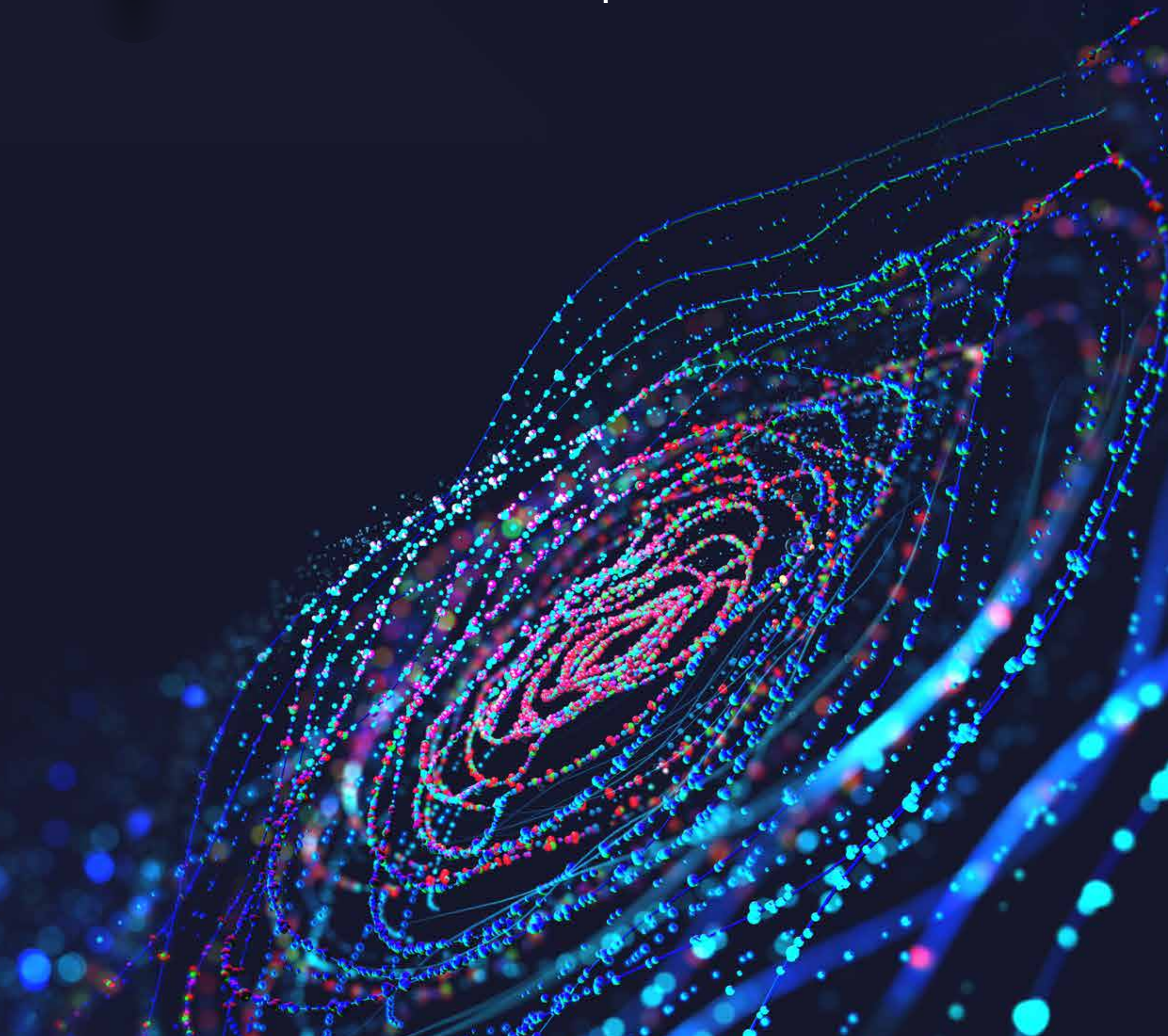
WHITEPAPER

NETWORK OBSERVABILITY

A Tech Mahindra Perspective



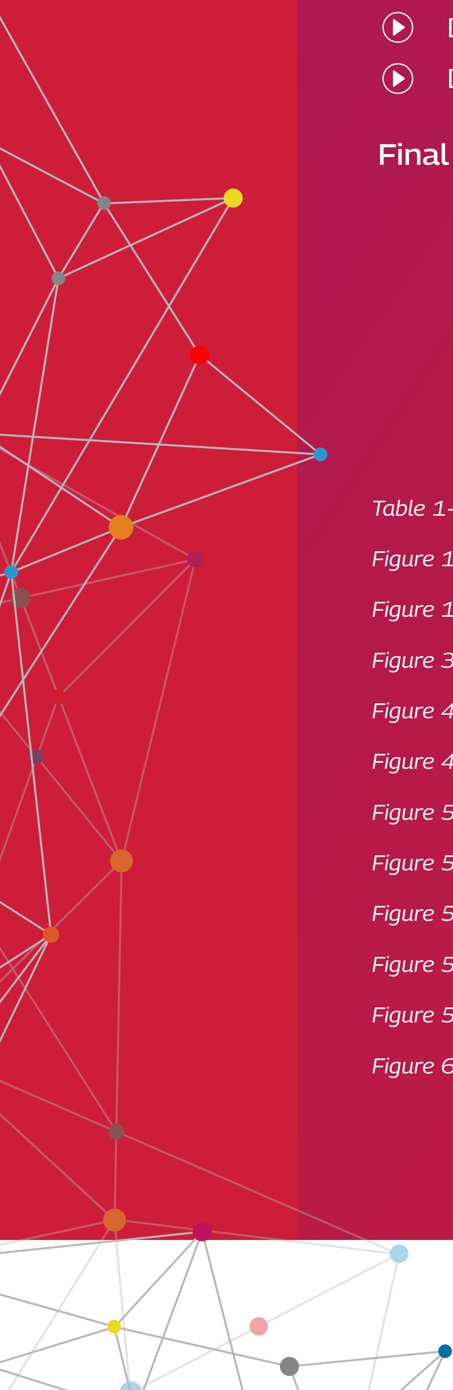
Abstract

Today, the industry is at the juncture where telecommunication networks are evolving. Communication service providers are getting transformed into digital economy players as they go beyond providing merely connectivity solutions; and to effectively collaborate across industry boundaries. Clearly, traditional ways and means to support operational needs of complex networks across collaborated ecosystem are falling short. From our global experience, we have summarized CXO asks and possible solution approach.

