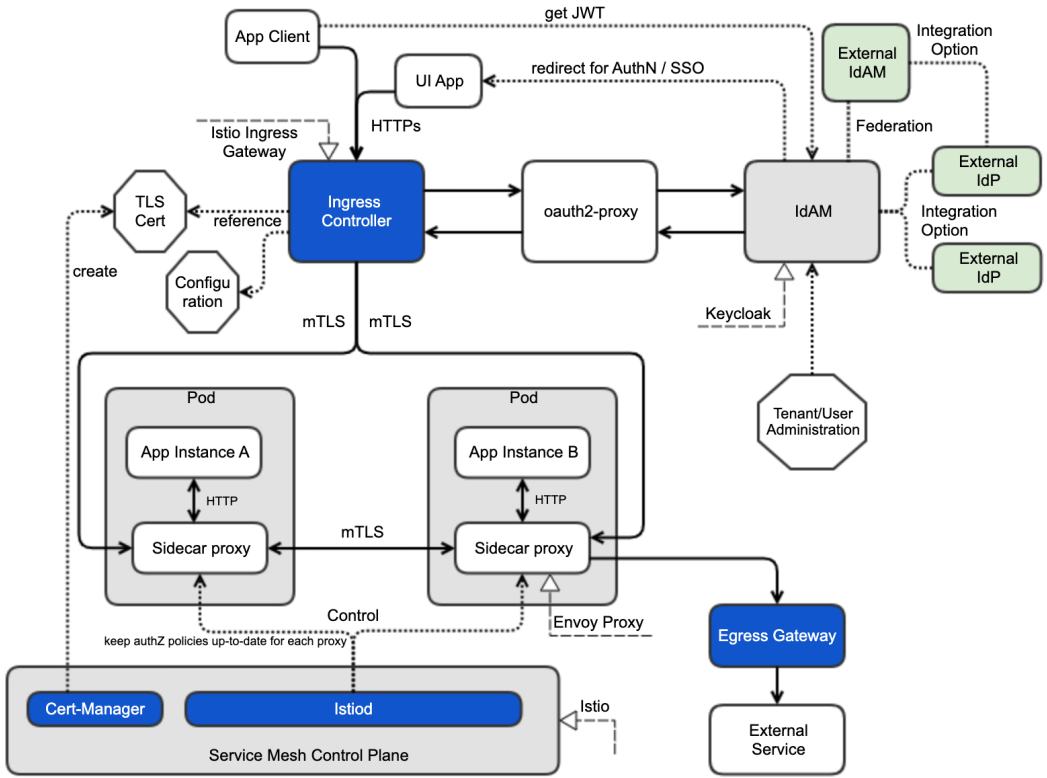
- Handling security at the platform level
- Enabling uniform security across applications
- Minimizing app code changes and increasing integrity, extensibility and customization
- Facilitating 3rd Party application secure communication with configurable AuthN/AuthZ
- Allowing Service Providers/vendors secure integration between ONAP and others
  - e.g., Vendor VNFM/CNFM, NF integration, External Apps
- Service Mesh-pattern Security fulfills ONAP Security requirements

Benefits to ONAP, ORAN and other External communities

- Uniform, open-source- and standard-based security is a foundation for secure integration between ONAP, ORAN and other external communities
- Keeping product impacts for security minimum
- Enabling Applications to focus on their own functionalities, not handling security directly
- Allowing 3rd Party Microservices to participate in ONAP security by configuration

ONAP Security Architecture

LFN Developer & Testing Forum



ONAP uses these Open-Source Security Patterns:

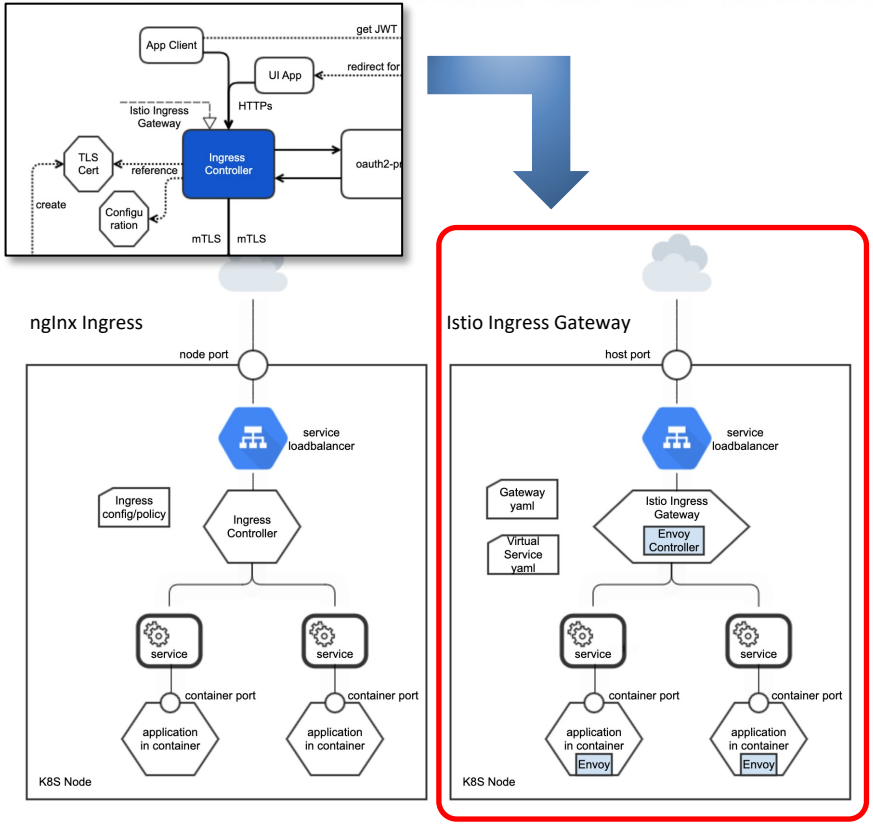
- Istio Ingress Gateway handles incoming external traffic
- oauth2-proxy redirects OAuth2-based requests to IdAM.
- Keycloak is the IdAM for AuthN/AuthZ, SSO, multi-tenancy and user management including RBAC controls
  - For UI Apps, provides redirection for SSO / AuthN / AuthZ
  - For App Clients, provides JWT and AuthN / AuthZ
- Istio acts as the Service Mesh Control Plane
  - Keep policies and authZ policies up-to-date for sidecar proxies
  - Applications are sitting behind sidecar proxies
- Policies and configurations are used for Ingress Controller and the Envoy proxies
- All the component communications are secured by mTLS.
- This Architecture allows Service Providers to integrate their own security ecosystem (external IdAM, External IdP)

It is being implemented – See Service Mesh Update for London

For Architecture details, see [ONAP Next Generation Security & Logging Architecture]

Acronyms
- IdAM = Identity Access Management
- mTLS = mutual Transport Layer Security
- OAuth2 = Open Authorization
- HTTPs = Hypertext Transfer Protocol Secure

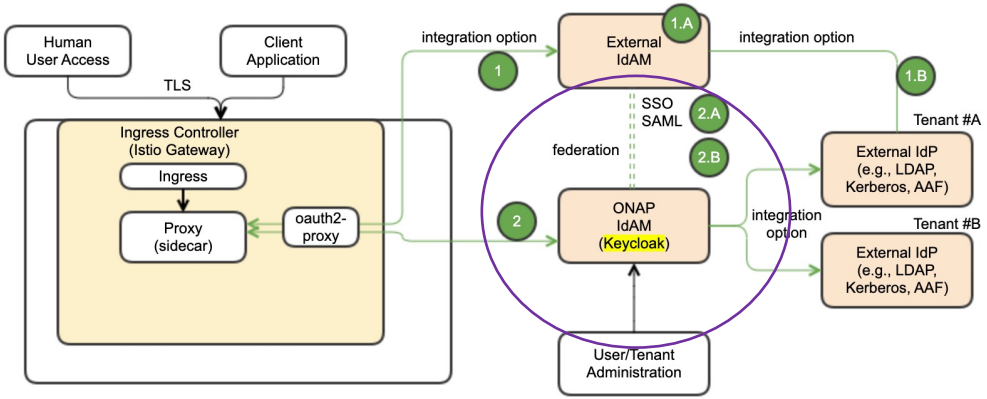
Ingress Controller Realization



- In ONAP, Istio Ingress Gateway is chosen for Ingress, instead of Kubernetes/NgInx Ingress
  - It is a POD with Envoy Controller that does the routing
  - It is configured by Gateway and Virtual Service metadata for intelligent routing, such as rules, load-balancing, traffic rate limiting, policy-based checking, metrics collections
- All external secure communications go thru the Istio Ingress Gateway
  - No Nodeport any longer, with the Istio Ingress Gateway option
  - Applications that are using Kubernetes/NgInx Ingress must migrate to use the Istio Ingress Gateway
- Using oauth2-Proxy, Istio Ingress Gateway interacts with KeyCloak (IdAM) for SSO / AuthN / Authorization
- Once the client request is approved, it routes the request to the Application Service(s)
- For Service Mesh with Ingress setup, see ONAP on Service setup guide