CXO	Key Asks	Solution Approach
CEO	Industry 4.0, an insightful approach to monetize investments, drive customer centricity, and improve sustainability	Ability to correlate data in real time across industry segments from infrastructure, network to applications. Get timely insights to monetize investments and driving customer centricity. These insights will be used to manage infrastructure in efficient way to reduce the carbon footprints.
CFO	Quick return on investments, highly optimized operations. Save OPEX to fund CAPEX	Real-time and autonomous operations to protect OPEX, provide real time and early insights to predict capacity issues that enables optimal CAPEX to spend.
CTO	Exploit data availability in real time across networks, organization boundaries to provide best in class customer experience	Build a network data lake that can ensure real time data movement in multi-cloud deployment environments. This data comprises of logs, network events, traces, and metrics. And then add a layer above to achieve real time correlation across these data sets to provide early insights.
CIO	Achieve web scale operations, move from reactive to proactive and predictive operations. Leverage as much as data that is generated by existing monitoring, assurance applications	Build a layer above existing monitoring applications that can correlate data across applications, networks, infrastructure. Predict possible rainy-day scenarios, keep customers proactively informed and protect existing IT investments.
COO	To run operation effectively with efficient use of data collected. Achieve best in class operational efficiency	Go beyond existing ways that deal with known network and infrastructure issues. Build a solution to find unknown that can affect network/service, sustainability, and customer experience.

Table 1-1 Observability Opportunity

Key Takeaways



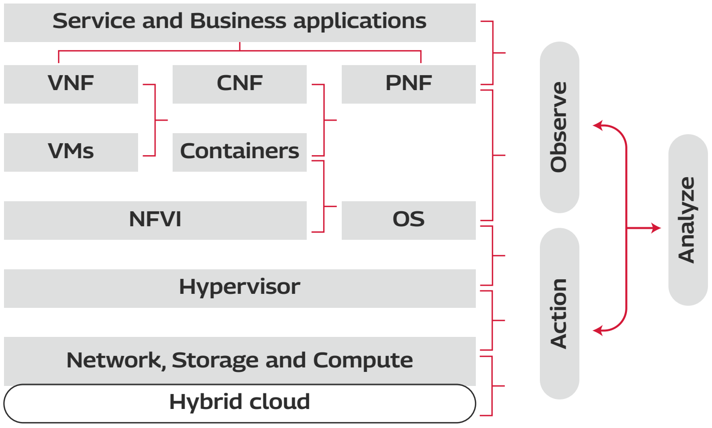
Introduction

Network observability can be the solution to cater to some pertinent CXO asks. But often observability gets confused with traditional monitoring and assurance applications. Let's look at the differentiator and value add that observability brings to the table. Traditional monitoring is focused on checking parameters and events configured by network equipment vendors in element management system (EMS). This means you know something that can happen affecting the network and service and needs attention; and you have already defined a related event in the network. But you may or may not know the exact reason behind this. This is what is referred to as known-unknown.

Based on this recent trend where network and network functions, service and related microservices are spread across multi-cloud environment in different cloud regions, it is important to think beyond monitoring and go for observing the network/service. These are some of the compelling reasons as to why we need to have network observability in multi-cloud environments.

- Ability to handle huge data sets in multi cloud environment and be able to derive meaningful insights from that data
- Driving customer centricity. Customer experience monitoring is more important than just checking the network and service is up and running
- Autonomous network operations warrant early insights with real time data feeds than a post-facto analysis with time lapsed data
- Ability to map dependency and complex relations between cloud regions, infrastructure, network functions, applications, underlay networks, and host OS
- Achieve truly DevOps culture (implement sight reliability engineering) and Webscale operations
- Globally, everyone is worried and talking about reduction in emission of carbon and improve the sustainability

Why Network Observability is Important in Current Scenario



- Complexities in current network and service architecture
- Microservices based architecture
- Different technologies
- Interlayer dependency
- Unpredicted behavior
- Proactive and predictive analysis
- Trying to find unknown which is not defined or agreed

Figure 1-2 Why Observability

By virtue of implementing a network observability solution, we would be able to derive something which is happening in the network that we are not aware of or is not predefined.

Over and above monitoring, we get below functional features through network observability:

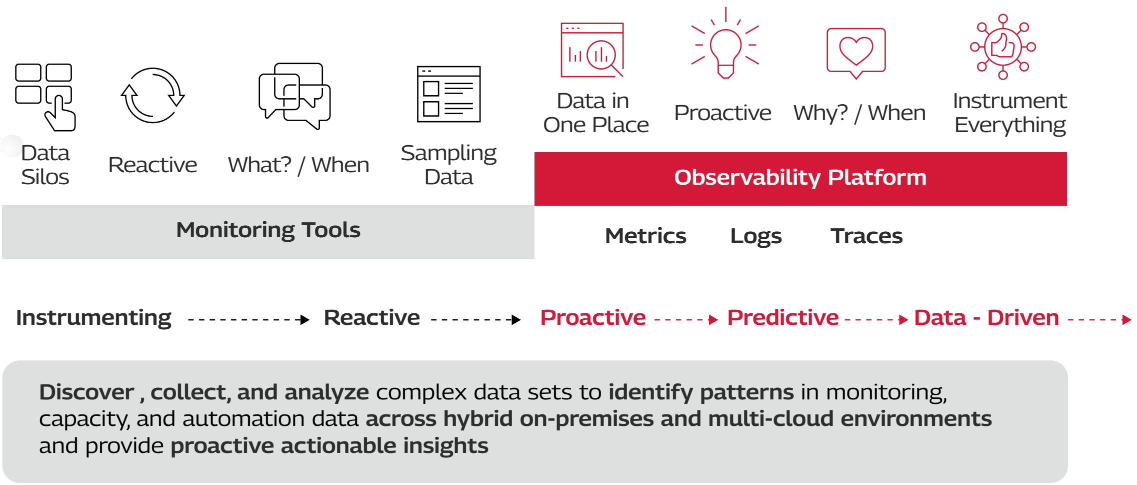
Limited data availability - In today's monitoring scenario, OSS systems or operation teams are getting only limited data which is preconfigured in network. This provides limited visibility of their network and services. This becomes a barrier to predict network and service performance; and put a limit on root cause analysis (RCA) capability.

Telemetry support - Currently, collection of data is limited and supports traditional technologies. The reason behind this is due to structure of the data, frequency of collection, and processing efficiency, collection and storage of data is very basic building block for observability. In network observability, we support all type of data with different frequency and format.

Predictive insights - Currently we are focusing on prediction which is based on reactive events. To drive customer centricity, it is important to perform prediction based on all the available data sets so that we can predict and find unknowns.

No EMS future - Do we really need tradition monitoring? Although this question is too early to answer but when we have a fully containerized network, everything can be instrumented using observability without any dependency on EMS.

Network Observability – New Approach



Discover , collect, and analyze complex data sets to **identify patterns** in monitoring, capacity, and automation data **across hybrid on-premises and multi-cloud environments** and provide **proactive actionable insights**

Figure 1-3 Observability a new approach

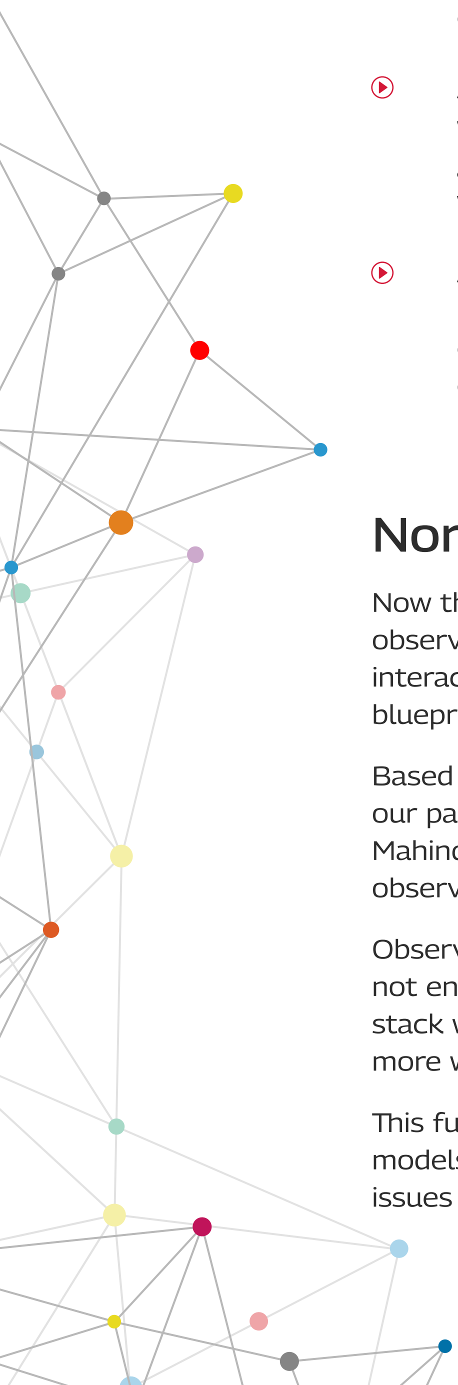
Observability is collecting different types of data from network, service, and processing them to find out some meaningful and important insight/predict insights. Figure 13 above, summarizes as to how observability is different from monitoring tools and what value does it bring to operations.

Key Telco Use Case

Based on our experience working with multiple customers across the world and representing different forum, we realized that telcos must move fast towards observability to achieve the benefit which they are expecting from technologies like 5G, CNFs, DevNetOps, and automation.

Some of the key use cases/drivers for them to shift from monitoring to observability are: -

- As a Chief Marketing Officer (CMO), I want to know my customer's network and service behavior, and trends in their consumption of data so that I can drive customer centricity by providing them early insights about the status of their service/network, consult them on the most optimal price plans for their usage
- As an innovation head, I want to know how the network, services, and tools are behaving so that I can bring new technologies and tools like AI/ML, DevNetOps, SRE/NRE to provide best-in-class service and customer experience with key KPIs like RFT, and improved MTTR
- As a planning head, I want to know how the network/service is operating, what are the sector are affecting most and where the demands are growing so that I can plan the network/service and enable fast revenue with reduced time to market.
- As an operations excellence team, I want to know the actual network/service/carbon emission status, so that appropriate measures can be taken to make sure network/services meet the QoS and customer expectation and it will help to reduce the penalties.



North Star View

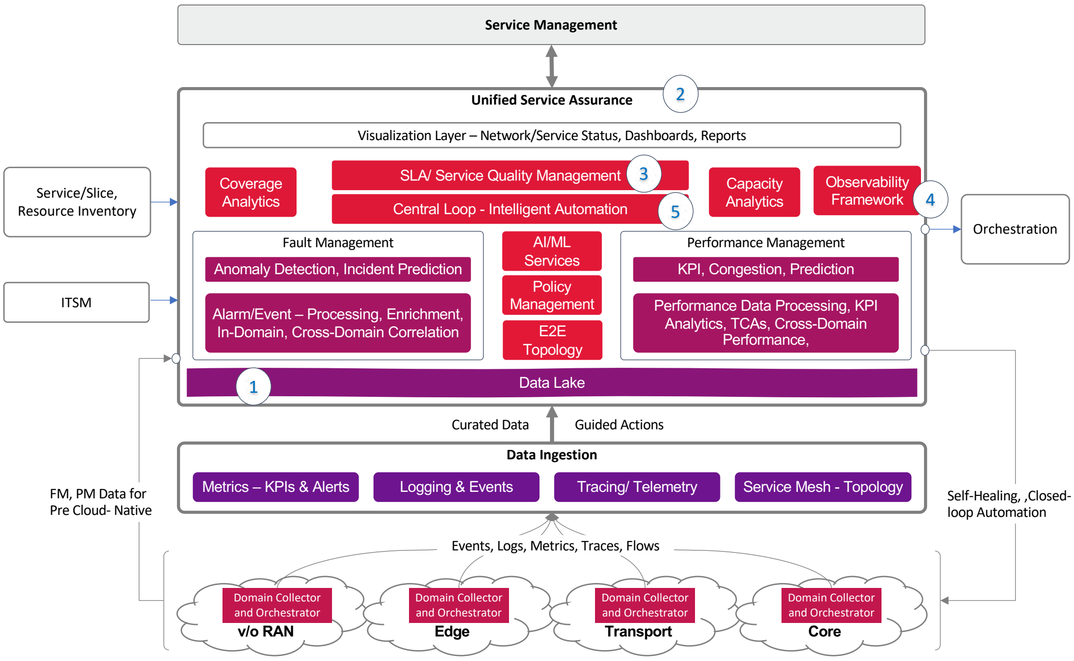
Now that we have established and appreciated the need for network observability solution, the most important point raised by CXOs that we have interacted with is the question: how it would fit in to our existing architecture blueprint. Or what would be north star view for this solution?