KeyCloak Functions as Reference IdAM

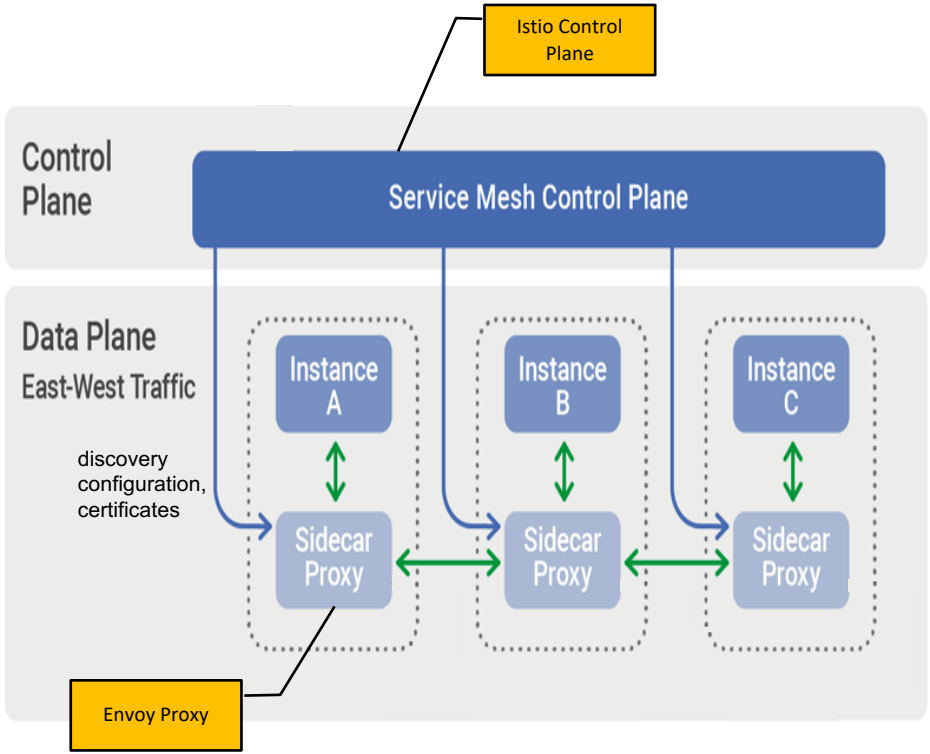
LFN Developer & Testing Forum



#	Choice
1	If the vendor chooses their own IdAM (external IdAM), the AuthN/AuthZ GW should be able to use the external IdAM, instead of ONAP IdAM – full integration choice
	A. The external IdAM will handle AuthN/AuthZ for ONAP incoming traffic; session cookie handling, session handling
	B. The external IdAM will typically use their own IdP (external IdP) to verify their own user identities
2	If ONAP IdAM is used, based on business use cases, the ONAP IdAM could interface with the external IdAM, based on IdAM-to-IdAM interface configuration
	A. SSO
	B. Authentication federation (SAML) – not part of Istanbul

- Is deployed as the ONAP reference IdAM
- Creates public and private key and JWT Token in KeyCloak
- Manages tenant/realm, user, group and roles administration
- Uses its internal IdP, and it can be configured to use external IdP
- Can be replaced with an external IdAM that is compatible with OIDC/OAuth2
- Can be configured to use multiple IdPs for Multi-tenancy
- The oauth2-proxy is used to bridge between ONAP Ingress and KeyCloak for traffic redirect for SSO and AuthN/AuthZ
- For SSO, the login page is NOT part of the application and is configured at KeyCloak
- for more KeyCloak use cases, see ONAP Security & Logging Architecture