Based on our global experience delivering similar solutions and learnings from our participations in industry standards like TMF ODA, and ONAP, Tech Mahindra has built north star view for this solution. This view shows as to how observability (4) co-exists with the rest of the OSS solution components.

Observability only for higher layers like virtualized infra, services or customer is not enough but we need to make sure that we have full stack observability. Full stack will include rest of the parts like Infrastructure, Network, transport, and more which play important role in distributed network.

This full stack view enables self-healing and closed loop automation. Detailed models can be built using observability information from each layer to detect issues that occur across layers.

Next Generation One Assurance Reference Architecture


Service Management
Unified Service Assurance
2
Visualization Layer – Network/Service Status, Dashboards, Reports
Service/Slice, Resource Inventory
ITSM
Coverage Analytics
SLA/ Service Quality Management
3
Central Loop - Intelligent Automation
5
Capacity Analytics
Observability Framework
4
Orchestration
Fault Management
Anomaly Detection, Incident Prediction
Alarm/Event – Processing, Enrichment, In-Domain, Cross-Domain Correlation
AI/ML Services
Policy Management
E2E Topology
Performance Management
KPI, Congestion, Prediction
Performance Data Processing, KPI Analytics, TCAs, Cross-Domain Performance,
1
Data Lake
Curated Data
Guided Actions
Data Ingestion
Metrics – KPIs & Alerts
Logging & Events
Tracing/ Telemetry
Service Mesh - Topology
FM, PM Data for Pre Cloud- Native
Self-Healing, ,Closed-loop Automation
Events, Logs, Metrics, Traces, Flows
Domain Collector and Orchestrator
v/o RAN
Domain Collector and Orchestrator
Edge
Domain Collector and Orchestrator
Transport
Domain Collector and Orchestrator
Core

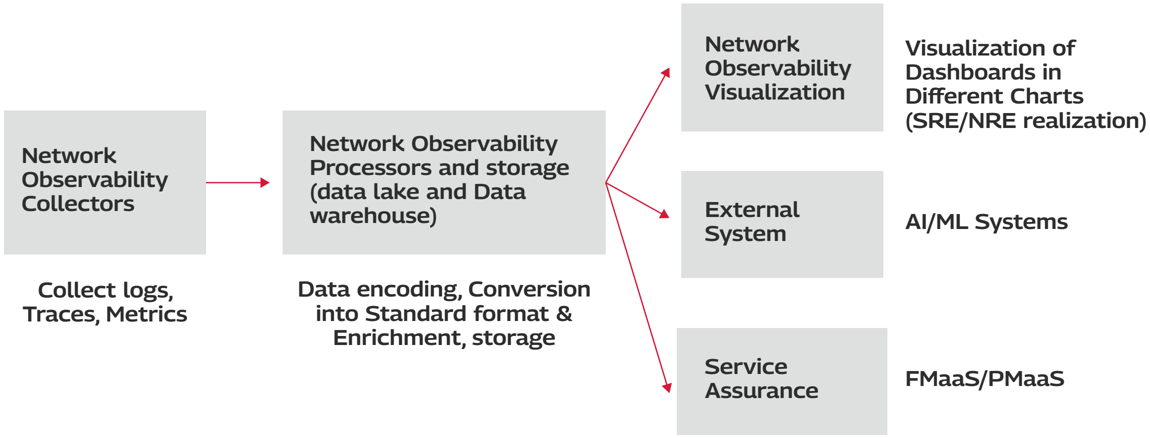
Comprehensive, end-to-end insights that correlates performance, faults, and events to applications, services, and customers	AIOps with intelligent closed-loop automation	AI and ML driven intelligent analytics	ML-powered, advanced root cause analysis	Collects and analyzes —applications, virtual and physical devices, legacy and next-gen technology

Figure 3-1 TechM view on OneAssurance

Transformation Towards OBF

At Level 0 functional representation, network observability solution has four major components.

Network Observability – L0 Functional Architecture


Network Observability Collectors
Collect logs, Traces, Metrics
Network Observability Processors and storage (data lake and Data warehouse)
Data encoding, Conversion into Standard format & Enrichment, storage
Network Observability Visualization
Visualization of Dashboards in Different Charts (SRE/NRE realization)
External System
AI/ML Systems
Service Assurance
FMaaS/PMaaS

****All the different stages Operator/OEM may not to rewrite but they can use existing solution considering it fulfill the essential characteristics of Network observability like it support different frequency, format, real time and AI/ML based analytics, and more.