Service Mesh Realization in ONAP



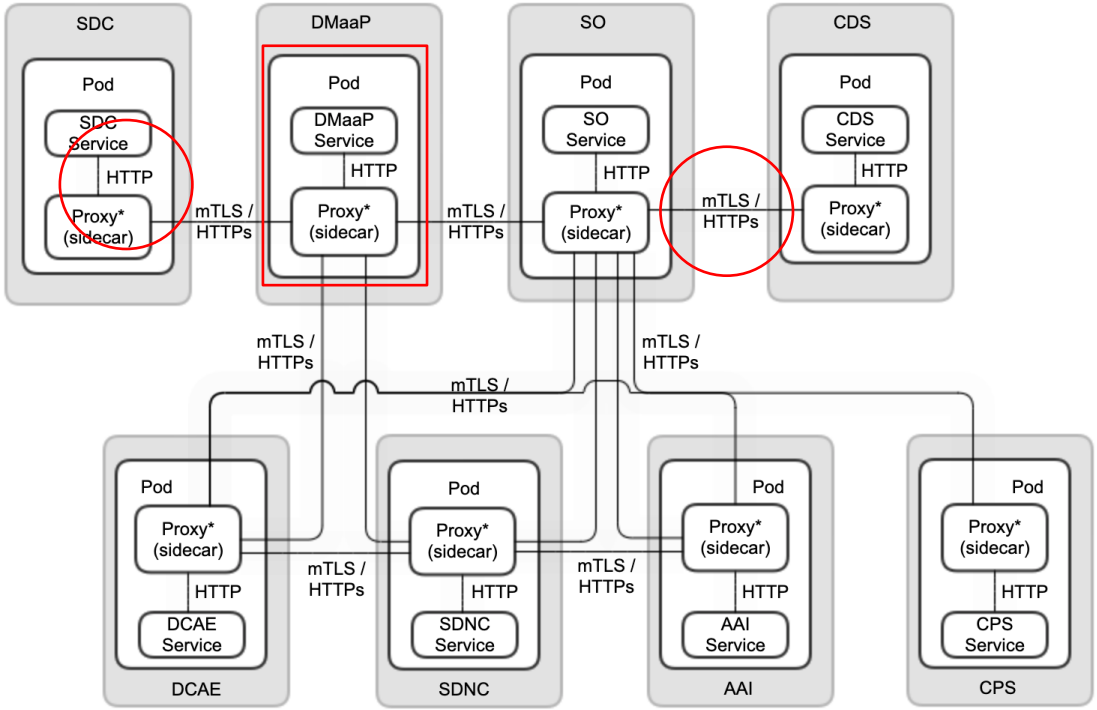
Service Mesh Pattern

- A Service Mesh is a pattern for controlling communications between services by configuration / policies
- It aims to achieve secure communications, authorization, service discovery, load balancing and traffic routing
- It is dedicated and configurable while running at the platform layer instead of the application layer

Istio

- Fulfills ONAP Service-to-Service Security and Communication requirements
- Makes traffic management transparent to the application
- Moves this secure communication out of the application and into the control layer
- Is lightweight and simple to configure regardless of the application size
- Istio Control Plane provides discovery, configuration, certificates and others to all the sidecar proxies.