Figure 4-1 Observability- L0 Functional Architecture

Collection layer - observability is focused on collection of

- Logs
- Metrices
- Traces

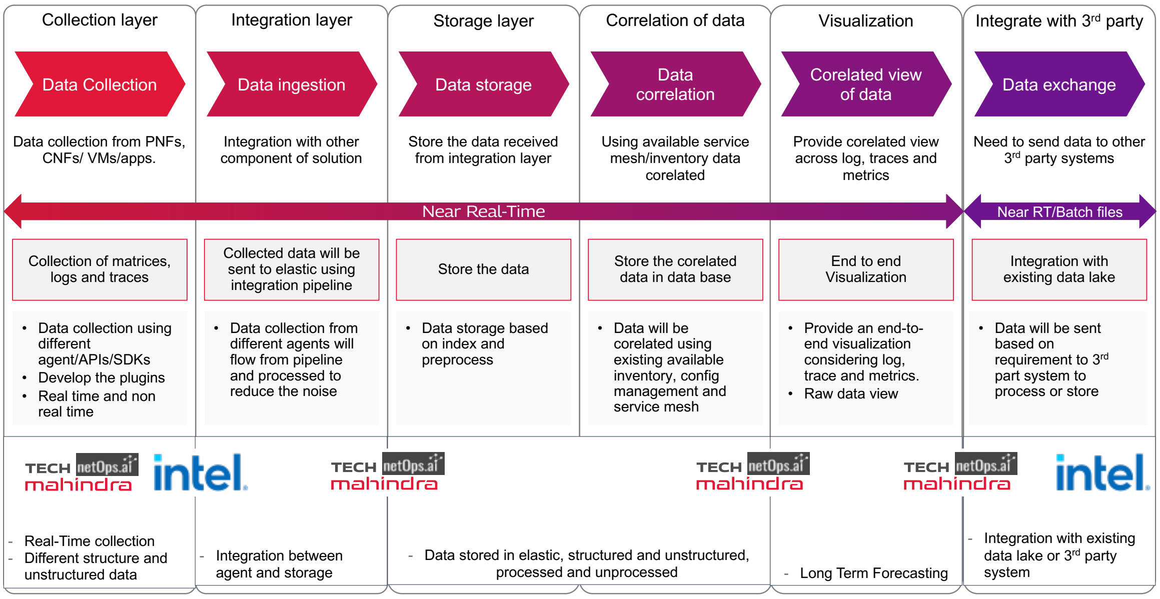
Using the telemetry observability solution will collect the data from different sources available. Some popular data sources are Prometheus, Nagios, beats agents, and OpenTelemetry the data coming in from these sources is not in the same format and frequency. Observability collection layer must adopt these data format and frequency.

Preprocessing with storage – Once observability solution collects the data, very first thing is to preprocess of data to contextualize and remove noise. Processed data need to be stored in a data lake so that different system can consume it like FM for fault, PM for performance, SIEM for security and vulnerability and many more.

GUI – To know what is exactly happening and what we derive out of data we must define some specific KPIs which will not only give an indication what's happening but also suggest what could be done to avoid any possible outage or degradation. This equips the DevOps team to take necessary corrective measures to preempt events.

External integration – To integrate the output of observability solution with systems like fault, performance management, and predictive insights.

Different Activities with Network Observability



Collection layer	Integration layer	Storage layer	Correlation of data	Visualization	Integrate with 3rd party
Data Collection	Data ingestion	Data storage	Data correlation	Corelated view of data	Data exchange
Data collection from PNFs, CNFs/ VMs/apps.	Integration with other component of solution	Store the data received from integration layer	Using available service mesh/inventory data corelated	Provide corelated view across log, traces and metrics	Need to send data to other 3rd party systems
Near Real-Time					Near RT/Batch files
Collection of matrices, logs and traces	Collected data will be sent to elastic using integration pipeline	Store the data	Store the corelated data in data base	End to end Visualization	Integration with existing data lake
• Data collection using different agent/APIs/SDKs • Develop the plugins • Real time and non real time	• Data collection from different agents will flow from pipeline and processed to reduce the noise	• Data storage based on index and preprocess	• Data will be corelated using existing available inventory, config management and service mesh	• Provide an end-to-end visualization considering log, trace and metrics. • Raw data view	• Data will be sent based on requirement to 3rd part system to process or store
- Real-Time collection - Different structure and unstructured data	- Integration between agent and storage	- Data stored in elastic, structured and unstructured, processed and unprocessed		- Long Term Forecasting	- Integration with existing data lake or 3rd party system

Figure 4-2 Different activities for network observability

Tech Mahindra's Network Observability Platform

Day 1 - Build environment

As day 1 below steps are performed

- Building opensource based netops observability environment to collect telemetry
- Using CI/CD deployment of agents like Beats, OTEL solution with different applications
- Enhancement of Helm chart to deploy this entire solution with few click
- Using default GUI, showing the different data collected

Below is the architecture overview in line with our OneAssurance solution.

Current Architecture available in TechM Lab

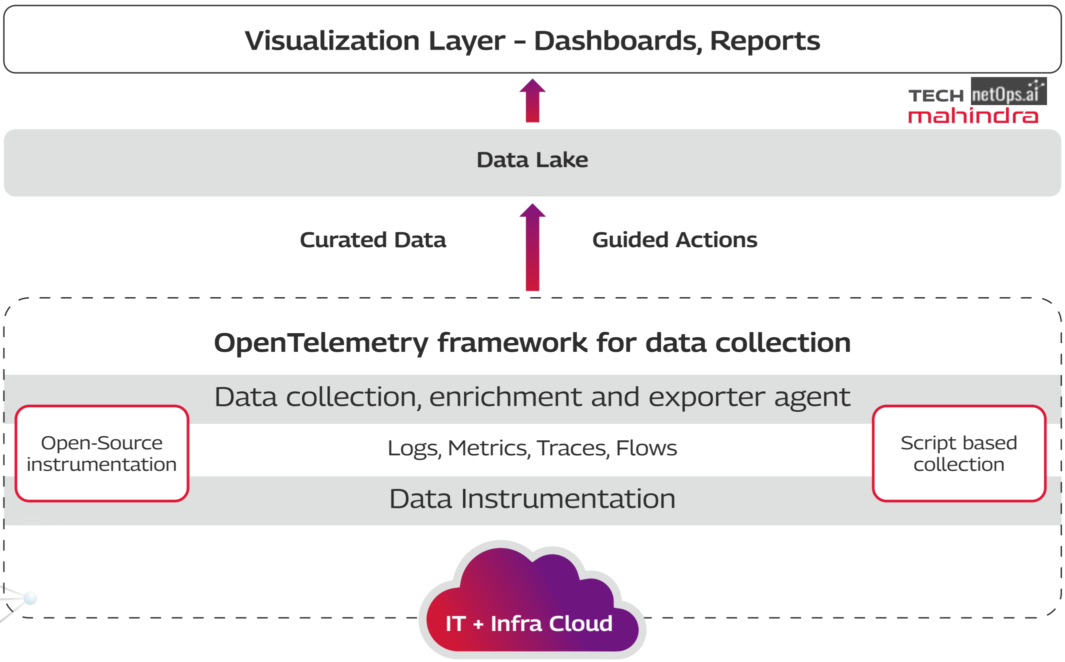

Visualization Layer - Dashboards, Reports
TECH mahindra netOps.ai
Data Lake
Curated Data
Guided Actions
OpenTelemetry framework for data collection
Data collection, enrichment and exporter agent
Open-Source instrumentation
Logs, Metrics, Traces, Flows
Script based collection
Data Instrumentation
IT + Infra Cloud

Figure 5-1 Current TechM lab architecture for observability

Considering this as Day 1 solution here we are only showing Raw data with some standard charts. We are using OpenTelemetry for instrumentation, collection, and enrichment of data because of its nature to collect all the data without deploying multiple agents.

OpenTelemetry overview - OpenTelemetry is CNCF project which is merger of 2 old project OpenTracing and OpenCensus community. Basically, it is collection of auto-instrumentation, APIs and SDKs which will help to instrument and collect the logs, matrices, and distributed tracing. This solution also helps to contextualize the data automatically. This will help to understand the behavior of applications/functions and their functional performances.

Below diagram depicted high level difference between Traditional telemetry and OpenTelemetry.

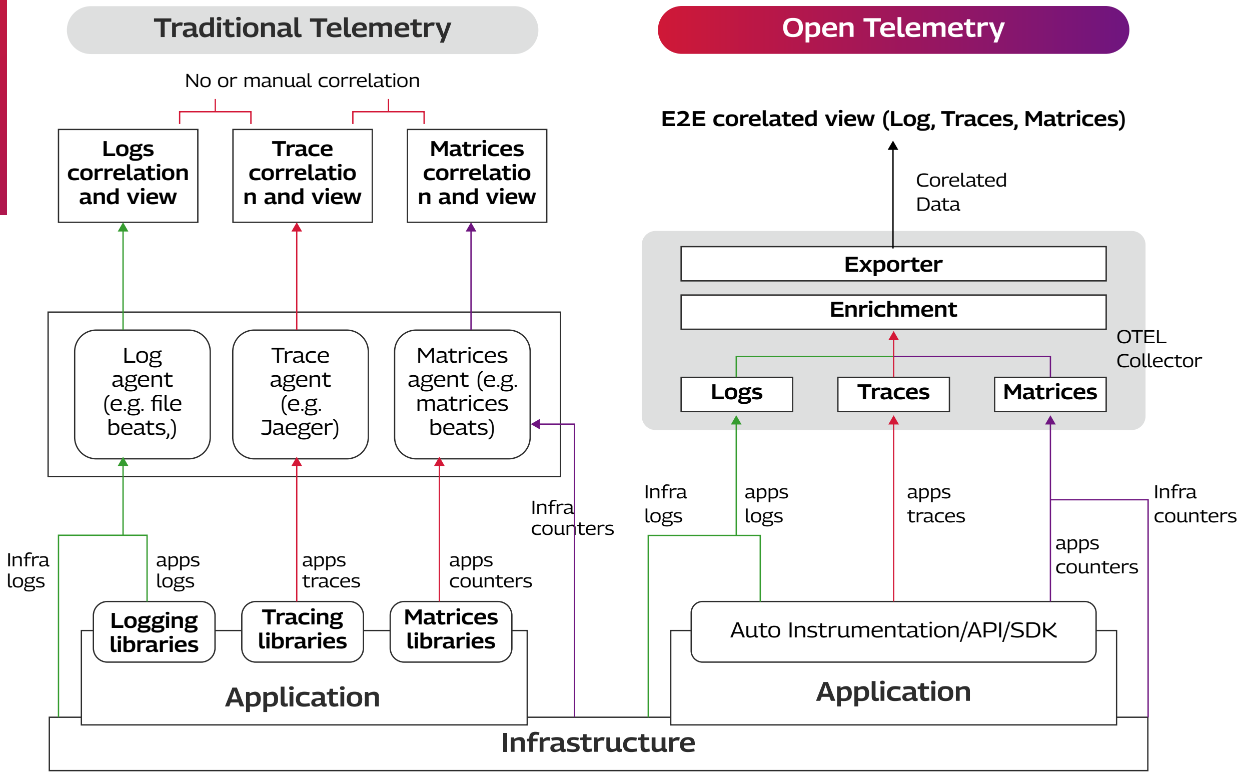


Figure 5-2 OpenTelemetry view

OpenTelemetry can have its disadvantages:

- Library Requirements As mentioned in Figure 5-2, we can see that we need to build different libraries for collecting the different data but using open telemetry it is done using single library.
- Collection Requirement – We can see that we need different agent for collection different type of data like for logs it will be Jaeger agent for files it will be file beat. But for OpenTelemetry it will be collected using single OTEL collector.
- OTEL collector support collection of data from different sources like Prometheus, Jaeger.
- OTEL collector also support export integration with Prometheus, Jaeger, Elastic, and Splunk almost all the observability tool.
- Vendor agnostics solution to export and import the data.
- Data can send to multiple destination in parallel.