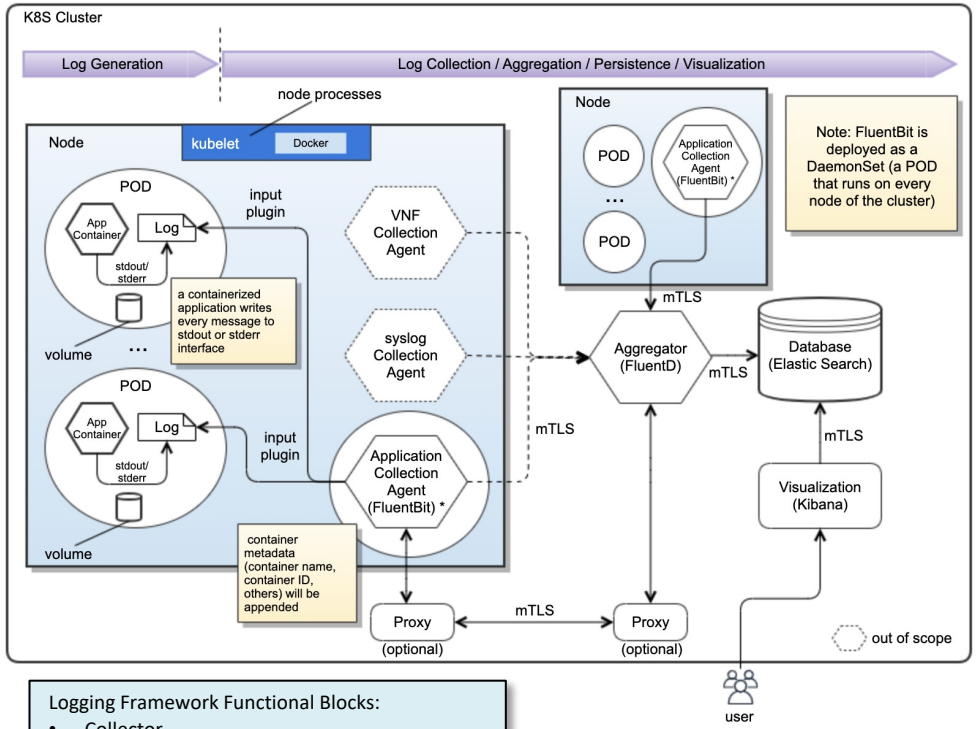
Communication Protocol Between Service & Proxy and Between Proxies



- The communication protocol between Service and Proxy is “HTTP”
- ONAP Application does not handle security directly and it is sitting behind its own proxy (sidecar)
  - For that, Service and its Proxy will sit on the same POD to make their communications internal
- The communication protocol between Proxies (sidecars) is mTLS via “HTTPs”
- ONAP Service-to-Service communications will be handled by using Sidecar Proxies. This can make MSB use optional for those communications

ONAP Logging Framework Architecture

LFN NETWORKING
LFN Developer & Testing Forum



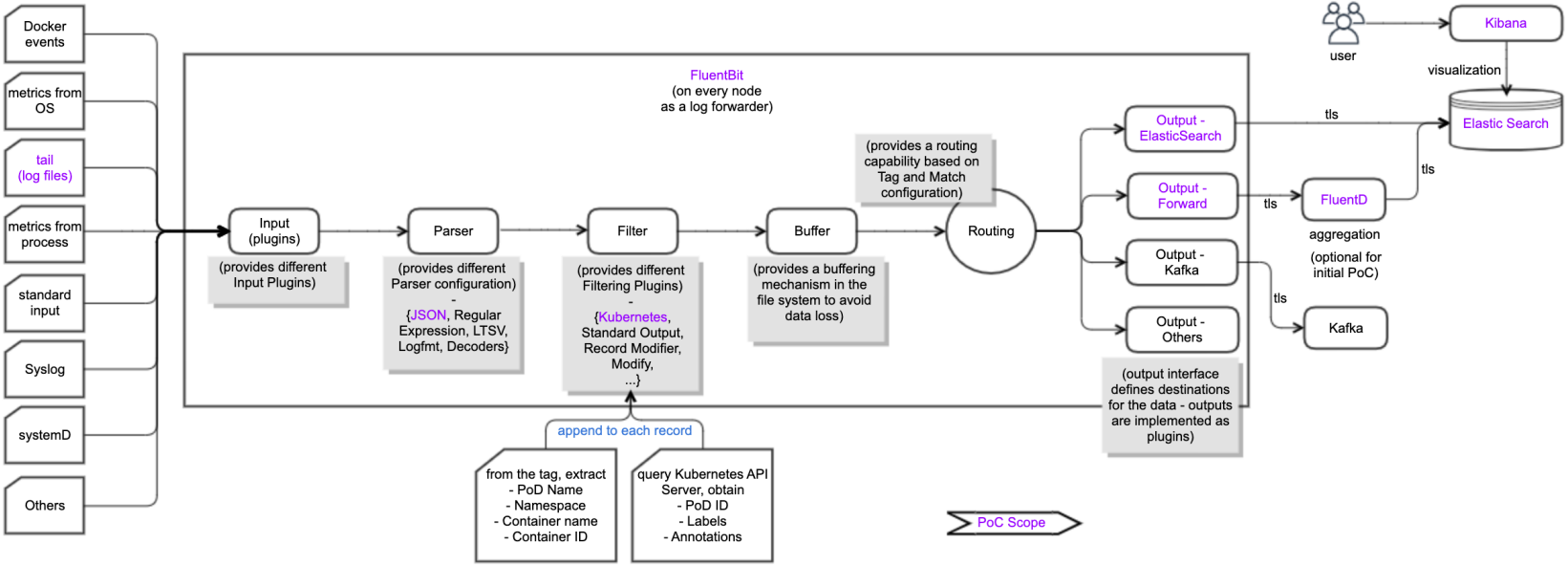
Logging Framework Functional Blocks:

- Collector
- Aggregator
- Database
- Visualization

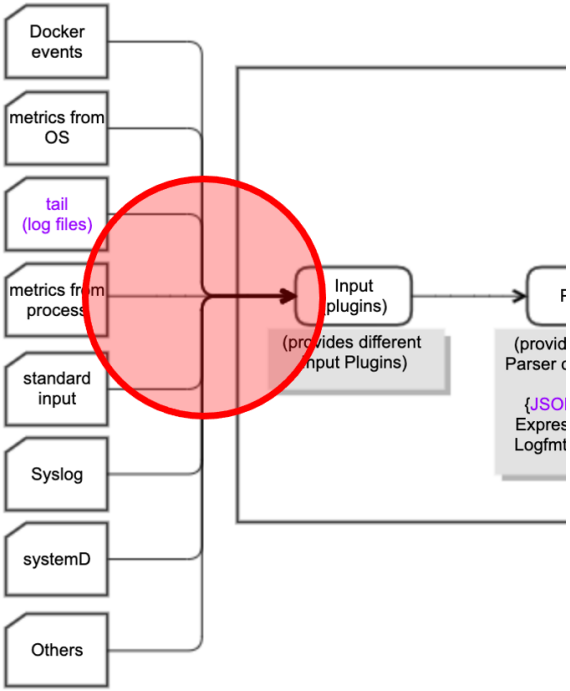
- Supports open-source- and standard-based logging
- Separates log generation from log collection/aggregate/persistence/visualization/analysis
  - ONAP Applications handle log generation only
  - Logging Framework Stack will handle the rest
- Assumes all the ONAP applications push their logs into STDOUT/STDERR
- Provides reference implementation and can be realized by choices of vendors
- Deploys the Log Collector to every node and the Collector pushes log data to the Aggregator/Database
- Conforms to SECCOM Global Requirements for the standardized log format
- Communications between Logging functional blocks will be secure

FluentBit Internal Pipeline for Logging

- For ONAP Logging reference implementation, FluentBit is used as the log collector.
- FluentBit will be deployed to every node to collect log data from each application POD
- Fluentbit will append container metadata to conform to Log Global Requirement
- In ONAP, FluentBit is configured to receive ONAP App log data thru the “tail” FluentBit input plugin



Logging Framework PoC Update



PoC Status:

- Deployed FluentBit as a Daemonset to deploy it to every node in ONAP
- Configured FluentBit to receive ONAP App log data thru the “tail” FluentBit input plugin
- Used FluentBit Kubernetes Filter to add container metadata
- Pushed normalized log output to ElasticSearch thru the FluentBit Output-ElasticSearch plugin
- Visualized log data thru Kibana

Security Issues:

- FluentBit Daemonset with root-access is used to access log files in ONAP App PODs
- In ONAP, root-access users are not allowed for security reasons
- Without root-access, deployed FluentBit cannot access log files in ONAP App PODs
- There was an attempt to use Daemonset SecurityContext, dropping all the capabilities except file access, but this is not a clean solution.

Suggested Solutions:

- Reroute log files to where FluentBit can access
- The rerouting mechanism is under investigation

ONAP Mainstream Architecture Study

LFN NETWORKING
LFN Developer & Testing Forum

ONAP Mainstream Architecture aims to facilitate ONAP adaptation and extensibility for Service Providers / DevOps. The following would be study areas:

- More component/sub-component modularity and independence
- Component interface and behavior normalization / standardization
- Extensible, customizable and substitutional component functions and mechanisms
- Well-balanced common/platform services vs. autonomous services
- Pick & choose and Aggregation of functions
- Unified and Platform-level security and logging across ONAP and SP OAM & Network Resource domains
- ONAP Mainstream Architecture skeleton outline

LF NETWORKING

LFN Developer & Testing Forum