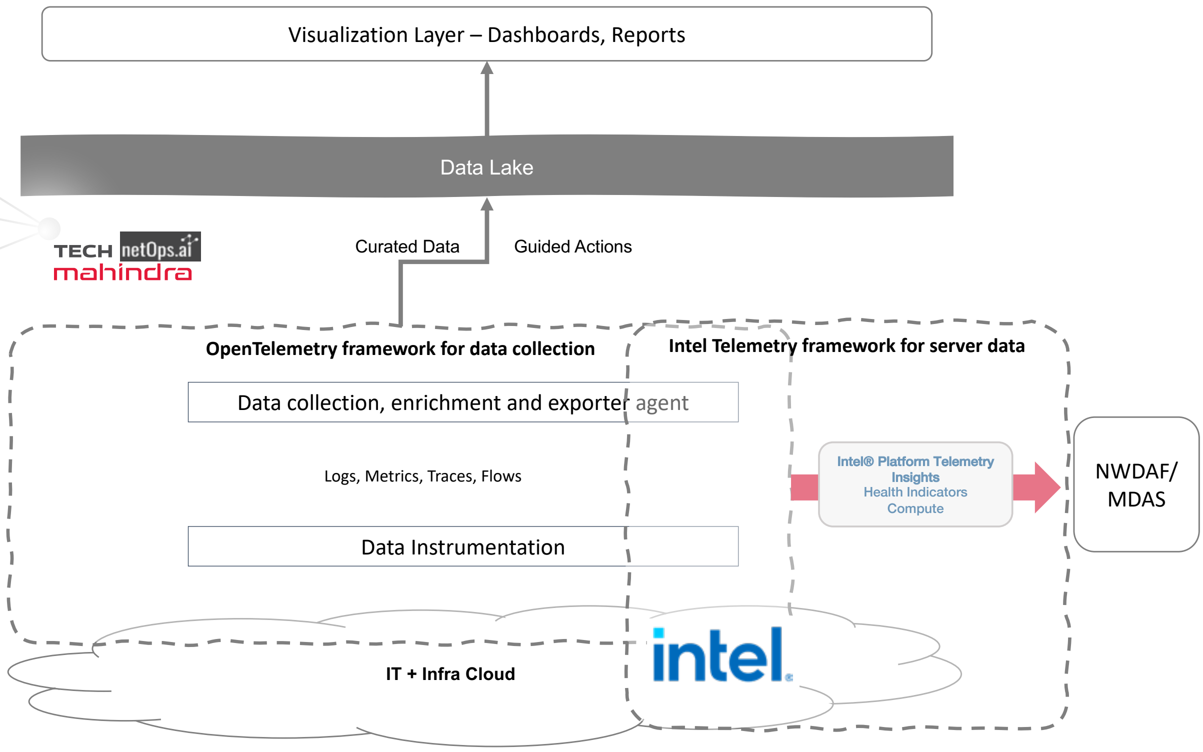
Day 2 – Enhancements in solution

A basic environment is now ready. We need to make use of the solution using these key activities which will happen in next 3-6 months.

- Integrate with Intel observability platform to provide overview about infrastructure behavior and end to end impact
- Processing data (to realize predictive insights) using AI/ML framework available in ELK solution
- Alerting based on configured threshold
- Integrate 5G solution available in AWS lab like CasaSystem, ENEA with ELKs observability solution
- Cross analysis the data like traces, metrics, events, and logs(out of the box based on predefined algorithms)
- Start visualizing the data
- Integrate with 3rd party system for scenario like AWS Sagemaker for AI/ML, close-loop, and self-healing for orchestration

Intel has an observability framework that can be leveraged to build network observability solutions. Using open-source telemetry collectors like Telegraf, infrastructure data can be collected. Leveraging OpenTelemetry collector as universal collector for preprocessing and sending data to elastic. On Kibana will help to visualize corelated view for the end customer to provide the complete stack view.

Day 2 - TechM Lab


Visualization Layer – Dashboards, Reports
Data Lake
TECH netOps.ai mahindra
Curated Data
Guided Actions
OpenTelemetry framework for data collection
Intel Telemetry framework for server data
Data collection, enrichment and exporter agent
Logs, Metrics, Traces, Flows
Data Instrumentation
Intel® Platform Telemetry Insights
Health Indicators
Compute
NWDAF/ MDAS
IT + Infra Cloud
intel

Intel Telemetry Framework - One of the challenges we observe in the industry, to offer a complete network observability solution, it needs to be able to monitor data from all layers of the stack. Network function metrics provide KPI's for the service running but without correlating them with how the underpinning infrastructure is operating the end user does not have the complete picture. Data from the underpinning infrastructure is a key component in being able to root cause issues in a timely manner to ensure no loss of service and end to end visibility.

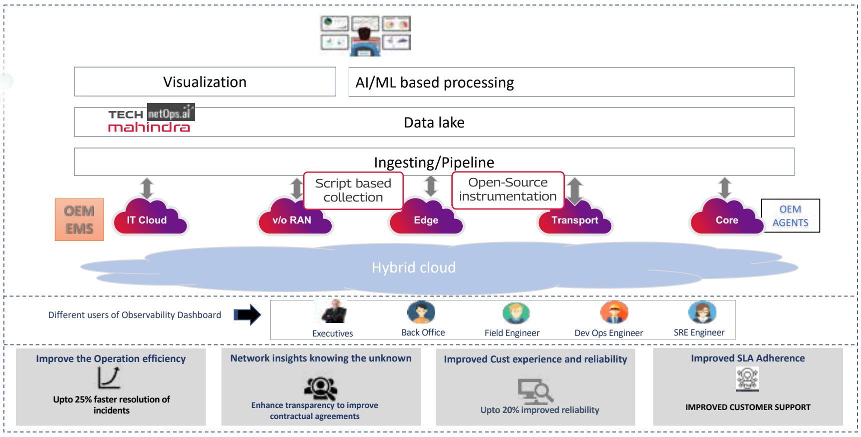
Intel platform telemetry insights uses this fine infrastructure telemetry and provides meaningful information and actionable data across single or multiple servers. These insights distil IA metrics into networking and operational data covering platform health, resource utilization and congestion, and can be used in automated control and orchestration systems to trigger remediation actions. Currently where all the industries are focusing on reducing operational carbon, these insights can help identify opportunities to save power and carbon across networks, and influence closed loop actions to save power in a more efficient manner.

These metrics are available as derived insights, which are available as both insights which can be scraped and available as infrastructure status change notifications, which are individual alerts generated using Kafka streaming.

Day 3 -Advancement of solution (considering 5G)

- More integration with respect to next gen Services and support models like NWDAF, RIC, and MDAF
- Define the cross KPIs for 5G to give E2E understanding
- Define KPIs across infrastructure and network functions
- Build end to end call flow from customer PoV
- Realizing SRE/NRE, and cloud loops

Tech Mahindra Network Observability Solution View



Visualization
AI/ML based processing
TECH mahindra netOps.ai
Data lake
Ingesting/Pipeline
Script based collection
Open-Source instrumentation
OEM EMS
IT Cloud
v/o RAN
Edge
Transport
Core
OEM AGENTS
Hybrid cloud
Different users of Observability Dashboard
Executives
Back Office
Field Engineer
Dev Ops Engineer
SRE Engineer
Improve the Operation efficiency
Upto 25% faster resolution of incidents
Network insights knowing the unknown
Enhance transparency to improve contractual agreements
Improved Cust experience and reliability
Upto 20% improved reliability
Improved SLA Adherence
IMPROVED CUSTOMER SUPPORT

Figure 5-5 Observability end state

- ✓ Leverage TechM Observability Tools
- ✓ To enable a single pane of glass for observing the network, IT and applications
- ✓ Collect more complex data type both real time and non real time
- ✓ Current operation is focus on known but observability is focus on unknown
- ✓ Enabler for operation team to manage systems and handling operations more
  - Effectively
  - Efficiently
  - Proactively

Final Thoughts: Embracing The Power of Network Observability

Going forward, network observability will be the focus for communication service providers (CSPs) and enterprises. We also understand that CSPs are keen to move towards hyperscalers. Hence, it is important to think how to support multi cloud infrastructure. The objective is to build unified observability solution that caters to the complex environment.

As observability is not limited with only monitoring and performing analysis on top but it goes beyond that. Our north star vision holistically considers entire architecture where network observability going to play and important role. Below is the pictorial representation of our vision.

E2E Reliability Using Observability

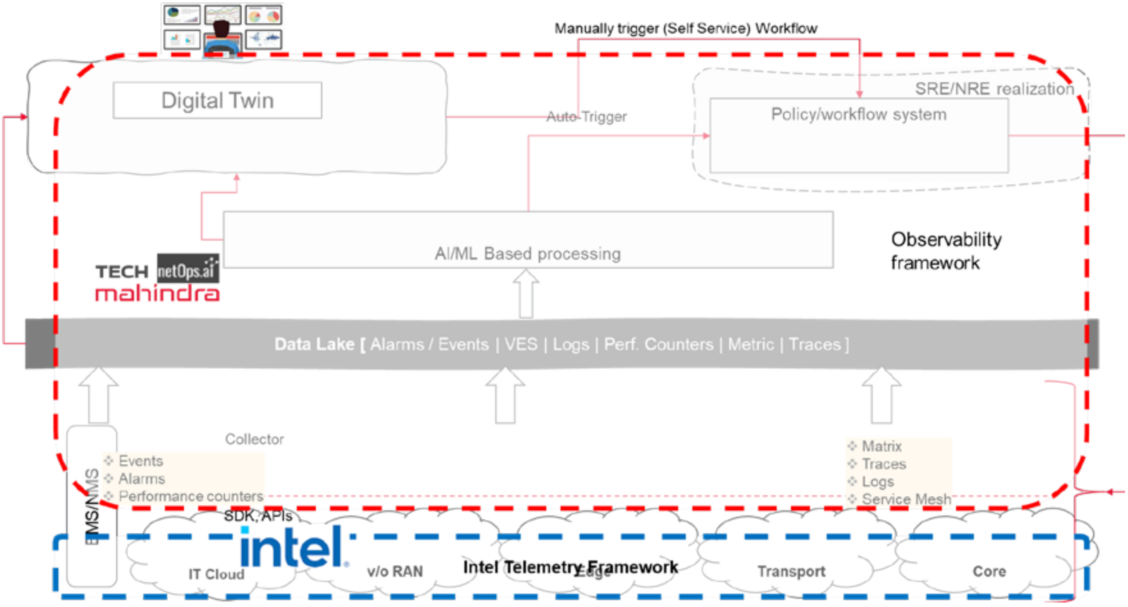


Figure 6-1 E2E reliability using Observability

The layer architecture which we are talking is mentioned below

OpenTelemetry layer - In traditional way this layer is having 2 parts, collection layer and preprocessing layer. We are merging these 2 layers to make it simple, vendor agnostic solution using OpenTelemetry. observability frameworks help us to define a plug and play kind of solution to connect with required downstream system, network functions which spread on hybrid cloud.

Data lake - Once data is out from the observability layer, it is contextualized. This data may need to refine more and need to enrich with some more information like inventory, configuration. So, all the data from observability layer need to push to data lake. Data lake will also help to push the data to other external system which may require this information for further analysis.

Analytics Engine - On top of data lake a unified layer of analytics needs to introduce. This layer will perform data analysis based on the different type of data. In traditional world this is done by FM, PM, and SIEM. What if we consider all these data as just 'data' and feed into AI/ML engine and try to find out different issues like any problem, degradation, any latency issue, possible cause of problem or any other relevant input.

Integration layer - This layer refers to TM Forum ODA architecture 'decoupling and integration'. This will help to integrate data layer and analytics engine with other components using standard protocol like Kafka, and RestAPI

Orchestration layer - Once data is processed under analytics engine, it may need to trigger some close loop action like re-configure the network. Such close loop flows, and policies are defined under orchestration layer.

Digital Twin - A user interface is important for end user to understand how the network, service and systems are behaving. Considering the availability of data, we need to create different charts, and heatmaps. This will ease the operation teams work and help them to quickly identify the issue if any.

Authors

Nilesh Kumar Sharma

Sr. Solution Architect OSS, Tech Mahindra

Dhananjay Pavgi

Competency Head OSS, Tech Mahindra

John Browne

Principal Engineer at Intel
Network And Edge Group

Emma Collins

Product Manager at Intel
Network and Edge Group